

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Gabriel Farencena Righi

**DELTAS CIRCULARES: PROPOSTA E AVALIAÇÃO DO IMPACTO DA
ARITMÉTICA MODULAR NA COMPRESSÃO DOS ÍNDICES DE
QUALIDADE DO SEQUENCIAMENTO DE GENOMAS**

Santa Maria, RS
2021

Gabriel Farencena Righi

**DELTAS CIRCULARES: PROPOSTA E AVALIAÇÃO DO IMPACTO DA ARITMÉTICA
MODULAR NA COMPRESSÃO DOS ÍNDICES DE QUALIDADE DO
SEQUENCIAMENTO DE GENOMAS**

Trabalho Final de Graduação apresentado ao
Curso de Bacharelado em Ciência da Compu-
tação da Universidade Federal de Santa Ma-
ria (UFSM, RS), como requisito parcial para
obtenção do grau de **Bacharel em Ciência
da Computação**.

ORIENTADORA: Prof.^a Andrea Schwertner Charão

COORIENTADOR: Prof. Vinícius Vielmo Cogo

©2021

Todos os direitos autorais reservados a Gabriel Farencena Righi. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.

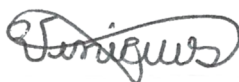
End. Eletr.: gfrighi@inf.ufsm.br

GABRIEL FARENCENA RIGHI

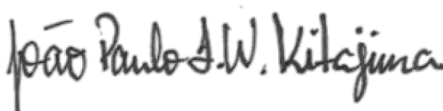
DELTAS CIRCULARES: PROPOSTA E AVALIAÇÃO DO IMPACTO DA ARITMÉTICA MODULAR NA COMPRESSÃO DOS ÍNDICES DE QUALIDADE DO SEQUENCIAMENTO DE GENOMAS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Ciência da Computação**.

Aprovado em 11 de Fevereiro de 2021:



Vinícius Vielmo Cogo, Dr. (ULisboa)
(Presidente/Coorientador)



João Paulo Kitajima, Dr. (Mendelics)

Sergio Luis Mergen, Dr. (UFSM)

RESUMO

DELTAS CIRCULARES: PROPOSTA E AVALIAÇÃO DO IMPACTO DA ARITMÉTICA MODULAR NA COMPRESSÃO DOS ÍNDICES DE QUALIDADE DO SEQUENCIAMENTO DE GENOMAS

AUTOR: Gabriel Farencena Righi

ORIENTADORA: Andrea Schwertner Charão

COORIENTADOR: Vinícius Vielmo Cogo

A genômica é uma área da genética que vem contribuindo para diversos avanços científicos, como, o do sequenciamento completo e análise de genomas de diversas espécies. Os dados gerados pelas máquinas de sequenciamento são inicialmente armazenados em modo texto no formato FASTQ, os quais podem chegar, por exemplo, a cerca de 300GB para uma única célula humana. Armazenar e analisar a sequência de milhares ou milhões de organismos é uma tarefa que requer o uso eficiente de muitos recursos computacionais, o que valida a importância de algoritmos de compressão para esta área. Os índices de qualidade (QS—*quality scores* em inglês) são a parte mais difícil de comprimir nos arquivos do sequenciamento de genomas, devido ao seu dicionário ser extenso e gerar um número muito grande de combinações possíveis. O objetivo deste trabalho é propor uma transformação de dados, os deltas circulares (CD—*circular deltas* em inglês), para os índices de qualidade do sequenciamento de genomas e avaliar o seu impacto na compressão deste tipo de dados. Essa transformação aproveita-se da observação de que índices de qualidade vizinhos variam pouco de um para o outro e explora o uso de aritmética modular para minimizar a representação destas variações. A utilização do cálculo da entropia exprime a quantidade de bits necessários para representar cada sinal, se destacando como peça fundamental no processo de avaliação. Conforme analisado a partir dos testes executados, por mais que a entropia dos valores QS fossem maiores que aqueles ND e CD, ao final a compressão se apresentou favorável a arquivos com linhas QS, permitindo uma razão de compressão superior ao arquivos com linhas ND e CD.

Palavras-chave: bioinformática; armazenamento de dados; compressão de dados; entropia

ABSTRACT

CIRCULAR DELTAS: PROPOSAL AND EVALUATION OF THE IMPACT OF MODULAR ARITHMETIC ON THE COMPRESSION OF QUALITY INDEXES OF GENOME SEQUENCING

AUTHOR: Gabriel Farencena Righi
ADVISOR: Andrea Schwertner Charão
CO-ADVISOR: Vinícius Vielmo Cogo

Genomics is an area of genetics that has contributed to several scientific advances, such as the complete sequencing and analysis of genomes of different species. The data generated by the sequencing machines are initially stored in text mode in FASTQ format, which they can reach, for example, about 300GB for a single human cell. Storing and analyzing the sequence of thousands or millions of organisms is a task that requires the efficient use of many computational resources, which validates the importance of compression algorithms for this area. Quality scores (QS) are the most difficult part to compress in genome sequencing files, because their dictionary is extensive and generates a very large number of possible combinations. The objective of this work is to propose a data transformation, the circular deltas (CD), for the quality scores of genome sequencing and to evaluate their impact on the compression of this type of data. This transformation takes advantage of the observation that neighboring quality scores vary little from one to the other and explores the use of modular arithmetic to minimize the representation of these variations. The use of the entropy calculation expresses the number of bits needed to represent each signal, standing out as a fundamental part in the evaluation process. As analyzed from the tests performed, even though the entropy of the QS values were greater than those ND and CD, in the end the compression was favorable to files with QS lines, allowing a compression ratio higher than files with ND and CD lines.

Keywords: bioinformatics; data storage; data compression; entropy

LISTA DE FIGURAS

Figura 2.1 – Demonstração do intervalo de representação encontrado nos Índices de qualidade.	16
Figura 2.2 – Comparação entre intervalos de representação dos Índices de qualidade, Deltas normais e Deltas circulares.	17
Figura 2.3 – Diagrama de probabilidades de duas máquinas distintas.	19
Figura 2.4 – Árvore de Huffman para a entrada ABCD.	22
Figura 2.5 – Árvore de Huffman para a entrada ABBCCCDDDD.	22
Figura 3.1 – Diagrama referente às etapas de desenvolvimento do trabalho	25
Figura 4.1 – Offset +75 representado em caracteres ASCII	26
Figura 4.2 – Conversão QS → ND do arquivo <i>SRR618664_1_100</i>	27
Figura 4.3 – Comparação entre intervalos ND e CD	28
Figura 4.4 – Transformação ND para QS	29
Figura 4.5 – Processo de conversão roundtrip para QS, ND e CD	30
Figura 4.6 – Histogramas genéricos QS, ND e CD respectivamente, com base no FASTQ <i>SRR618664_1_100</i>	33
Figura 4.7 – Histograma do dicionário por posição referente a primeira posição do arquivo FASTQ <i>SRR618664_1_100</i>	33
Figura 4.8 – Arquivo contendo os identificadores de acesso dos arquivos FASTQ e suas entradas.	35
Figura 4.9 – Arquivo contendo os dados filtrados dos arquivos acessados.	35
Figura 5.1 – Tabela com tamanhos dos arquivos QS, ND e CD comprimidos utilizando Huffman Codes	40
Figura 5.2 – Tabela com tamanhos dos arquivos QS, ND e CD comprimidos utilizando BSC	41
Figura 5.3 – Tabela com tamanhos dos arquivos QS, ND e CD previamente reduzidos utilizando Huffman e comprimidos utilizando BSC	42
Figura 5.4 – Tabela tamanhos dos arquivos QS, ND e CD comprimidos utilizando GZIP	42
Figura 5.5 – Tabela com tamanhos dos arquivos QS, ND e CD previamente reduzidos utilizando Huffman e comprimidos utilizando GZIP	43
Figura 5.6 – Tabela com tamanhos dos arquivos QS, ND e CD comprimidos utilizando ZPAQ	43
Figura 5.7 – Tabela com tamanhos dos arquivos QS, ND e CD previamente reduzidos utilizando Huffman e comprimidos utilizando ZPAQ	44
Figura 5.8 – Gráfico demonstrando a comparação da taxa de compressão entre as ferramentas BSC, GZIP, ZPAQ e Huffman	44

LISTA DE TABELAS

Tabela 1.1 – Representações de caracteres por bits	13
Tabela 5.1 – As principais máquinas de sequenciamento dos genomas disponíveis no SRA, o número (e percentagem) de genomas do SRA que ela representa, o número de entradas FASTQ obtidas para a avaliação e o tamanho (em bytes) dos arquivos obtidos.	38
Tabela 5.2 – Valores de entropia QS, ND e CD para cada máquina, bem como a Média Aritmética e Ponderada por entrada.....	39
Tabela 5.3 – Comparação da diferença de caracteres entre os arquivos ND e CD.....	45

SUMÁRIO

1 INTRODUÇÃO	10
1.1 CONTEXTO	10
1.2 OBJETIVOS	12
1.3 JUSTIFICATIVA	13
1.4 ORGANIZAÇÃO DO TEXTO	14
2 FUNDAMENTAÇÃO	15
2.1 FORMATO FASTQ	15
2.2 CONVERSÕES	16
2.3 ENTROPIA DOS DADOS	18
2.4 REPOSITÓRIO DE GENOMAS	20
2.5 SERIALIZAÇÃO	21
2.6 HUFFMAN CODES	21
3 METODOLOGIA	24
4 DESENVOLVIMENTO	26
4.1 CONVERSÕES	26
4.1.1 QS → ND	27
4.1.2 QS → CD	27
4.1.3 ND → QS	28
4.1.4 CD → QS	29
4.2 DICIONÁRIOS, HISTOGRAMAS E ENTROPIA	31
4.2.1 Dicionário genérico	32
4.2.2 Dicionário por posição	33
4.3 METODOLOGIA PARA OBTENÇÃO DOS DADOS	34
4.4 HUFFMAN CODES	35
5 AVALIAÇÃO	37
5.1 DESCRIÇÃO DO AMBIENTE DE AVALIAÇÃO E ARQUIVOS UTILIZADOS	37
5.2 ENTROPIA	39
5.3 HUFFMAN CODES	40
5.4 COMPRESSÃO	40
5.4.1 BSC	41
5.4.2 GZIP	42
5.4.3 ZPAQ	43
5.5 ANÁLISE COMPARATIVA	44
6 CONSIDERAÇÕES FINAIS	46
6.1 CONCLUSÃO	46
6.2 TRABALHOS FUTUROS	47
REFERÊNCIAS BIBLIOGRÁFICAS	48

1 INTRODUÇÃO

Pode-se dizer que a história da Bioinformática teve início no ano de 1951, com Fred Sanger sequenciando o aminoácido da Insulina (SANGER; TUPPY, 1951). Dois anos depois, James D. Watson e Francis Crick descrevem a estrutura em dupla hélice do DNA (WATSON; CRICK, 1953), onde se iniciou o processo de mapeamento e melhor compreensão da estrutura do DNA. Em 1961, Marshall Nirenberg e Heinrich Matthaei decifram o código genético usando homopolímeros de ácidos nucleicos a fim de traduzir aminoácidos específicos (NIRENBERG; MATTHAEI, 1961). Onze anos depois, Paul Berg juntamente com Robert Symons e David Jackson realizaram a primeira recombinação de uma molécula de DNA (BERG; JACKSON; SYMONS, 1972). Somente no ano de 1977, F. Sanger, S. Nicklen e A. R. Coulson mapearam o vírus Φ X174 (SANGER; NICKLEN; COULSON, 1977), reconhecido como o primeiro genoma completamente mapeado, definindo o grande marco histórico para a Bioinformática. Então, surge um consenso de que era necessário um banco internacional de ácidos nucleicos, e em um encontro realizado pela National Science Foundation na Universidade Rockefeller em 1979, é emitida uma chamada para a criação dessa base de dados, que tem seu início oficial em 1982, sendo denominada GenBank (GOAD, 1982) (BENSON et al., 2013).

O conceito de Bioinformática foi introduzido ainda no início da década de 70, para definir o estudo de processos informacionais nos sistemas bióticos (HOGEWEG, 2011). A necessidade de um estudo utilizando tecnologia computacional se deu de fato pela crescente coleção de sequências de aminoácidos, que forneceram uma quantia muito grande de dados os quais não podiam ser aproveitados na totalidade sem a capacidade de processamento dos computadores.

A biologia molecular demonstrava que as macromoléculas carregavam informações, fornecendo assim uma importante ligação entre a biologia e a computação, a fim de processar esses dados de forma eficiente.

Com isso, as primeiras máquinas de sequenciamento começaram a ser produzidas, a fim de possibilitar o processo de ler e tratar informações genéticas que era inviável de ser executado sem o auxílio computacional.

1.1 CONTEXTO

As máquinas de sequenciamento são responsáveis por processar imagens de amostras de DNA, que por sua vez produzem arquivos no formato de texto FASTQ (COCK et al., 2009), tal processo é nomeado *basecalling*. Esses arquivos são compostos de um grande número de entradas, onde cada uma é composta por 4 linhas específicas: a primeira linha

possui um comentário identificador, a segunda passa a informar a sequência de DNA (A, C, G, T e N) sequenciada pela máquina, a terceira possui um comentário semelhante ao primeiro e na quarta linha encontram-se os Índices de Qualidade (do inglês *Quality Scores* – QS), os quais indicam o grau de confiança da máquina no sequenciamento do DNA sendo que, para cada nucleobase sequenciada pela máquina (A, C, G, T e N), esta atribui-lhe um valor para o índice de qualidade (QS) e tais valores se destacam como ponto chave do presente trabalho. Esses índices são definidos como uma taxa de erro, calculada a partir da fórmula:

$$Q = -10 \log_{10} p \quad (1.1)$$

Onde p é a estimativa da probabilidade da leitura da nucleobase estar errada. Por exemplo, considere a leitura do caractere "(" que é a representação do valor 40 em ASCII. Tal valor se refere a: $1/10^4$, resultando em 1 erro a cada 10000 valores. Ou seja, quanto maior o valor QS, maior o índice de confiança da máquina referente àquele genoma sequenciado.

Um problema encontrado no sequenciamento de genomas foi o fato desse processo se tornar custoso, exigindo um longo tempo de processamento e produzindo arquivos que requerem muito espaço de armazenamento na infraestrutura computacional. Pois é sequenciado inúmeras células de diversos indivíduos em paralelo, a fim de obter genomas de células diferentes que são separados no processo de sequenciamento, por exemplo uma única célula humana pode chegar a de 300GB. Os QS ocupam o mesmo espaço de armazenamento em modo texto (p. ex., no formato FASTQ) que as sequências de DNA, porém o processo para compressão de sequências de QS é mais complexo do que a compressão das sequências de DNA. Uma das razões para esta dificuldade maior é que a notação usada para expressar os QS (*Phred quality scores*) possui um dicionário de opções bastante grande e utiliza valores que vão de 0 a 93, demandando pelo menos 7 bits para representar cada QS. Além disso, diferentes máquinas de sequenciamento produzem arquivos FASTQ com características ligeiramente diferentes.

Algumas máquinas possuem um grau limitado de confiança, como a máquina Illumina HiSeq 2000 (Illumina Inc., 2010) (uma das mais utilizadas para o sequenciamento de genomas no mundo (COGO, 2020)), que permite apenas atingir valores de QS entre 0 e 40 (os quais podem assim ser armazenados com 6 bits cada). Este documento utiliza a Illumina HiSeq 2000 como exemplo principal, mas no decorrer do trabalho serão consideradas também outras máquinas de sequenciamento de genomas.

Nesse contexto, surgiu a necessidade de se empregar métodos de compressão de dados, de forma que estudos que utilizem dados do sequenciamento de genomas se tornem mais eficientes sem consumir muito espaço na infraestrutura computacional. Para tal, é essencial uma representação de dados ocupando um número menor de bits, assim como estudar a entropia de dados para verificar como a compressão dos mesmos pode ser utilizada para diminuir o espaço de armazenamento (COGO, 2020). Como já observado por outros estudos (p. ex., (KOZANITIS et al., 2011)), índices de qualidade vizinhos dife-

rem pouco de um para o outro, sendo que uma transformação para Deltas Normais (ND – *normal deltas* em inglês, representando a diferença de valores entre cada dois caracteres) se expressa como uma alternativa para representar esses dados, utilizando um intervalo diferente mas com uma distribuição normal de seus valores centrada no zero. Todavia, o intervalo dos ND $[-40,40]$ na máquina Illumina HiSeq 2000, por exemplo, necessita de 7 bits para ser representado, pois possui 81 opções diferentes. Assim, surge a ideia de se utilizar **aritmética modular** para reduzir a quantidade de bits necessária, fazendo uma transformação para valores circulares (CD – *circular deltas*) em um intervalo com aproximadamente metade do número de opções. Ainda citando o caso da Illumina HiSeq 2000, o intervalo dos CD passaria então para $[-20,20]$, podendo ser representado com 6 bits por ter apenas 41 opções.

1.2 OBJETIVOS

Este trabalho tem o objetivo de explorar uma transformação de dados para os Índices de Qualidade (QS) dos arquivos obtidos no sequenciamento de genomas (i.e., FASTQ), utilizando aritmética modular. Espera-se confirmar a hipótese de que esta transformação contribua para a redução destes dados e, conseqüentemente, permita uma economia do espaço de armazenamento.

A seguir, são listados os objetivos específicos deste trabalho:

- Automatizar o download de partes específicas dos arquivos FASTQ, com o intuito de evitar buscar todas as entradas de um genoma para filtrá-los depois;
- Implementar a transformação dos dados de Índices de qualidade (QS) para Deltas normais (ND) e para Deltas circulares (CD), e em seguida implementar a função inversa a fim de validar o processo, disponibilizando-as como uma biblioteca de código aberto;
- Implementar o cálculo da frequência/distribuição dos valores nos dados (p. ex., histograma) para todas representações (QS, ND e CD);
- Estudar e implementar o cálculo da entropia nos dados (que sirva tanto para QS, ND e CD);
- Implementar ou utilizar uma solução para o cálculo de Huffman Codes, com base nas distribuições dos valores QS, ND e CD;
- Selecionar, executar e comparar diversas ferramentas de compressão com as diferentes representações de dados (QS, ND e CD).

1.3 JUSTIFICATIVA

Arquivos FASTQ são arquivos de texto gerados pelas máquinas de sequenciamento, considerados como os dados em bruto do processo, após a etapa prévia de basecalling. Devido ao alto custo monetário e temporal, laboratórios evitam sequenciar novamente os genomas. Dessa forma, a compressão dos dados é um ponto chave nesse processo, a fim de que pesquisadores possam estudar genomas de forma eficiente, evitando a necessidade de muito espaço para o armazenamento de dados. Devido à dificuldade de comprimir os QS, apresentada anteriormente, a transformação dos dados para deltas circulares é uma abordagem a ser estudada com o objetivo de otimizar esse processo.

Em virtude ao baixo grau de confiança de algumas máquinas, os QS podem se limitar a intervalos pequenos, por exemplo [0-40] no caso de (Illumina Inc., 2010). A variação de QS subsequentes é pequena (KOZANITIS et al., 2011; WAN VO NGOC ANH, 2012), o que motiva a ideia de converter os QS em deltas normais (ND), os quais se resumem à diferença entre valores para os caracteres ASCII vizinhos. Porém, os ND são representados dentro de um intervalo maior (p. ex., [-40,40]), o que significa um aumento do número de opções e, conseqüentemente, do número de bits para representá-los. Como forma de contornar esse suposto problema, indica-se uma transformação adicional para converter os ND para CD, que se exemplifica como uma transformação que utiliza aritmética modular, a fim de representar os dados com apenas 6 bits. Na Tabela 1.1, é possível verificar o espaço ocupado em bits para cada representação dos valores.

Representação	Intervalo	Bits por valor
UTF-8	[0–127]	8 bits por char
ASCII	[0–127]	7 bits por char
Índices de qualidade (QS, Phred notation)	[0–93]	7 bits por valor
Índices de qualidade (QS, na Illumina HiSeq 2000)	[0–40]	6 bits por valor
Deltas normais (ND)	[-40–40]	7 bits por valor
Deltas circulares (CD)	[-20–20]	6 bits por valor

Tabela 1.1 – Representações de caracteres por bits

Inicialmente, a representação UTF-8, muito utilizada em arquivos de texto, se destaca como um tipo de codificação binária de comprimento variável que ocupa 8 bits por *char* para representar os valores QS. No entanto vale destacar que essa representação pode representar qualquer caractere universal, e para tal pode utilizar até 4 bytes. Contudo, para fins de estudo baseado nos valores QS, consideramos apenas os valores representados dentro do intervalo desses valores com 8 bits. Além desta, encontra-se a clássica representação ASCII (*Código Padrão Americano para o Intercâmbio de Informação*, em português), a qual utiliza 7 bits para representar cada caractere do intervalo QS, que assim como o caso de UTF-8, pode ocupar mais bits para valores fora desse intervalo. Em sequência

encontra-se a *Phred notation*, usada para expressar os QS, também utilizando 7 bits. Nas últimas 3 linhas da Tabela 1.1 encontram-se as representações mais utilizadas nesse trabalho: os índices de qualidade (QS), encontrados nos arquivos FASTQ da máquina Illumina HiSeq 2000, representados por 6 bits, e os Deltas Normais (diferença de valores entre cada dois caracteres), representados por 7 bits. Por fim, tem-se a representação em Deltas Circulares, utilizando apenas 6 bits por valor.

Pode-se perceber, na Tabela 1.1 que a representação em Deltas Circulares, criada a partir da utilização de aritmética modular, reduz o intervalo dos deltas inicialmente de $[-40,40]$ para $[-20,20]$, mantendo 6 bits por valor e a distribuição normal dos valores.

1.4 ORGANIZAÇÃO DO TEXTO

O restante deste texto está organizado em outros cinco capítulos. O Capítulo 2 traz uma explanação do formato FASTQ, apresenta uma fundamentação sobre o funcionamento das conversões para Deltas Normais e Deltas Circulares, sobre o repositório de genomas e a utilização da ferramenta SRA Toolkit, que foi utilizada para filtrar arquivos antes das conversões, sobre o cálculo e a importância da entropia dos dados, sobre o uso de serialização e, por fim, a teoria dos Huffman Codes. O Capítulo 3 apresenta a metodologia utilizada para o desenvolvimento desse trabalho, com suas etapas e tarefas. O Capítulo 4 apresenta detalhes sobre o desenvolvimento, referindo-se às conversões, aos histogramas, os cálculos de entropia e as operações usadas no SRA Toolkit. O Capítulo 5 aborda a aplicação dos métodos apresentados durante o projeto, bem como os resultados e uma avaliação sobre os mesmos. Por fim, o Capítulo 6 faz uma retomada em todos os pontos abordados no projeto, bem como destaca pontos em aberto que podem incentivar trabalhos futuros.

2 FUNDAMENTAÇÃO

Neste capítulo, apresenta-se inicialmente uma breve explicação sobre o formato FASTQ. Também, expõe-se um embasamento sobre as transformações propostas para fins de compressão de dados, seguido da descrição de um conjunto de ferramentas úteis para a concretização deste trabalho. Na sequência, apresenta-se o método do cálculo da entropia dos dados e a sua importância, o uso da serialização de objetos, e por fim, uma breve explanação sobre a metodologia dos Huffman Codes.

2.1 FORMATO FASTQ

Conforme foi abordado, os arquivos em formato FASTQ são ponto chave no estudo proposto neste projeto. Para tal, uma explanação detalhada deve ser feita, sobre a forma como os dados se apresentam dentro dessa estrutura.

Tais arquivos seguem o mesmo padrão, pode-se dizer que os dados se dividem em porções de 4 linhas, ou seja, a cada 4 linhas, a mesma estrutura é mantida, modificando-se apenas os dados presentes nesse conjunto. A primeira linha se inicia com a presença do caractere @, que identifica a mesma. Em seguida, o identificador da sequência, bem como da máquina que a sequenciou, e de forma opcional, pode existir uma breve descrição. A segunda linha contém o conjunto de caracteres, referentes a sequência de nucleobase de DNA, geralmente representada pelos caracteres (A, C, G, T e N), visto que todos os arquivos utilizados nessa pesquisa seguem esse padrão, porém, existe uma gama de caracteres maiores em outras representações. A terceira possui duas opções de projeção. A primeira, apenas uma linha contendo o caractere +, e a segunda, o mesmo caractere presente na primeira posição da linha, e em seguida os mesmos valores apresentados na primeira linha, se fazem presentes nessa terceira linha. Isso é um parâmetro de configuração da sequenciação, de forma que um arquivo FASTQ pode apresentar qualquer uma das duas configurações, seguindo sempre esse padrão. Na quarta linha encontram-se os valores mais importantes para essa pesquisa, pois apresentam-se os QS, que conforme apresentado anteriormente, indicam a probabilidade de que a leitura do nucleobase presente na segunda linha esteja errada. Para exemplificar, considere a entrada abaixo:

```
@SRR12763696.9179783 9179783 length=302
TTTATGGAGACCAAA...
+SRR12763696.9179783 9179783 length=302
AAAAAEEEEEEEEEE...
```

Percebe-se que a primeira linha se inicia com o caractere @ contém o identificador, bem como a terceira linha mantém a mesma estrutura, se iniciando com o caractere + Na segunda linha encontra-se a sequência de nucleobases, e na quarta linha, cada valor se refere a taxa de erro de que esta nucleobase está errada. Por exemplo, O valor A na primeira posição da quarta linha, se refere a taxa de erro do valor T presente na primeira posição da segunda linha.

2.2 CONVERSÕES

As conversões exploradas neste trabalho têm origem em uma tese que apresentou um pipeline de armazenamento introduzindo consciência na privacidade, economia e auditabilidade no ecossistema de armazenamento de dados para genomas humanos (COGO, 2020). Durante essas pesquisas anteriores, vislumbrou-se o potencial das conversões para diminuir o espaço de armazenamento utilizados nos arquivos FASTQ, tema que poderia ser explorado em trabalhos futuros.

Se resalta a informação que os valores QS subsequentes possuem uma variância pequena, ou seja caracteres vizinhos tendem a estar próximos em valores ASCII, e essa diferença calculada é representada pelos Deltas normais. Com isso, para diminuir a quantidade de bits necessários para a representação dos QS, propõe-se uma transformação de dados para Deltas normais e Deltas circulares. Inicialmente, os valores de QS encontrados dentro dos arquivos FASTQ são representados em 93 valores ASCII distintos, dentro do intervalo [33,126], onde o caractere '!' representa o valor 0 e o caractere '~' representa o valor 93, demonstrado na Figura 2.1.

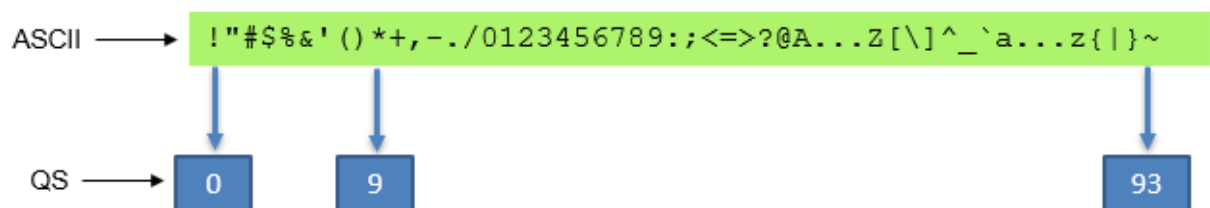


Figura 2.1 – Demonstração do intervalo de representação encontrado nos Índices de qualidade.

Como ilustra a Figura 2.2, a fim de diminuir a quantidade de bits necessários para representar os valores, estuda-se a possibilidade de transformar os dados inicialmente representados por ASCII para Deltas Normais que trabalham no intervalo [-40,+40], sendo representados nos arquivos mantendo o primeiro caractere e os demais recebendo apenas o valor do deslocamento entre os valores [33,126] ASCII. Após obtermos a conversão dos valores para Deltas Normais, a transformação para Deltas Circulares utiliza aritmética modular, calculando o resto da divisão de valores e especificando em qual intervalo cada

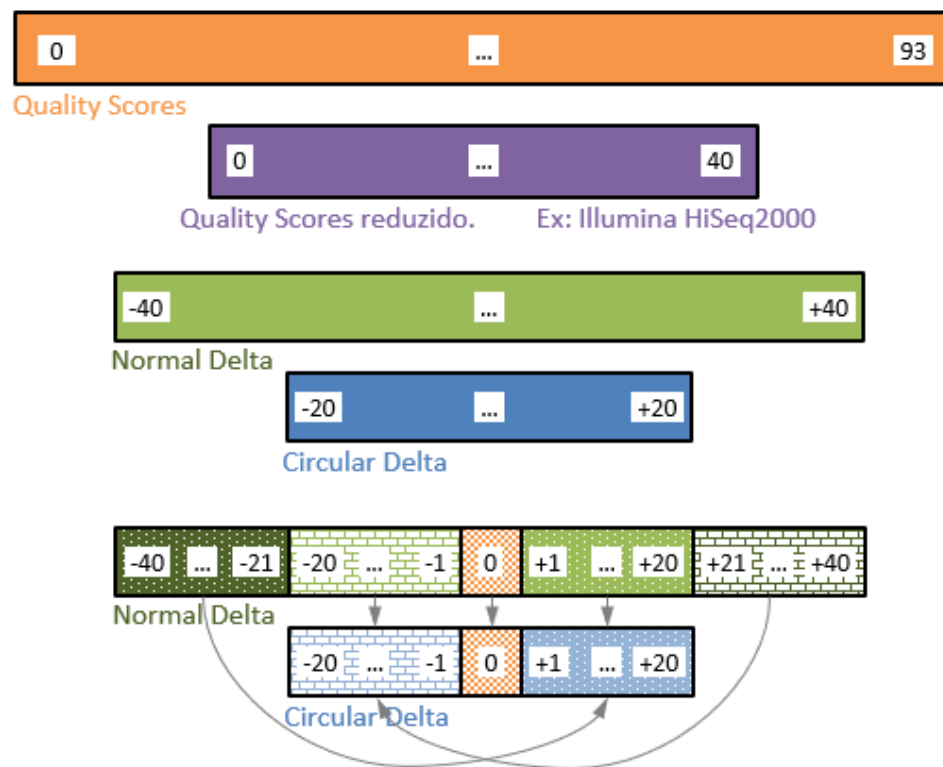


Figura 2.2 – Comparação entre intervalos de representação dos Índices de qualidade, Deltas normais e Deltas circulares.

valor está representado. Valores anteriormente pertencentes ao intervalo $[+21, +40]$ passam a ser representados dentro do intervalo $[-20, -1]$, assim como os valores do intervalo $[-40, -21]$ passam a ser representados dentro do intervalo $[+1, +20]$. Isso é ilustrado na Figura 2.2, que utiliza como exemplo valores dentro do intervalo atingido pela máquina de sequenciamento Illumina HiSeq 2000 (Illumina Inc., 2010).

Tendo como base as propostas de (KOZANITIS et al., 2011) a respeito de Deltas Normais, em que fica em aberto o estudo da profundidade e impacto das transformações na entropia dos dados, surge a oportunidade de explorar mais a fundo, e estatisticamente, a consequência da conversão em arquivos de diversas espécies e genomas sequenciados por diferentes máquinas de sequenciamento. Nos trabalhos de (COGO; PAULO; BESSANI, 2020) e (COGO, 2020), explora-se a possibilidade de Deltas Circulares de forma limitada, visto que a análise foi feita exclusivamente do ponto de vista do ganho de compressão, e não de entropia, com genomas de uma única espécie (humanos) e somente uma máquina de sequenciamento (Illumina Inc., 2010). Assim, no presente trabalho, apresenta-se como possibilidade de estudo o impacto dos Deltas Circulares referente a entropia e, consequentemente, sua contribuição para a compressão de arquivos de diferentes máquinas de sequenciamento.

Dessa forma, o uso das conversões deve ser usado com o intuito de reduzir o tamanho dos arquivos, sendo comprovado tal fato a partir do uso da entropia de dados,

que é contextualizada na Seção 2.3 desse capítulo.

2.3 ENTROPIA DOS DADOS

Entropia é um termo muito usado dentro dos estudos da física, química e matemática. Na área da computação, mais especificamente no estudo da teoria da informação, a entropia se define como o nível médio de informação ou incerteza inerente aos resultados possíveis de uma determinada variável. Seu conceito foi introduzido por Claude Shannon em 1948 (SHANNON, 1948). Para melhor exemplificar o conceito de entropia no estudo da teoria da informação, considere o seguinte exemplo:

Sejam duas máquinas 1 e 2, que produzem sequências de letras A, B, C e D, consideradas as fontes de informação. O objetivo desse exemplo é definir qual autômato produz mais informações.

Cada máquina possui uma probabilidade fixa para a produção de cada caractere. A máquina 1 produz todas as letras com a mesma probabilidade (0,25).

$$P(A) = 0,25$$

$$P(B) = 0,25$$

$$P(C) = 0,25$$

$$P(D) = 0,25$$

Porém, a máquina 2 possui probabilidades distintas para a produção de cada símbolo, sendo esta:

$$P(A) = 0,5$$

$$P(B) = 0,125$$

$$P(C) = 0,125$$

$$P(D) = 0,25$$

As informações fornecidas por qualquer uma das máquinas dependem da quantidade de perguntas necessárias para prever a próxima letra a ser escrita. A quantidade de informações é diretamente proporcional ao número de perguntas necessárias. Para ambas as máquinas, considere um diagrama com repostas de Sim/Não. A Figura 2.3(a) representa o diagrama de probabilidades da máquina 1, enquanto que a Figura 2.3(b) representa o diagrama de probabilidades da máquina 2.

Para cada caractere da máquina 1, seriam necessárias 2 perguntas para obter a sua devida resposta, visto que a probabilidade de escrita das letras é a mesma. Por exemplo, para alcançar a letra “D” seria preciso responder a primeira pergunta (A ou B?) para avançar ao ramo da direita e, em seguida, responder a pergunta (C?), resultando na obtenção do caractere “D”.

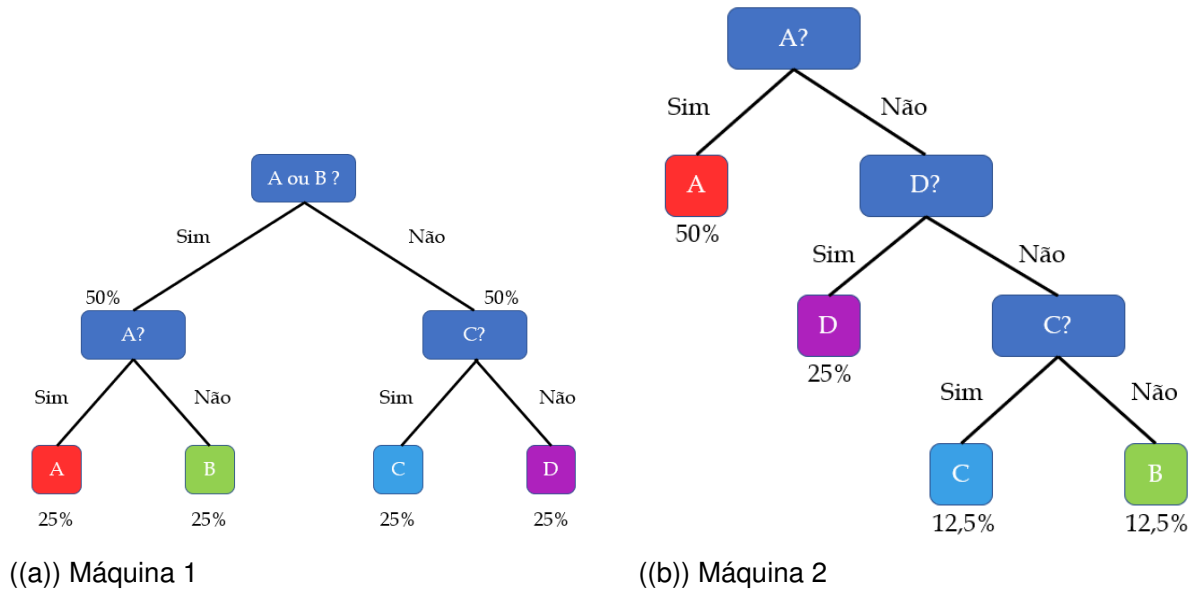


Figura 2.3 – Diagrama de probabilidades de duas máquinas distintas.

O cálculo para a média de perguntas necessárias para ambas as máquinas é feito através da multiplicação da probabilidade de ocorrência com a quantidade de perguntas essenciais para atingir aquele determinado caractere. A equação a seguir demonstra o cálculo referente a máquina 1.

$$\begin{aligned}
 E &= 2 * P(A) + 2 * P(B) + 2 * P(C) + 2 * P(D) \\
 E &= 2 * (4 * (0.25)) \\
 E &= 2
 \end{aligned}$$

No caso da máquina 2, para chegar ao caractere “A” é preciso uma pergunta, para “D” duas, e “C” e “B” 3 perguntas. Dessa forma o cálculo é executado da seguinte maneira:

$$\begin{aligned}
 E &= 1 * P(A) + 3 * P(B) + 3 * P(C) + 2 * P(D) \\
 E &= 0.5 + 3 * (0.125) + 3 * (0.125) + 2 * (0.2) \\
 E &= 1.75
 \end{aligned}$$

Logo, podemos perceber que a máquina 2 gera menos entropia do que a máquina 1.

Vale destacar que as equações acima se assemelham ao cálculo da Entropia de Shannon, que pode ser definida pela equação a seguir, que trabalha com as probabilidades de ocorrência de um aglomerado em um conjunto maior, utilizado base binária. Tal cálculo será usado nas Seções 4.2.1 e 4.2.2.

$$E = \sum_{i=1}^n P(x_i) \log_2 P(x_i) \quad (2.1)$$

Sendo assim, neste trabalho, pretende-se usar a entropia dos dados para comparar os arquivos FASTQ com os arquivos ND e CD, avaliando se as conversões de fato dimi-

nuem a quantidade de bits necessários para representar os dados e consequentemente, reduzem o espaço de armazenamento necessário dos arquivos (VASINEK; PLATOS, 2017).

2.4 REPOSITÓRIO DE GENOMAS

O SRA (Sequence Read Archive) é um repositório público de dados de sequenciamento de genomas, que fornece diversos arquivos para estudos e pesquisas, os quais serão utilizados neste trabalho para desenvolver as conversões.

Como mencionado anteriormente, os arquivos FASTQ podem se tornar muito grandes e, consequentemente, consumir muito espaço de armazenamento. Por mais que exista uma grande quantidade de dados disponíveis, como citado em (LEINONEN; SUGAWARA; SHUMWAY, 2010), é pouco eficiente buscar uma enorme quantia de dados de genomas completos para apenas se utilizar as linhas referentes aos índices de qualidade.

Dessa forma, faz-se necessária uma ferramenta capaz de filtrar as informações de cada arquivo com o propósito de obter apenas as informações importantes para as conversões. O SRA Toolkit oferece diversas opções para manipulação de arquivos, de forma a otimizar o processo de busca, selecionando apenas algumas entradas de interesse em um arquivo FASTQ. Essa abordagem permite analisar e comparar mais genomas de mais máquinas de sequenciamento do que se buscamos em todo o arquivo antes de analisá-lo.

Em síntese, o SRA Toolkit é uma ferramenta que permite ao usuário consultar um banco de dados contendo informações referentes a diversos genomas, dando permissão para executar comandos de manipulação e obtenção de arquivos. As ferramentas disponibilizadas pelo SRA Toolkit que foram analisadas e comparadas neste trabalho são: *fastq-dump*, *fasterq-dump*, e *vdb-dump*. As ferramentas possuem funcionalidades semelhantes, sendo o *fasterq-dump* uma versão mais otimizada de *fastq-dump*, habilitando a funcionalidade de multi-threading, mas se tornando limitado a arquivos inteiros. Já o *vdb-dump* permite obter informações referentes a cada entrada dos arquivos, sendo assim o mais recomendado para a tarefa de busca de arquivos no banco, enquanto que o *fastq-dump* se apresenta como o mais aconselhável para manipular os arquivos, retornando as informações mais importantes de cada entrada.

As ferramentas disponibilizadas pelo SRA Toolkit possibilitam obter os dados-alvo desse estudo, bem como filtrar e manipular os mesmos, auxiliando as tarefas desse trabalho. A implementação para tais funções é apresentada na Seção 4.3.

2.5 SERIALIZAÇÃO

No âmbito da ciência da computação, no contexto de armazenamento e transmissão de dados, o conceito de serialização se dá pelo processo de tradução de estruturas de dados ou estado de objeto em um formato que possa ser armazenado e posteriormente reorganizado no mesmo ou em outro sistema computacional. Comumente, a serialização é o processo de transformar um conjunto de informações em uma sequência de bytes.

Além da possibilidade de gravação de dados em disco, a mesma possui algumas vantagens de uso e é bastante abordada na área de redes de computadores. A serialização de um objeto gera um stream de bytes, sendo esse necessário para envio de informações dentro de uma rede. Outro ponto positivo que podemos destacar é o uso da serialização como método para a implementação de chamadas de procedimento remoto (RPC), que utiliza comunicação entre processos para oferecer a um programa a possibilidade de invocar um procedimento em outro espaço de armazenamento, (p.ex., outro computador conectado a mesma rede).

A serialização de objetos é um elemento chave que busca reduzir o tempo de transferência, saturação de rede, processamento dos dados enviados e armazenamento das informações. Como exemplo disso tem-se o trabalho de (CASTILLO JONATHAN ROSALES, 2018), que propôs um algoritmo para otimizar a serialização binária, diminuindo em 25% o tamanho dos arquivos serializados, se comparados aos arquivos convencionais, e em 50% o tempo de serialização. Dessa forma, o presente trabalho busca utilizar serialização a fim de diminuir o espaço necessário para armazenar as informações e otimizar o processo de transferência de dados.

2.6 HUFFMAN CODES

Os códigos de Huffman foram primitivamente propostos em 1952 durante uma tese de doutorado proposta por David A. Huffman. Desde então, esse método de compressão se tornou mundialmente famoso e passou a ser usado em diversas ocasiões.

Os Huffman codes baseiam-se na probabilidade de ocorrência dos símbolos em um determinado conjunto. A partir da identificação da repetição de cada símbolo, cria-se uma árvore binária, de forma que os símbolos que mais são reiterados naquele dado conjunto, ficam mais próximos da raiz, enquanto que aqueles que raramente aparecem, são representados no fundo da árvore. Dessa forma, os símbolos com maior repetição, tendem a ser representados com a menor quantidade de bits possíveis, enquanto que símbolos que raramente são identificados dentro do conjunto, são representados com uma quantia maior de bits.

Para exemplificar o processo de criação da árvore de Huffman, considere a situação

demonstrada na Figura 2.4), que apresenta uma árvore gerada para um conjunto de dados que considera apenas os caracteres *A*, *B*, *C* e *D*. Nesse caso, inicialmente considerou-se que cada valor possui uma repetição, de forma que o total de caracteres pertencentes no conjunto é 4. Para criar a árvore, cada etapa consiste em criar uma bifurcação entre os valores com menor frequência, em que o novo nó pai criado contém a soma dos dois nós filhos.

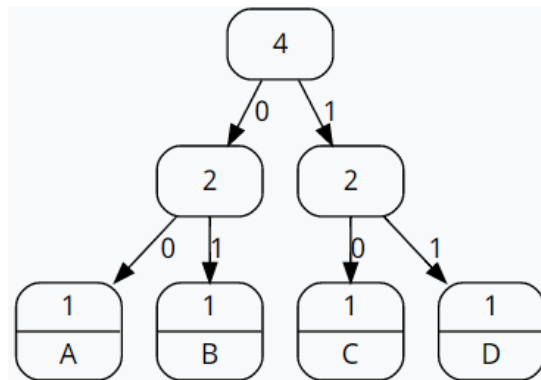


Figura 2.4 – Árvore de Huffman para a entrada ABCD.

Percebe-se que todos os caracteres precisam de no mínimo 2 bits para serem representados.

Agora, considere o exemplo apresentado na Figura 2.5, que contém os mesmos caracteres contidos na Figura 2.4, mas com uma diferença na frequência de cada valor. Nesse caso, a entrada total se apresenta como: *ABBCCDDDD*. Percebe-se que aqueles caracteres que possuem uma maior frequência no conjunto, tendem a ser representados pelo menor número de bits possíveis, enquanto que aqueles com menor frequência, são representados com mais bits.

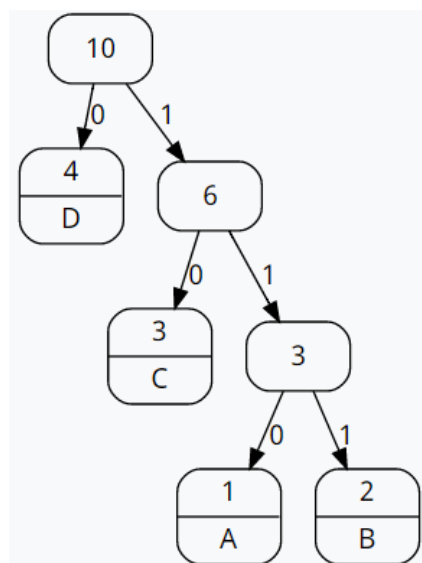


Figura 2.5 – Árvore de Huffman para a entrada ABBCCDDDD.

Conforme apresentado pelos dois exemplos, se a entrada *ABBCCCDDDD* utilizasse a árvore apresentada na Figura 2.4, o total de bits usados para representar essa sequência de caracteres seria 20, enquanto que a mesma entrada, representada pela árvore da Figura 2.5 utilizaria 19 bits. Com isso, percebe-se que quanto maior for a frequência do caractere dentro de um conjunto, maior seria a compressão, considerando o número de bits, pois uma grande repetição daquele caractere, influenciaria em boa parte da entrada ser representada com poucos bits, reduzindo consideravelmente o tamanho do resultado final.

Essa proposta de compressão sem perdas, apresenta-se como uma ótima opção para redução do espaço de armazenamento dos arquivos dessa pesquisa.

3 METODOLOGIA

Este trabalho configura-se como uma pesquisa exploratória e aplicada, destinada a solucionar um problema recorrente em bioinformática. A hipótese é que a transformação aplicada aos índices de qualidade contribuirá para a redução do valor da entropia, que pode posteriormente causar um impacto na redução do espaço de armazenamento. Desta forma, o trabalho envolveu o desenvolvimento de uma solução que implementou a transformação proposta e também experimentos para avaliá-la quantitativamente.

O presente Trabalho de Conclusão de Curso se relacionou com atividades de Iniciação Científica desenvolvidas em uma equipe com dois estudantes e dois professores orientadores. Cada aluno ficou responsável por uma parte da pesquisa/implementação, de forma a obter um maior conhecimento em um tema específico e, no final, ambas as partes se unificaram, por possuírem dependências entre elas.

Um dos estudantes esteve responsável por explorar as ferramentas do SRA Toolkit, necessárias para filtrar as entradas dos arquivos FASTQ e otimizar o tempo necessário para fazer os testes. Com isso, foi possível filtrar apenas as informações importantes para o trabalho, evitando a aglomeração de arquivos que consomem muito espaço de armazenamento, os quais foram necessários para o desenvolvimento das conversões.

O estudante autor do presente trabalho ficou responsável por implementar os métodos de conversão dos dados para Deltas Normais e Deltas Circulares. Para avaliar as conversões, foram utilizados os dados filtrados com SRA Toolkit.

Após a implementação das conversões em forma de uma biblioteca, foi implementado o cálculo da frequência/distribuição dos valores nos dados, utilizando histogramas para demonstrar os valores obtidos e desenvolvido o cálculo da entropia nos dados (que serviu tanto para QS, ND e CD). Depois, focou-se no estudo dos Huffman codes e de como eles puderam auxiliar no processo de distribuição de valores, usando cálculos de entropia. Na sequência, selecionou-se ferramentas de compressão, como GZIP, BZIP2, ZPAQ, LPAQ8, BSC, LZMA2, etc., a fim de demonstrar que a redução da entropia facilita a compressão dos dados, permitindo uma maior razão de compressão e consequentemente uma economia de espaço de armazenamento.

Finalmente, foram configurados e realizados experimentos com um conjunto de arquivos produzidos pelas máquinas de sequenciamento, com intenção de coletar métricas que caracterizem, quantitativamente, o impacto da solução proposta.

Para melhor exemplificar o processo desenvolvido, a Figura 3.1, representa todos os passos, sendo o sequenciamento a primeira etapa, para obtenção dos arquivos FASTQ, a qual foi desconsiderada neste trabalho. Na sequência, a implementação das conversões dos valores QS para ND e CD. O cálculo da distribuição dos valores dos dados, gerando histogramas, juntamente com o cálculo da entropia. E ao final, a etapa de avaliação onde

foram analisados todos os arquivos FASTQ obtidos, comprimidos e, uma etapa referente as conclusões e trabalhos futuros a serem desenvolvidos, a partir dos resultados encontrados ao final do projeto.

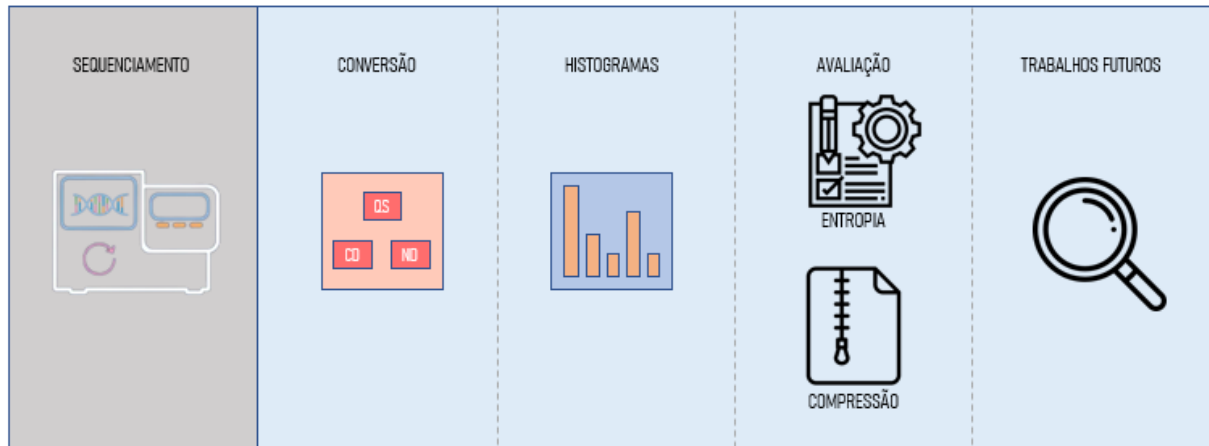


Figura 3.1 – Diagrama referente às etapas de desenvolvimento do trabalho

4 DESENVOLVIMENTO

Este capítulo apresenta o desenvolvimento dos códigos usados no decorrer do projeto. É apresentado inicialmente os códigos de conversões entre QS ↔ ND ↔ CD. Em sequência, encontram-se as implementações dos dicionários, cálculo de frequência e entropia dos dados, e por fim os códigos usados para filtragem com a ferramenta SRA Toolkit. Todos os arquivos podem ser encontrados no seguinte link: Github.

4.1 CONVERSÕES

Os códigos referentes às conversões dos Índices de qualidade para os Deltas Normais e Deltas Circulares foram desenvolvidos na linguagem C++ (STROUSTRUP, 1998), utilizando uma biblioteca auxiliar para manipulação de arquivos (*fstream*) e outras bibliotecas para representação e armazenamento dos dados (*string* e *vector*).

Todos os programas de conversões recebem como parâmetro de entrada o arquivo a ser lido, e como saída produzem um novo arquivo contendo os valores processados, p.ex., entrada: arquivo.fastq, saída: normalDelta.txt. Devido aos intervalos dos Deltas Normais $[-40, +40]$, assim como dos Deltas Circulares $[-20, +20]$, há um impedimento para representar todos os valores ASCII dos Índices de Qualidade $[33, 126]$, pois seria inviável representar o valor “-40” dentro desse intervalo.

N.Delta	ASCII Char	ASCII valor
- 40	#	35
- 39	\$	36
- 38	%	37
...
-2	I	73
-1	J	74
0	K	75
+1	L	76
+2	M	77
...
+38	q	113
+39	r	114
+40	s	115

Figura 4.1 – Offset +75 representado em caracteres ASCII

Para solucionar esse problema, foi definido que ambas as conversões devem seguir

um deslocamento de offset de +75, evitando essa dificuldade de representação de sinais ASCII. Sendo assim, o valor -40 passou a ser representado pelo caractere “#” (35), o -39 pelo caractere “\$” (36), assim como o valor 0 passa a ser representado pelo “K” (75), como demonstra a Figura 4.1.

4.1.1 QS → ND

A conversão dos Índices de Qualidade para os Deltas Normais é feita com base na diferença de valores entre caracteres subsequentes, visto que a variação de valores desses símbolos vizinhos tende a ser pequena.

Inicialmente, todas as linhas QS dos arquivos FASTQ são transformadas para Deltas Normais, mantendo fixo o primeiro caractere e, a partir do mesmo, calcula-se o valor do próximo símbolo. Isso é ilustrado na Figura 4.2, que apresenta como exemplo a conversão dos 16 primeiros valores QS da linha 12 do arquivo SRR618664_1_100, que corresponde às 100 primeiras entradas do arquivo FASTQ do genoma sequenciado com o identificador SRR618664, para Deltas Normais. O valor ASCII do primeiro caractere @ é 64, e o valor do próximo símbolo lido é 64(@). Dessa forma, a resultante do cálculo da diferença entre o caractere na posição 1 da string com o caractere na posição 2 é 0. Logo, a próxima variável ND a ser escrita é 0. Na sequência, encontra-se o símbolo C, que possui um resultado da diferença de valores entre os caracteres nas posições 2 (@ → 64) e 3 (C → 67) da entrada QS igual a +3. Logo, o próximo caractere a ser inserido na sequência ND será 3.

ASCII	→	64	64	67	70	70	70	68	70	71	71	72	71	70	73	73	71
QS	→	@	@	C	F	F	F	D	F	G	G	H	G	F	I	I	G
		↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
ND	→	@	0	3	3	0	0	-2	2	1	0	1	-1	-1	3	0	-2

Figura 4.2 – Conversão QS → ND do arquivo SRR618664_1_100

4.1.2 QS → CD

Para a obtenção dos Deltas Circulares, faz-se um mapeamento usando aritmética modular para definir os valores dentro do intervalo [-20,+20]. Como se pode perceber, o intervalo inicial de [-40,+40] é reduzido pela metade, necessitando que um valor circular

referencie dois valores normais. Como ilustra a Figura 4.3, o valor -1 pode ser tanto -1 como 40, assim como o valor -2 pode ser tanto o próprio valor -2, quanto 39.

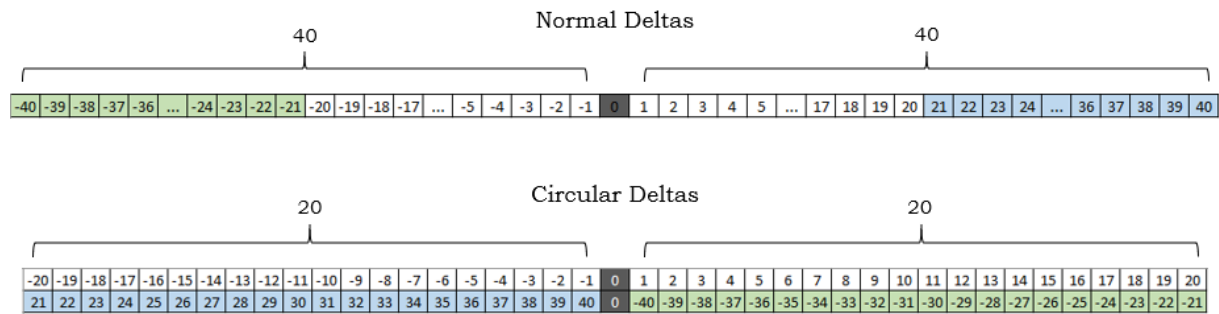


Figura 4.3 – Comparação entre intervalos ND e CD

A conversão de Índices de Qualidade para Deltas Circulares é feita a partir da obtenção dos valores em Deltas Normais, como explicado na Seção 4.1.1. Após a conversão dos valores QS para ND, calcula-se o valor de cada CD a partir de uma função desenvolvida que recebe como parâmetros o valor ND e o intervalo total, referente ao intervalo de representação dos CD $[-20, +20]$, sendo esse 41. O valor calculado dentro da função é a resultante da função:

Algorithm 1: Conversão QS \rightarrow CD

```

1 Function circular_distance(NDvalue, interval):
2   int CDvalue = (NDvalue % interval + interval) % interval;
3   if (CDvalue < (interval/2)) then
4     return CDvalue;
5   end
6   return CDvalue - interval;
```

A operação encontrada na linha 2 do algoritmo 1: Conversão QS \rightarrow CD executa a mesma operação que a função `Math.floorMod` da linguagem Java (JAVA, 1995), que retorna o módulo da operação, considerando o sinal de cada elemento. Dessa forma, é possível obter os valores CD a partir dos valores ND já calculados no método apresentado na Seção 4.1.1.

4.1.3 ND \rightarrow QS

A transformação ND \rightarrow QS foi usada para validar os resultados obtidos pela transformação inversa. Comparando-se os dois arquivos FASTQ com linhas QS, é possível verificar que não há discrepâncias e, dessa forma, comprova-se que o processo de transição ND \rightarrow QS apresentado na Seção 4.1.1 está correto.

Para tal, o processo usado para retornar os ND para QS é feito a partir da obtenção das linhas ND. Após a obtenção das linhas ND, tendo o caractere na primeira posição com

pivô, soma-se o mesmo com o valor na segunda posição, que refere-se ao deslocamento em valores ASCII a partir do caractere anterior. Após a obtenção do segundo caractere, salva-se o mesmo em uma variável temporária, para que seja usado como o novo pivô a cada repetição. O processo se repete em todas as posições até chegar ao final da linha.

Tomando como exemplo a sequência ND apresentada na Figura 4.2, a Figura 4.4 apresenta o retorno para valores QS, demonstrando o processo inverso e consequentemente validando o processo apresentado na Seção 4.1.1.

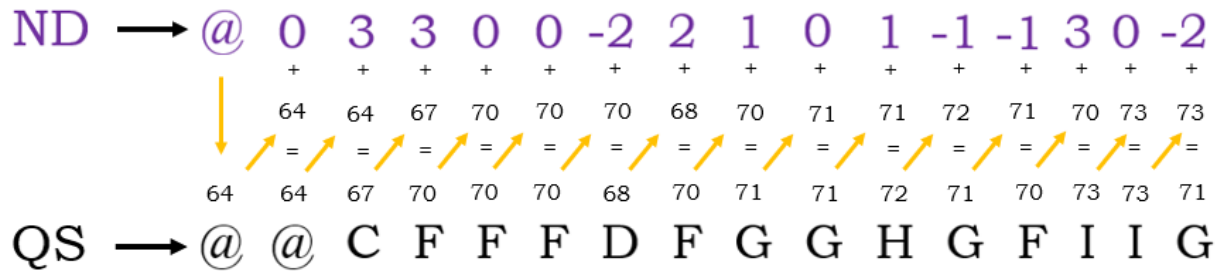


Figura 4.4 – Transformação ND para QS

4.1.4 CD → QS

A transformação de CD para QS se faz de certa forma mais simples. Assim como a conversão ND → QS apresentada na Seção 4.1.3, utiliza-se o valor atual lido, e a operação sempre é feita considerando o pivô juntamente com o valor na próxima posição do vetor que representa a linha lida do arquivo FASTQ. O primeiro valor sempre é fixo, mantendo o caractere ASCII contido na linha. A partir do segundo valor, considerando que os mesmos precisam estar dentro do intervalo dos QS, inicia-se o processo de condição (if) em que o valor -33 corresponde ao offset dos QS na notação Phred [0,93], sendo assim, o valor 0 se iguala ao ASCII 33, bem como o valor 41 corresponde diretamente ao tamanho do intervalo dos CD [-20,+20]. Caso necessário, este cálculo será tornado genérico consoante a máquina de sequenciamento utilizada. Após, a opção switch case encontrada na linha 2 do algoritmo 2: Conversão CD → QS, define qual valor é o correspondente de acordo

com o resultado obtido da operação:

Algorithm 2: Conversão CD \rightarrow ND.

```

1  if (anterior -33 + valor > 41 || anterior -33 + valor < 0) then
2      switch valor do
3          case -20 do
4              valor = 21;
5              break;
6          end
7          . . .
8          case 20 do
9              valor = -21;
10             break;
11         end
12     end
13 end

```

Percebe-se que, com o cálculo acima, é possível definir se o valor está dentro do intervalo estipulado, para que defina o valor correto em ND. Após obter o valor ND e o valor anterior, que já está em formato ND, soma-se os dois e utiliza-se a mesma operação apresentada na Seção 4.1.3, que exemplifica como retornar valores ND para QS.

Para melhor exemplificar todo o processo de *roundtrip*, considere a seguinte sequência de caracteres: # - K A Figura 4.5 ilustra a sequência desenvolvida para validar esse processo.

QS		35	45	75
	#	-	K	
ND		45-35 = 10	75-45 = 30	
	#	10	30	
CD		#	10	-11
ND		#	10	30
		35+10 = 45(-)	45+30 = 75(K)	
QS		#	-	K

Figura 4.5 – Processo de conversão roundtrip para QS, ND e CD

Inicialmente, a conversão QS \rightarrow ND se dá pela diferença entre os valores ASCII de cada caractere, mantendo o primeiro valor como pivô, resultando em: # 10 30. Na sequência, transforma-se esses valores para CD, conforme apresentado na Seção 4.1.2, a resultante do cálculo do módulo considerando o valor (10) com o valor do intervalo (41) $[-20, +20]$ é 10. Esse valor é menor que $\text{interval}(41)/2$, logo, esse e valor é mantido

como 10. Para o valor 30, o mesmo é feito, em que o resultado do cálculo do módulo é 30, porém dessa vez o valor é maior que $\text{interval}(41)/2$, então é retornado $\text{CDvalue}(30) - \text{interval}(41)$, no caso 11. Com isso, os valores foram convertidos para CD. Para retornar os mesmos para ND, executa-se o programa apresentado na Seção 4.1.4, que após confirmar que o valor está dentro do intervalo permitido, `if(... || anterior(10) - 33 + valor (-11) < 0)`, retorna o valor correspondente, nesse caso 11, como pode ser visto na Figura 4.3. Após transformar os valores para ND, basta fazer as somas e transformar para ASCII. No caso $\# + 10 = 45$ que equivale ao "-", assim como $45 + 30 = 75$ que é representado pelo caractere "K". Com isso finalizamos o processo *roundtrip* para validar o método de conversão de valores $\text{QS} \leftrightarrow \text{ND} \leftrightarrow \text{CD}$.

4.2 DICIONÁRIOS, HISTOGRAMAS E ENTROPIA

Para melhor organização e estudo de casos específicos referentes aos caracteres usados nos arquivos FASTQ, ND e CD, foram desenvolvidos dois programas que geram dicionários. Um dicionário é genérico e cataloga todos os caracteres presentes dentro de um arquivo, demonstrando a quantidade de ocorrências daqueles dentro do mesmo. Outro dicionário foi desenvolvido considerando cada posição das linhas QS, ND e CD encontradas dentro dos arquivos FASTQ (p.ex., um dicionário contendo todos os caracteres presentes nas primeiras posições de cada linha do arquivo).

Para melhor visualização, foram desenvolvidos histogramas de cada dicionário, para apresentar visualmente a diferença de ocorrências de cada caractere dentro de um dicionário específico. Com o intuito de auxiliar no envio e na utilização das fases seguintes de processamento dos dicionários utilizados, foi criada uma função para serializar os mesmos, bem como um programa para executar a desserialização.

A utilização dos histogramas permite calcular facilmente as representações de Huffman, citada anteriormente na Seção 1.2, que serão utilizadas para cada símbolo de acordo com a probabilidade do símbolo acontecer. Vale destacar que esse método define que símbolos com grande recorrência tendem a ter uma representação menor em binário, enquanto que símbolos com poucas ocorrências tendem a ter representações maiores.

Por fim, ambos os programas calculam a entropia dos dados que, como foi apresentada no Capítulo 2, resume-se em: a quantidade de bits necessários para representar um determinado símbolo, ou um conjunto de símbolos. Ambos os programas foram desenvolvidos na linguagem Python (ROSSUM, 1991), usando Python3 como padrão. Utilizou-se bibliotecas auxiliares, como a *fileinput* para a leitura de arquivos, *matplotlib* para a plotagem de gráficos de histogramas, *numpy* e *math* para operações matemáticas, *os* para manipular diretórios e *pickle* para fazer uso da serialização dos dados.

4.2.1 Dicionário genérico

O dicionário genérico foi criado para representar todos os caracteres contidos nas linhas QS, CD e ND dentro dos arquivos FASTQ. Inicialmente, inicializa-se uma estrutura de dados do tipo dicionário, disponível dentro da linguagem, e percorre-se o arquivo FASTQ, processando apenas as linhas que contêm os dados importantes. Para cada caractere lido, se o mesmo não estiver dentro do dicionário, acrescenta-se uma nova chave referente ao caractere e inicializa-se seu contador de ocorrências com valor 1, caso contrário, apenas incrementa-se o valor já existente.

Ao final do processo, é feita a serialização do objeto, bem como a plotagem dos gráficos e por fim o cálculo da entropia. Isso é ilustrado na função abaixo, que recebe como parâmetros a lista de valores referentes às ocorrências dos caracteres e o somatório total desses valores.

Algorithm 3: Cálculo de entropia

```

1 Function calculate_entropy(list, sum):
2   ent = 0;
3   for i in range(0, len(list), 1) do
4     ent = ent + ((list[i]/sum) * log2(list[i]/sum))
5   end
6   return -ent;
```

Como exemplo, ao executar programa utilizando o arquivo SRR618664_1_100. Da forma resultante, a entropia encontrada foi 3.1231, e o histograma criado pode ser visualizado na Figura 4.6(a).

Pode-se perceber que a maior parte dos caracteres dentro do dicionário refere-se a ocorrências do símbolo ASCII “#”, e a quantidade de bits necessários para representar todos os dados dentro desse dicionário é 4.

Ao executarmos o mesmo programa para os arquivos ND e CD gerados a partir do arquivo FASTQ SRR618664_1_100, podemos perceber que há uma diminuição significativa na entropia dos dados, devido à redução do intervalo de representações, como demonstra a Figura 4.6(b), que resultou em uma entropia total de 2.8304.

Por fim, o histograma do arquivo CD demonstrado na Figura 4.6(c), gerado com base no FASTQ SRR618664_1_100, demonstrou uma entropia ainda menor, com o valor de 2.7777.

Vale destacar que ambos os histogramas gerados estão com um offset de +75, logo o caractere “#” com valor 35, ao ser deslocado, passou a ser representado pelo caractere “K”. Conforme os histogramas gerados, os mesmos permitem calcular facilmente as representações de Huffman, visto que operam considerando a probabilidade de ocorrer cada símbolo específico.

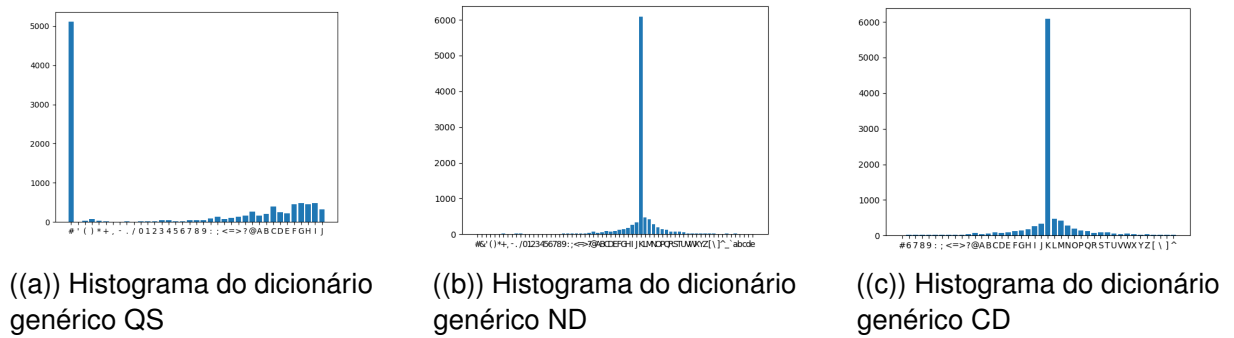


Figura 4.6 – Histogramas genéricos QS, ND e CD respectivamente, com base no FASTQ SRR618664_1_100.

4.2.2 Dicionário por posição

O dicionário por posição foi desenvolvido com o intuito de permitir a visualização da ocorrência de caracteres em cada posição das linhas dos arquivos, bem como a entropia das mesmas. É possível visualizar, no histograma da Figura 4.7, os caracteres contidos na primeira posição de cada linha do arquivo. A entropia calculada em cada ponto do arquivo FASTQ possui uma variação entre 1.0916 e 3.2610.

Da mesma forma que o dicionário genérico, o dicionário por posição permite uma melhor visualização da ocorrência dos caracteres em cada entrada do arquivo, gerando a possibilidade de utilização dos Huffman Codes, a fim de diminuir a quantidade de bits necessários para representar os dados.

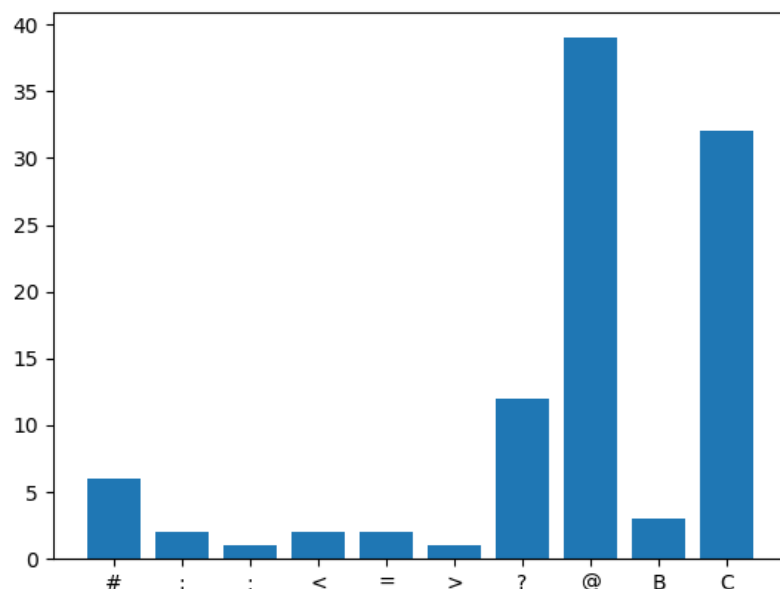


Figura 4.7 – Histograma do dicionário por posição referente a primeira posição do arquivo FASTQ SRR618664_1_100.

Para fins práticos, a abordagem na avaliação considera apenas o histograma genérico, porque as análises são feitas com arquivos contendo sequências de tamanhos muitos

distintos, de forma que invalidaria as conclusões nos casos em que se apresentassem posições com pouca representação. Dessa forma, o dicionário com posição tem o intuito apenas de ampliar as formas de verificar os arquivos, podendo acessar cada posição de cada linha.

4.3 METODOLOGIA PARA OBTENÇÃO DOS DADOS

Como foi apresentado na Seção 2.4, o SRA Toolkit disponibiliza diversas ferramentas para consultar e manipular os dados de genomas contidos em um conjunto de bancos de dados. Após feitas as primeiras análises e testes, a ferramenta do SRA Toolkit que mais se destacou para o acesso e busca de informações dentro do servidor foi a *vdb-dump*, utilizada no programa de acesso aos dados. Já para a manipulação dos arquivos, escolheu-se a ferramenta *fastq-dump*, que melhor se adaptou as operações solicitadas.

Para melhor manipulação dos arquivos, foram desenvolvidos programas em bash, para serem executados em um terminal do sistema operacional. Foi usado como padrão o terminal Linux.

Implementou-se um programa para executar o acesso aos dados que estão armazenados dentro do banco de dados no servidor. O arquivo de acesso permite fazer buscas dos dados FASTQ, retornando o próprio arquivo e a quantidade de dados presentes dentro do mesmo (p.ex., arquivo: SRR12763735.fastq, número de entradas: 19.909,767), otimizando o download e a organização dos arquivos. Sendo assim, são gerados arquivos de saída para salvar esses dados, bem como arquivos de acesso de falhas caso ocorra algum problema durante o processamento, evitando a necessidade de recomeçar o processo. A Figura 4.8 apresenta um exemplo do arquivo *acession_size.log*, gerado com a execução do programa. Percebe-se que cada linha contém o arquivo fastq e a quantidade de entradas do mesmo.

Na sequência, implementou-se outro programa com o intuito de ler o arquivo gerado anteriormente, contendo a lista de identificadores de acesso, com os arquivos FASTQ e o número de entradas, e retornar apenas as entradas FASTQ. Dessa forma, é possível retornar apenas entradas específicas de cada arquivo, evitando a necessidade de processar o mesmo inteiramente. A Figura 4.9 apresenta um exemplo de saída após a execução do programa citado, responsável pela busca de entradas a partir de um arquivo FASTQ solicitado.

Dessa forma, como foi proposto na Seção 2.4, as ferramentas disponibilizadas pelo SRA Toolkit tornaram o processo eficiente, filtrando as entradas e retornando somente os dados necessários. Vale destacar que a seleção de dados considera arquivos FASTQ de todas as espécies, não apenas contendo genomas humanos.

sequência, todo o arquivo é percorrido, e a cada caractere lido, incrementa-se o valor de ocorrência do mesmo no dicionário. Após feita essa etapa inicial, utilizou-se a biblioteca Huffman (PYTHON, 2016) que implementa a árvore binária com os valores já distribuídos de acordo com a ocorrência. Ao final transformou-se esses valores para um *array de bytes* e em sequência, os dados foram salvos no novo arquivo.

O pseudo-código abaixo exemplifica o processo para obtenção do arquivo com valores em bytes.

Algorithm 4: Pseudo-código Huffman Codes

```
1 Function main():  
2   Cria dicionario();  
3   Lê arquivo();  
4   Para cada caractere em cada linha;  
5     Incrementa ocorrência no dicionario;  
6   Cria árvore binária com o dicionário();  
7   Transforma os valores em Array de bytes();  
8   Escreve bit a bit no arquivo();
```

5 AVALIAÇÃO

Esse capítulo tem o objetivo de apresentar e analisar os dados obtidos através dos testes feitos para comprovar se de fato, as teorias propostas nos capítulos anteriores, em conjunto com os programas desenvolvidos, são positivas.

5.1 DESCRIÇÃO DO AMBIENTE DE AVALIAÇÃO E ARQUIVOS UTILIZADOS

Ao iniciarmos a etapa de obtenção e desenvolvimento de resultados, se fez necessário a utilização de um poder de processamento mais forte que aquele anteriormente disponibilizado para gerar todos os programas iniciais. Com essa afirmação, a etapa seguinte passou a ser feita utilizando uma máquina num *cluster* que foi acessado remotamente. Tal máquina é definida como uma *Dell Power Edge R430 com 2 processadores Intel Xeon E5-2670 v3 (2.30GHz, 12C, 24T), 128GB de RAM DIMM DDR4 (0.4 ns) e 1 disco SCSI de 300GB (15k RPM)*. E para o sistema operacional, foi utilizado o *Ubuntu 20.04.1 LTS (kernel v1 5.4.0-58-generic, x86 64)*.

Inicialmente, para a obtenção dos arquivos, foi necessário a implementação de scripts em *bash* que utilizassem as ferramentas disponibilizadas pelo SRA Toolkit. Foram gerados arquivos contendo listas de acessos a genomas sequenciados em cada máquina. Esses são compostos de linhas contendo a entrada, ou seja, o identificador do genoma sequenciado, presente na primeira linha de cada estrutura, (p.ex., SRR12717033). A partir do programa desenvolvido anteriormente, apresentado na Seção 2.4, criou-se um programa que acessa os arquivos contendo as listas de acesso, e por padrão, definiu-se que seriam buscadas 10 entradas FASTQ para cada identificador de genoma de cada máquina. Assim, foram gerados os arquivos utilizados para comprovar todas as propostas definidas como foco de estudo.

Por mais que o processo de obtenção dos arquivos FASTQ fosse feito em uma máquina com alto poder computacional, e devido ao curto período de tempo disponibilizado para os testes, chegou-se a conclusão que a proposta inicial de um estudo comprovado para todas as máquinas de sequenciamento selecionadas para o projeto tornou-se inviável. Dessa forma, o estudo focou-se em comprovar as técnicas tendo como base as principais máquinas. Verificou-se que 95% dos genomas disponíveis no SRA foram sequenciados com apenas 11 plataformas de sequenciamento, todas da marca Illumina (Illumina Inc., 2021). A Tabela 5.1 demonstra a distribuição de genomas disponíveis no SRA pelas diferentes máquinas e as informações sobre os dados obtidos para a avaliação.

Como pode-se perceber nesta tabela, três das 11 principais máquinas foram descartadas nesta avaliação. Os arquivos das máquinas Illumina MiSeq (Illumina Inc., 2018)

e Illumina HiSeq 2000 (Illumina Inc., 2010) chegaram a possuir mais 600 MB e 190 MB, respectivamente. Porém, devido a inconsistências encontradas nos dados obtidos destas máquinas (p. ex., geração de bytes fora do padrão ASCII de leitura) ambos os arquivos deixaram de ser utilizados para a pesquisa. Além disso, os dados da máquina Illumina NovaSeq 6000 (Illumina Inc., 2020) também foram desconsiderados nesta pesquisa porque eles continham algumas entradas com centenas de milhares de índices de qualidade iguais, o que tornava este arquivo um *outlier* que precisaria de uma validação mais aprofundada. Assim, foi levado em conta, a utilização de 8 arquivos, gerados através de buscas nas máquinas: *Illumina Hiseq 2500*, *Nextseq 500*, *Hiseq 4000*, *HiSeq X Ten*, *GA II*, *HiSeq 3000*, *GA IIx* e *NextSeq 550*.

Vale ainda destacar que os arquivos gerados pelas entradas obtidas das máquinas de sequenciamento possuem tamanhos variáveis, o que impacta tanto no tempo de obtenção dos mesmos, bem como no tempo de processamento para conversões e compressões. Percebe-se que a quantidade de entradas varia de máquina para máquina, o que impactou no tamanho dos arquivos gerados, bem como devido a inconsistências na busca, uma filtragem foi feita, que resultou na redução total do tamanho dos arquivos. O fato de todas as máquinas serem da marca Illumina permitiu utilizar o intervalo de índices de qualidade previstos inicialmente entre o zero e quarenta. Reforçamos que os algoritmos implementados neste trabalho são genéricos o suficiente para serem adaptados para máquinas de outras marcas, com outros intervalos de dados.

Máquina	Genomas no SRA	Entradas obtidas	Tamanho obtido
01 Illumina MiSeq	432562 (29%)	—	—
02 Illumina HiSeq 2500	333884 (22%)	19983	6025279
03 Illumina HiSeq 2000	305283 (21%)	—	—
04 Illumina NextSeq 500	101746 (7%)	36576	9769816
05 Illumina HiSeq 4000	72833 (5%)	34146	9355993
06 Illumina NovaSeq 6000	55752 (4%)	—	—
07 Illumina HiSeq X Ten	43754 (3%)	31263	9234032
08 Illumina Genome Analyzer II	20329 (1%)	23871	3941521
09 Illumina HiSeq 3000	19149 (1%)	31950	8912555
10 Illumina Genome Analyzer IIx	14014 (1%)	6630	1290496
11 Illumina NextSeq 550	12330 (1%)	10124	2585977
Total	1411636 (95%)	194543	51115669

Tabela 5.1 – As principais máquinas de sequenciamento dos genomas disponíveis no SRA, o número (e percentagem) de genomas do SRA que ela representa, o número de entradas FASTQ obtidas para a avaliação e o tamanho (em bytes) dos arquivos obtidos.

Dessa forma, foi analisado a entropia referente aos valores de QS de cada arquivo, bem como a entropia dos valores convertidos para ND e CD. Além disso, foi utilizado a

implementação de Huffman Codes apresentada na Seção 4.4 a fim de reduzir o espaço de armazenamento de cada arquivo considerando a escrita em bytes. Também foram comparadas três ferramentas de compressão (BSC, GZIP e ZPAQ), levando em conta a porcentagem de compressão de cada uma referente aos arquivos iniciais, bem como os mesmos já convertidos para Huffman. Ainda na Tabela 5.1, são demonstrados o número de entradas FASTQ obtidas bem como o tamanho de cada arquivo (em bytes) avaliado.

5.2 ENTROPIA

Considerando a entropia resultante das linhas QS, ND e CD dos arquivos, a Tabela 5.2 realça a teoria que conforme os valores QS são convertidos para ND e posteriormente CD, a entropia tende a diminuir, visto que são necessários menos bits para representar cada valor. Em algumas máquinas, a entropia QS apresentou-se com o menor valor entre os calculados para ND e CD. Porém, nesses casos os valores são bem próximos, o que não causa um impacto tão grande na análise completa. Se avaliarmos a média aritmética, podemos perceber que de fato os valores reduziram depois de cada etapa de conversão. Distingue-se também que o valor de entropia tende a diminuir consideravelmente entre os valores QS e ND. Enquanto que entre ND e CD, os valores reduzem, mas esse intervalo de diferenças tende a ser menor.

Através desses resultados, inicialmente comprova-se que são necessários menos bits para representar cada sinal, permitindo uma redução no espaço de armazenamento dos arquivos, levando em conta apenas as linhas QS, ND e CD.

Máquina	Entropia QS	Entropia ND	Entropia CD
02 Illumina HiSeq 2500	3,7541	2,9259	2,8698
04 Illumina NextSeq 500	2,1144	2,4228	2,4113
05 Illumina HiSeq 4000	1,9467	2,0274	2,0092
07 Illumina HiSeq X Ten	2,3943	1,9233	1,8669
08 Illumina Genome Analyzer II	4,1335	2,6631	2,6202
09 Illumina HiSeq 3000	2,3306	2,3330	2,2957
10 Illumina Genome Analyzer IIx	3,8888	3,0103	2,9745
11 Illumina NextSeq 550	2,3247	2,5309	2,4763
Média Aritmética	2,8609	2,4796	2,4405
Média Ponderada por entrada	2,6530	2,3652	2,3295

Tabela 5.2 – Valores de entropia QS, ND e CD para cada máquina, bem como a Média Aritmética e Ponderada por entrada.

5.3 HUFFMAN CODES

Conforme foi proposto na Seção 2.6 a utilização dos Huffman Codes, sendo esse uma forma de compressão que leva em conta o índice de reincidência de cada caractere, bem como a implementação apresentada na Seção 4.4, foram executados testes de compressão nos arquivos QS, ND e CD a fim de comparar essa abordagem com as demais ferramentas a serem estudadas. Os valores obtidos podem ser visualizados na Figura 5.1.

	QS original	QS Huffman	Redução	ND original	ND Huffman	Redução	CD original	CD Huffman	Redução
t02_illumina_hiseq2500	6025279	2855552	-52,61%	6025279	2242676	-62,78%	6025279	2201188	-63,47%
t04_illumina_nextseq500	9769816	2693532	-72,43%	9769816	3083818	-68,44%	9769816	3070363	-68,57%
t05_illumina_hiseq4000	9355993	2361144	-74,76%	9355993	2579628	-72,43%	9355993	2558818	-72,65%
t07_illumina_hiseqXten	9234032	2842036	-69,22%	9234032	2458209	-73,38%	9234032	2394488	-74,07%
t08_illumina_gall	3941521	2070825	-47,46%	3941521	1353989	-65,65%	3941521	1333053	-66,18%
t09_illumina_hiseq3000	8912555	2670839	-70,03%	8912555	2734069	-69,32%	8912555	2693974	-69,77%
t10_illumina_gallx	1290496	638522	-50,52%	1290496	493003	-61,80%	1290496	487486	-62,22%
t11_illumina_nextseq550	2585977	770031	-70,22%	2585977	849499	-67,15%	2585977	831888	-67,83%
			%			%			%
Média Simples		2112810,125	-63,41%		1974361,375	-67,62%		1946407,25	-68,10%

Figura 5.1 – Tabela com tamanhos dos arquivos QS, ND e CD comprimidos utilizando Huffman Codes

Percebe-se que de acordo com os valores obtidos, uma redução de aproximadamente 4,21% de diferença entre as taxas de compressão de QS para ND, e de aproximadamente 0,48% no caso comparativo de ND para CD. Isso valida os dados obtidos referentes a entropia, porém esses ganhos não se apresentam de forma diretamente proporcionais à diminuição da entropia.

5.4 COMPRESSÃO

A fim de verificar a redução do tamanho dos arquivos, foi selecionado as três ferramentas (BSC, GZIP e ZPAQ). A escolha de tais ferramentas se apresenta por representarem um conjunto equilibrado de ferramentas com bom desempenho e que são utilizadas na prática. Por exemplo, o GZIP é das ferramentas de compressão mais utilizadas na área de bioinformática, inclusive em projetos como o 1000 Genomes project (SIVA, 2008). O ZPAQ demonstra-se como um dos algoritmos com melhor razão de compressão, assim como o BSC se apresenta como uma solução mais moderna que equilibra um bom desempenho e razão de compressão (COGO; PAULO; BESSANI, 2020). Para tais testes, optou-se por utilizar apenas as linhas QS, ND e CD dos arquivos, então os valores referentes ao tamanho dos arquivos consideram apenas arquivos contendo linhas QS, ND e CD, desconsiderando as linhas de identificação, genes sequenciados e comentários dos arquivos originais. Tal escolha foi feita pois a compressão dos valores presentes nas linhas contendo comentários, valores de DNA e identificadores podem ser feitos à parte, de forma a utilizar um algoritmo

para comprimir cada parte separadamente. O foco desse trabalho foi comprimir os valores contidos nas linhas QS, pois a compressão dos comentários e valores de DNA é um problema ortogonal a esta pesquisa, principalmente pelo fato de que já existem algoritmos que executam uma boa compressão de tais linhas. Os valores contidos nas linhas QS, ND e CD são demonstrados na Tabela 5.1. Os algoritmos desenvolvidos são genéricos para os valores QS, sendo possível ser utilizados para qualquer outro formato além do FASTQ, que contenha linhas QS, por exemplo o formato SAM/BAM para caso de sequência de genomas alinhados.

Para a utilização da ferramenta GZIP, foi utilizado o comando: `gzip -c inputFile > outputFile`. Para o BSC: `bsc e inputFile outputFile`. E para o ZPAQ: `zpaq a outputFile inputFile -m5`. Todos os comandos citados foram utilizados para comprimir os arquivos QS, ND e CD, bem como o os arquivos já previamente transformados para Huffman.

5.4.1 BSC

Os valores BSC demonstrados na Figura 5.2 expressam uma redução média de 81,73% para QS, 80,79% para ND e 80,30% para CD. Esperava-se que a porcentagem de redução seria maior nos arquivos ND e CD, visto que a entropia apontava para uma teoria de que o espaço de armazenamento QS seria inferior a ND e CD, tendo em vista a diminuição da quantidade de bits necessários para representar os sinais. Porém, percebe-se pela Figura 5.2 que no caso BSC, esses valores são inversos.

	QS original	Tamanho QS	Redução	ND original	Tamanho ND	Redução	CD original	Tamanho CD	Redução
t02_illumina_hiseq2500	6025279	1393800	-76,87%	6025279	1521040	-74,76%	6025279	1529844	-74,61%
t04_illumina_nextseq500	9769816	1581816	-83,81%	9769816	1626094	-83,36%	9769816	1627582	-83,34%
t05_illumina_hiseq4000	9355993	1257572	-86,56%	9355993	1263242	-86,50%	9355993	1581816	-83,09%
t07_illumina_hiseqXten	9234032	1092858	-88,16%	9234032	1142446	-87,63%	9234032	1154394	-87,50%
t08_illumina_gall	3941521	907490	-76,98%	3941521	957370	-75,71%	3941521	954646	-75,78%
t09_illumina_hiseq3000	8912555	1362362	-84,71%	8912555	1426208	-84,00%	8912555	1435836	-83,89%
t10_illumina_gallx	1290496	347498	-73,07%	1290496	369782	-71,35%	1290496	370248	-71,31%
t11_illumina_nextseq550	2585977	421854	-83,69%	2585977	438284	-83,05%	2585977	441964	-82,91%
			%			%			%
Média Simples		1045656,25	-81,73%		1093058,25	-80,79%		1137041,25	-80,30%

Figura 5.2 – Tabela com tamanhos dos arquivos QS, ND e CD comprimidos utilizando BSC

Com o intuito de utilizar a proposta dos Huffman Codes que representam valores com maior incidência em uma menor quantidade de bits, foi optado fazer a conversão dos valores para Huffman e na sequência aplicar a compressão utilizando o método proposto pelo BSC. A Figura 5.3 demonstra os valores obtidos aplicando o método para todos os arquivos. Percebe-se que em todos os casos, a taxa de compressão para os valores QS é maior que para ND e CD, e consequentemente, o tamanho do arquivo QS após a compressão, sempre se apresenta menor que aqueles com linhas ND e CD. Outro ponto

que podemos comparar, é o fato de que a taxa de compressão média para QS, ND e CD utilizando BSC na Figura 5.2 é maior do que a taxa de compressão média dos mesmos nos arquivos já transformados para Huffman e comprimidos com BSC apresentados na Figura 5.3.

	QS original	Tamanho QS	Redução	ND original	Tamanho ND	Redução	CD original	Tamanho CD	Redução
t02_illumina_hiseq2500	6025279	1545292	-74,35%	6025279	1752178	-70,92%	6025279	1728794	-71,31%
t04_illumina_nextseq500	9769816	1650912	-83,10%	9769816	1785498	-81,72%	9769816	1791020	-81,67%
t05_illumina_hiseq4000	9355993	1281480	-86,30%	9355993	1439132	-84,62%	9355993	1435106	-84,66%
t07_illumina_hiseqXten	9234032	1149548	-87,55%	9234032	1315028	-85,76%	9234032	1319494	-85,71%
t08_illumina_gall	3941521	1030626	-73,85%	3941521	1078582	-72,64%	3941521	1061830	-73,06%
t09_illumina_hiseq3000	8912555	1447106	-83,76%	8912555	1621662	-81,80%	8912555	1618598	-81,84%
t10_illumina_gallx	1290496	401670	-68,87%	1290496	416822	-67,70%	1290496	412970	-68,00%
t11_illumina_nextseq550	2585977	449594	-82,61%	2585977	520502	-79,87%	2585977	520890	-79,86%
			%			%			%
Média Simples		1119528,5	-80,05%		1241175,5	-78,13%		1236087,75	-78,26%

Figura 5.3 – Tabela com tamanhos dos arquivos QS, ND e CD previamente reduzidos utilizando Huffman e comprimidos utilizando BSC

5.4.2 GZIP

Através dos valores obtidos utilizando o método de compressão da ferramenta GZIP, percebe-se pela Figura 5.4 que assim como a ferramenta BSC, a porcentagem de redução no tamanho dos arquivos foi maior naqueles que possuíam as linhas QS, do que das linhas ND e CD.

	QS original	Tamanho QS	Redução	ND original	Tamanho ND	Redução	CD original	Tamanho CD	Redução
t02_illumina_hiseq2500	6025279	1780757	-70,45%	6025279	2019679	-66,48%	6025279	1994849	-66,89%
t04_illumina_nextseq500	9769816	2180827	-77,68%	9769816	2456967	-74,85%	9769816	2451818	-74,90%
t05_illumina_hiseq4000	9355993	1701568	-81,81%	9355993	1949748	-79,16%	9355993	1945330	-79,21%
t07_illumina_hiseqXten	9234032	1542895	-83,29%	9234032	1767293	-80,86%	9234032	1760137	-80,94%
t08_illumina_gall	3941521	1114666	-71,72%	3941521	1204417	-69,44%	3941521	1189365	-69,82%
t09_illumina_hiseq3000	8912555	1868500	-79,04%	8912555	2153151	-75,84%	8912555	2146858	-75,91%
t10_illumina_gallx	1290496	431699	-66,55%	1290496	462466	-64,16%	1290496	458160	-64,50%
t11_illumina_nextseq550	2585977	564330	-78,18%	2585977	645818	-75,03%	2585977	644654	-75,07%
			%			%			%
Média Simples		1398155,25	-76,09%		1582442,375	-73,23%		1573896,375	-73,41%

Figura 5.4 – Tabela tamanhos dos arquivos QS, ND e CD comprimidos utilizando GZIP

Dessa forma, decidiu-se utilizar a proposta dos Huffman Codes para reduzir o tamanho dos arquivos, e posteriormente a aplicação da compressão utilizando a ferramenta, demonstrado na Figura 5.5. Um ponto interessante que podemos abordar é a comparação entre os valores médios de redução dos arquivos originais comprimidos com a ferramenta e os arquivos já transformados para Huffman e posteriormente comprimidos. Nesse caso, os valores possuem uma diferença muito pequena, menos de 1% em todos os casos.

	QS original	Tamanho QS	Redução	ND original	Tamanho ND	Redução	CD original	Tamanho CD	Redução
t02_illumina_hiseq2500	6025279	1866903	-69,02%	6025279	2013479	-66,58%	6025279	1976253	-67,20%
t04_illumina_nextseq500	9769816	1958307	-79,96%	9769816	2599809	-73,39%	9769816	2616667	-73,22%
t05_illumina_hiseq4000	9355993	1534132	-83,60%	9355993	1986526	-78,77%	9355993	1980970	-78,83%
t07_illumina_hiseqXten	9234032	1446506	-84,34%	9234032	1812412	-80,37%	9234032	1770370	-80,83%
t08_illumina_gall	3941521	1230820	-68,77%	3941521	1178806	-70,09%	3941521	1159799	-70,57%
t09_illumina_hiseq3000	8912555	1781612	-80,01%	8912555	2200019	-75,32%	8912555	2178846	-75,55%
t10_illumina_gallx	1290496	453522	-64,86%	1290496	452792	-64,91%	1290496	447398	-65,33%
t11_illumina_nextseq550	2585977	525691	-79,67%	2585977	697934	-73,01%	2585977	690298	-73,31%
			%			%			%
Média Simples		1349686,625	-76,28%		1617722,125	-72,81%		1602575,125	-73,10%

Figura 5.5 – Tabela com tamanhos dos arquivos QS, ND e CD previamente reduzidos utilizando Huffman e comprimidos utilizando GZIP

5.4.3 ZPAQ

Assim como os resultados das ferramentas anteriores apontaram para uma redução maior nos valores QS, o mesmo ocorreu na ferramenta ZPAQ, como é possível analisar na Figura 5.6.

	QS original	Tamanho QS	Redução	ND original	Tamanho ND	Redução	CD original	Tamanho CD	Redução
t02_illumina_hiseq2500	6025279	1275445	-78,83%	6025279	1431868	-76,24%	6025279	1443334	-76,05%
t04_illumina_nextseq500	9769816	1530814	-84,33%	9769816	1578091	-83,85%	9769816	1585490	-83,77%
t05_illumina_hiseq4000	9355993	1161172	-87,59%	9355993	1210015	-87,07%	9355993	1216592	-87,00%
t07_illumina_hiseqXten	9234032	1048320	-88,65%	9234032	1094010	-88,15%	9234032	1104330	-88,04%
t08_illumina_gall	3941521	823739	-79,10%	3941521	909575	-76,92%	3941521	909760	-76,92%
t09_illumina_hiseq3000	8912555	1306775	-85,34%	8912555	1372415	-84,60%	8912555	1381347	-84,50%
t10_illumina_gallx	1290496	324765	-74,83%	1290496	354826	-72,50%	1290496	355587	-72,45%
t11_illumina_nextseq550	2585977	397166	-84,64%	2585977	412175	-84,06%	2585977	415695	-83,93%
			%			%			%
Média Simples		983524,5	-82,91%		1045371,875	-81,67%		1051516,875	-81,58%

Figura 5.6 – Tabela com tamanhos dos arquivos QS, ND e CD comprimidos utilizando ZPAQ

Percebe-se que conforme apresentado nas duas ferramentas anteriores, a taxa média de redução do tamanho dos arquivos foi maior em arquivos com linhas QS, comparados com aqueles contendo linhas ND e CD.

Da mesma forma que as ferramentas BSC e GZIP testaram a compressão para valores já convertidos para Huffman, o mesmo foi feito no caso do ZPAQ, demonstrado na Figura 5.7. E novamente, da mesma forma que aconteceu com as duas ferramentas anteriormente testadas, as taxas de compressão dos arquivos foram melhores nas situações em que os mesmos foram comprimidos somente com a ferramenta, sem a transformação prévia para Huffman.

	QS original	Tamanho QS	Redução	ND original	Tamanho ND	Redução	CD original	Tamanho CD	Redução
t02_illumina_hiseq2500	6025279	1436333	-76,16%	6025279	1684536	-72,04%	6025279	1663284	-72,39%
t04_illumina_nextseq500	9769816	1583541	-83,79%	9769816	1716089	-82,43%	9769816	1721597	-82,38%
t05_illumina_hiseq4000	9355993	1224916	-86,91%	9355993	1389791	-85,15%	9355993	1384005	-85,21%
t07_illumina_hiseqXten	9234032	1089658	-88,20%	9234032	1253357	-86,43%	9234032	1255694	-86,40%
t08_illumina_gall	3941521	967639	-75,45%	3941521	1057947	-73,16%	3941521	1041743	-73,57%
t09_illumina_hiseq3000	8912555	1386226	-84,45%	8912555	1561010	-82,49%	8912555	1549935	-82,61%
t10_illumina_gallx	1290496	386134	-70,08%	1290496	410768	-68,17%	1290496	406998	-68,46%
t11_illumina_nextseq550	2585977	415787	-83,92%	2585977	494166	-80,89%	2585977	492724	-80,95%
			%			%			%
Média Simples		1061279,25	-81,12%		1195958	-78,84%		1189497,5	-79,00%

Figura 5.7 – Tabela com tamanhos dos arquivos QS, ND e CD previamente reduzidos utilizando Huffman e comprimidos utilizando ZPAQ

5.5 ANÁLISE COMPARATIVA

Considerando os valores obtidos através da compressão dos dados usando as três ferramentas, BSC, GZIP e ZPAQ, bem como Huffman codes, chegou-se a conclusão de que a ferramenta que mais se destacou foi a ZPAQ, comprimindo diretamente os arquivos QS, ND e CD. Vale destacar, como foi mencionado anteriormente na Seção 5.3 que os valores de compressão de Huffman validaram a teoria da redução do tamanho dos arquivos de acordo com a entropia encontrada, por mais que tais dados não tenham se apresentado de forma diretamente proporcionais. A Figura 5.8 expressa um comparativo entre todas as ferramentas utilizadas. Leva-se em conta que os valores projetados se referem a média da variância de cada situação, a linha preta demonstra o intervalo de alternância entre os valores daquele caso em específico.

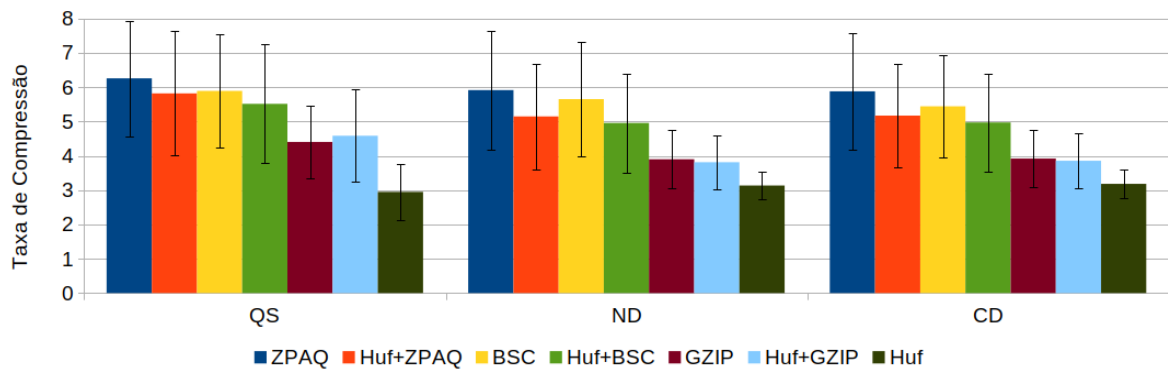


Figura 5.8 – Gráfico demonstrando a comparação da taxa de compressão entre as ferramentas BSC, GZIP, ZPAQ e Huffman

Pode-se dizer que a diferença sutil entre os valores encontrados na comparação de ND para CD, se dá pelo fato de que a ocorrência de valores ND que precisam ser transformados para CD durante a etapa de conversão é pequena, resultando em arquivos com a maioria dos caracteres repetidos. Esta conversão somente é necessária quando os valores ultrapassam o intervalo CD [-20,20]. Porém, tais valores não aparecem com tanta

frequência, visto que os QS tendem a ser valores ASCII próximos. A Tabela 5.3 demonstra a taxa de diferença entre caracteres de cada arquivo, no caso um comparativo com ND e CD. Conforme ilustrado, uma média de 3,27% dos caracteres nos arquivos foram alterados durante a conversão ND para CD.

Máquina	Número de caracteres	Caracteres diferentes	Porcentagem
02 Illumina HiSeq 2500	6025279	136777	2,27%
04 Illumina NextSeq 500	9769816	632863	6,48%
05 Illumina HiSeq 4000	9355993	214573	2,29%
07 Illumina HiSeq X Ten	9234032	262173	2,84%
08 Illumina Genome Analyzer II	3941521	47353	1,20%
09 Illumina HiSeq 3000	8912555	264576	2,97%
10 Illumina Genome Analyzer IIx	1290496	13382	1,04%
11 Illumina NextSeq 550	22585977	183360	7,09%
Média			3,27%

Tabela 5.3 – Comparação da diferença de caracteres entre os arquivos ND e CD.

Após a análise dos dados, comparando todas as ferramentas, percebe-se que para cada ferramenta avaliada, a taxa de compressão dos arquivos QS, ND e CD com o uso das mesmas se mostraram superiores às taxas referentes a compressão dos arquivos posterior a transformação para Huffman. Ao final, após a análise completa dos dados, salientou-se o fato de que em média, os arquivos QS possuem uma razão de compressão melhor que aqueles convertidos para ND e CD, por mais que a entropia fosse menor nos casos ND e CD comparado a QS.

6 CONSIDERAÇÕES FINAIS

Esse capítulo tem o objetivo de revisar todo o trabalho desenvolvido, destacando os principais pontos considerados, bem como comparar a proposta inicial do projeto, e a forma como os arquivos se apresentaram ao final das análises.

6.1 CONCLUSÃO

Os arquivos FASTQ podem consumir muito espaço de armazenamento, e os valores QS, aqueles que definem a certeza da máquina em um determinado sequenciamento de genomas, se apresenta como foco de estudo, visto que os valores podem convergir de [33,126] ASCII. Através da proposta inicial de (COGO; PAULO; BESSANI, 2020) que havia feito uma conversão dos valores QS para ND, o propósito desse trabalho passou a ser a conversão dos QS para ND e posteriormente CD, visto que reduziria o intervalo de representação para [-20,20] e consequentemente a quantidade de bits necessários para representar cada caractere.

Foram desenvolvidos programas para conversão das linhas QS para ND e CD, mantendo a estrutura padrão dos arquivos FASTQ. Posteriormente foi gerado programas para demonstrar histogramas e organizar os valores lidos dentro dos arquivos, finalizando com o cálculo da entropia de cada conjunto de caracteres. Essa etapa apontou para uma provável redução no tamanho dos arquivos, visto que a entropia representava a quantidade de bits necessários para representar os símbolos, e quanto menor a entropia, maior era o potencial de compressão. Tendo essa informação em destaque, optou-se por utilizar a proposta dos Huffman Codes, por ser a forma mais direta de se exprimir a entropia encontrada, demonstrando os dados comprimidos sem a necessidade de um algoritmo complexo. Com a utilização das ferramentas disponibilizadas pelo SRA Toolkit, os dados a serem usados para a pesquisa foram obtidos através de identificadores nos bancos de dados de arquivos FASTQ.

Dessa forma se iniciou o processo de testes e avaliações. Os testes foram feitos utilizando os arquivos já obtidos pelo SRA Toolkit, o que resultou em 8 arquivos das máquinas: Illumina Hiseq2500, Illumina Hiseq3000, Illumina Nextseq500, Illumina Next-seq550, Illumina Hiseq4000, Illumina HiseqXten, Illumina Gall e Illumina Gallx. Esses arquivos inicialmente geraram outros dois arquivos, um contendo linhas ND e outro contendo linhas CD. Após a geração dos histogramas e o cálculo da entropia de cada arquivo, percebeu-se que na maioria dos casos o valor de entropia reduzia consideravelmente entre QS e ND, tendo uma redução um pouco menor, mas ainda assim positiva no caso ND para CD. Tal fato apontava para um resultado positivo, considerando a redução do espaço de armaze-

namento dos arquivos.

Os testes realizados foram feitos em arquivos QS, ND e CD, comprimidos com as ferramentas: *BSC*, *GZIP* e *ZPAQ*. Também foram feitos testes com a compressão utilizando essas ferramentas em arquivos já convertidos para Huffman. Conforme apresentado na Figura 5.8, a ferramenta de compressão que mais se destacou foi ZPAC, seguido de BSC e GZIP.

Porém, os valores se apresentaram com melhor compressão nos arquivos originais QS, ND e CD, quando comparados com aqueles que sofreram a transformação para Huffman antes da compressão. Destaca-se o fato de que tanto na etapa de compressão com os arquivos originais, tanto quanto naqueles que foram comprimidos posteriormente Huffman, os arquivos com valores QS sofreram uma taxa de redução maior, e consequentemente geraram arquivos que necessitavam de um menor espaço de armazenamento.

6.2 TRABALHOS FUTUROS

De acordo com os dados obtidos ao final da pesquisa, alguns desafios ainda permanecem em aberto. Dessa forma, gera-se um incentivo a trabalhos futuros com o intuito de avaliar porque a razão de compressão não diminuiu, mesmo que dados como entropia apontavam para tal fator, bem como surge o incentivo de avaliar os valores gerados pelas três máquinas desconsideradas nos testes, a fim de fazer um comparativo e avaliar tais valores de forma mais exata. A fim de reduzir o número de bits por sinal, também se propõe uma implementação de ND com 7 bits e CD com 6 bits sem a necessidade de Huffman, ao invés de 8 bits da representação UTF-8 ou 7 bits ASCII. Uma outra opção será explorar a codificação aritmética nestes dados. Esta é outra forma de compressão com base na entropia, que possui uma proposta semelhante a dos Huffman Codes, mas que ao invés de substituir cada símbolo por uma nova representação em bits, codifica a mensagem em um único número dentro do intervalo $[0.0, 1.0]$. Por fim, existe a oportunidade de substituir a abordagem de compressão símbolo-a-símbolo baseada em probabilidades independentes e individuais por outras abordagens que considerem existir dependência entre sequências de símbolos para além dos deltas considerados neste trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS

BENSON, D. A. et al. Genbank. **Nucleic Acids Research**, Nucleic Acids Research, v. 41, n. 1, p. 36–41, jan. 2013. Disponível em: <<https://academic.oup.com/nar/article/41/D1/D36/1068219>>.

BERG, P.; JACKSON, D. A.; SYMONS, R. H. Biochemical Method for Inserting New Genetic Information into DNA of Simian Virus 40: Circular SV40 DNA Molecules Containing Lambda Phage Genes and the Galactose Operon of Escherichia coli. **Proceedings of National Academy of Sciences of the United States of America**, v. 69, n. 10, p. 1–6, out. 1972. ISSN 29042909. Disponível em: <<https://www.pnas.org/content/69/10/2904>>.

CASTILLO JONATHAN ROSALES, G. A. T. B. D. C. Optimizing binary serialization with an independent data definition format. **International Journal of Computer Applications**, Foundation of Computer Science (FCS), NY, USA, New York, USA, v. 180, n. 28, p. 15–18, mar. 2018. ISSN 0975 8887. Disponível em: <<http://www.ijcaonline.org/archives/volume180/number28/29152-2018916670>>.

COCK, P. J. A. et al. The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. **Nucleic Acids Res.**, Oxford University Press, Oxônia, Reino Unido, UK, v. 38, n. 6, p. 1–5, abr. 2009. ISSN 17671771. Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2847217/>>.

COGO, V. V. **Efficient, Dependable Storage of Human Genome Sequencing Data**. 2020. Tese (Doutorado) — Faculty of Sciences, University of Lisbon, PT, 2020.

COGO, V. V.; PAULO, J.; BESSANI, A. Genodedup: Similarity-based deduplication and delta-encoding for genome sequencing data. **IEEE Transactions on Computers**, IEEE Transactions on Computers, maio 2020. Disponível em: <<https://ieeexplore.ieee.org/document/6773067>>.

GOAD, W. **Genbank Overview**. [S.l.], 1982. Disponível em: <<https://www.ncbi.nlm.nih.gov/genbank/>>.

HOGEWEG, P. The roots of bioinformatics in theoretical biology. **PLOS Computational Biology**, Journal.pcbi, New York, NY, USA, v. 7, n. 3, p. 1, mar. 2011. Disponível em: <<https://pubmed.ncbi.nlm.nih.gov/21483479/>>.

Illumina Inc. **Illumina HiSeq 2000**. [S.l.], 2010. Disponível em: <https://www.illumina.com/documents/products/datasheets/datasheet_hiseq2000.pdf>.

_____. **Illumina MiSeq**. [S.l.], 2018. Disponível em: <https://www.illumina.com/content/dam/illumina-marketing/documents/products/datasheets/datasheet_miseq.pdf>.

_____. **Illumina NovaSeq 6000**. [S.l.], 2020. Disponível em: <<https://www.illumina.com/content/dam/illumina/gcs/assembled-assets/marketing-literature/novaseq-6000-spec-sheet-770-2016-025/novaseq-6000-spec-sheet-770-2016-025.pdf>>.

_____. **Illumina Website**. [S.l.], 2021. Disponível em: <<https://www.illumina.com/>>.

JAVA. **Class Math**. [S.l.], 1995. Disponível em: <<https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/lang/Math.html>>.

KOZANITIS, C. et al. Compressing Genomic Sequence Fragments Using SlimGene. **J. Comput. Biol.**, Mary Ann Liebert, Inc, San Diego, California, USA, v. 18, n. 3, p. 2–

3, mar. 2011. ISSN 401413. Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3123913/#>>.

LEINONEN, R.; SUGAWARA, H.; SHUMWAY, M. The Sequence Read Archive. **Nucleic Acids Research**, Nucleic Acids Research, v. 39, n. 1, p. 1–3, nov. 2010. ISSN D19D21. Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3013647/>>.

NIRENBERG, M. W.; MATTHAEI, J. H. The dependence of cell-free protein synthesis in e. coli upon naturally occurring or synthetic polyribonucleotides. **Proceedings of National Academy of Sciences of the United States of America**, Proceedings of National Academy of Sciences of the United States of America, National Institutes of Health, Bethesda, Maryland, v. 47, n. 10, p. 5–8, out. 1961. ISSN 1588-1602. Disponível em: <<https://www.pnas.org/content/47/10/1588>>.

PYTHON. **Huffman Pyhton**. [S.l.], 2016. Disponível em: <<https://pypi.org/project/huffman/#description>>.

ROSSUM, G. van. **Pyhton**. [S.l.], 1991. Disponível em: <<https://www.python.org/>>.

SANGER, F.; NICKLEN, S.; COULSON, A. DNA sequencing with chain-terminating inhibitors. **Proceedings of National Academy of Sciences of the United States of America**, Proceedings of National Academy of Sciences of the United States of America, Medical Research Council Laboratory of Molecular Biology, Cambridge CB2 2QH, England, v. 74, n. 12, p. 1–5, dez. 1977. ISSN 5463-5467. Disponível em: <<https://www.pnas.org/content/74/12/5463.abstract>>.

SANGER, F.; TUPPY, H. The amino-acid sequence in the phenylalanyl chain of insulin. 1. the identification of lower peptides from partial hydrolysates. **Biochemical Journal**, Biochem J., University of Cambridge, v. 49, n. 4, p. 1–19, set. 1951. ISSN 463481. Disponível em: <<https://portlandpress.com/biochemj/article-abstract/49/4/463/47212>>.

SHANNON, C. E. A mathematical theory of communication. **The Bell System Technical Journal**, Nokia Bell Labs, v. 27, n. 4, p. 623 – 656, out. 1948. ISSN 0005-8580. Disponível em: <<https://ieeexplore.ieee.org/document/6773067>>.

SIVA, N. 1000 Genomes project. **Nature Biotechnology**, Nature Publishing Group, v. 26, n. 256, p. 1–3, 2008. ISSN 546-1696. Disponível em: <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3013647/>>.

STROUSTRUP, B. **Standard C++**. [S.l.], 1998.

VASINEK, M.; PLATOS, J. Prediction and evaluation of zero order entropy changes in grammar-based codes. **Entropy**, v. 19, n. 5, 2017. ISSN 1099-4300. Disponível em: <<https://www.mdpi.com/1099-4300/19/5/223>>.

WAN VO NGOC ANH, K. A. R. Transformations for the compression of FASTQ quality scores of next-generation sequencing data. **Bioinformatics**, Bioinformatics, v. 28, n. 5, p. 1–2, mar. 2012. ISSN 628635. Disponível em: <<https://academic.oup.com/bioinformatics/article/28/5/628/246901>>.

WATSON, J. D.; CRICK, F. H. C. Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. **Nature**, Nature, Cavendish Laboratory, Cambridge., v. 171, n. 4356, p. 1–2, abr. 1953. ISSN 737738. Disponível em: <<https://www.nature.com/articles/171737a0>>.