ANTONIO FERRÃO NETO

Predição Computacional de Sítios de
Ligação de Fatores de Transcrição Baseada
em Gramáticas Regulares Estocásticas

São Paulo

2017

ANTONIO FERRÃO NETO

**Predição Computacional de Sítios de Ligação de
Fatores de Transcrição Baseada em Gramáticas
Regulares Estocásticas**

Versão original

Texto de Exame de Mestrado apresentado ao Instituto de Matemática e Estatística como parte dos requisitos para obtenção do título de Mestre em Ciências pelo Programa de Mestrado em Bioinformática.

Área de concentração: Bioinformática - Interunidades

Orientador: Profa. Dra. Arianne Machado Lima

Coorientador: Prof. Dr. Luiz Paulo Moura Andrioli

UNIVERSIDADE DE SÃO PAULO
PROGRAMA INTERUNIDADES DE PÓS-GRADUAÇÃO EM BIOINFORMÁTICA

São Paulo

2017

FICHA CATALOGRÁFICA

F373p Ferrão Neto, Antonio
Predição computacional de sítios de ligação de fatores de transcrição baseada em gramáticas regulares estocásticas / Antonio Ferrão Neto, [orient.] Ariane Machado Lima, [coorient.] Luiz Paulo Moura Andrioli. São Paulo : 2017.
161 p.

Dissertação (Mestrado) - Universidade de São Paulo
Orientadora: Profa. Dra. Ariane Machado Lima
Coorientador: Prof. Dr. Luiz Paulo Moura Andrioli

Programa Interunidades de Pós-Graduação em Bioinformática
Área de concentração: Bioinformática

1. Bioinformática. 2. Gramáticas formais por computador. 3. Sítios de ligação.
I. Lima, Ariane Machado, orientadora. II. Andrioli, Luiz Paulo Moura, coorientador.
III. Universidade de São Paulo. IV. Título.

CDD - 572.8

Texto de Mestrado de autoria de Antonio Ferrão Neto, sob o título “**Predição Computacional de Sítios de Ligação de Fatores de Transcrição Baseada em Gramáticas Regulares Estocásticas**”, apresentada ao Instituto de Matemática e Estatística da Universidade de São Paulo, para obtenção do título de Mestre em Bioinformática, aprovada em ____ de _____ de _____ pela comissão julgadora constituída pelos doutores:

Prof. Dr. _____

Instituição: _____

Presidente

Prof. Dr. _____

Instituição: _____

Prof. Dr. _____

Instituição: _____

Prof. Dr. _____

Instituição: _____

À minha querida mãe Ruth (in memoriam).

Agradecimentos

Agradeço primeiramente a Deus, responsável pela minha própria existência e por tudo o que eu possuo. Agradeço à minha orientadora e amiga Profa. Dra. Arianne Machado Lima e ao meu coorientador e amigo Prof. Dr. Luiz Paulo Moura Andrioli. Agradeço à minha família, especialmente ao meu irmão, Dr. Leônidas, à minha irmã Sônia e ao seu marido, Dr. Samuel, que têm me ajudado financeiramente e à minha filha Ketelly, principalmente pela sua compreensão em relação à minha ausência para a minha dedicação nas pesquisas.

“The development of an organism ... may be considered as the execution of a ‘developmental program’ present in the fertilized egg. ... A central task of developmental biology is to discover the underlying algorithm from the course of development.”

(Aristid Lindenmayer)

Resumo

FERRÃO, Antonio Ferrão Neto. **Predição Computacional de Sítios de Ligação de Fatores de Transcrição Baseada em Gramáticas Regulares Estocásticas**. 2017. 161 f. Texto de Dissertação de Mestrado em Bioinformática – Instituto de Matemática e Estatística da Universidade de São Paulo, Bioinformática, São Paulo, 2017.

Fatores de transcrição (FT) são proteínas que se ligam em sequências específicas e bem conservadas de nucleotídeos no DNA, denominadas *sítios de ligação dos fatores de transcrição* (SLFT), localizadas em regiões de regulação gênica conhecidas como *módulos cis-reguladores* (CRM). Ao reconhecer o SLFT, o fator de transcrição se liga naquele sítio e influencia a transcrição gênica positiva ou negativamente. Existem técnicas experimentais para a identificação dos locais dos SLFTs em um genoma, como *footprinting*, *ChIP-chip* ou *ChIP-seq*. Entretanto, a execução de tais técnicas implica em custos e tempo elevados. Alternativamente, pode-se utilizar as sequências de SLFTs já conhecidas para um determinado fator de transcrição e aplicar técnicas de aprendizado computacional supervisionado para criar um modelo computacional para tal sítio e então realizar a predição computacional no genoma. Entretanto, a maioria das ferramentas computacionais existentes para esse fim considera independência entre as posições entre os nucleotídeos de um sítio - como as baseadas em PWMs (*position weight matrix*) - o que não é necessariamente verdade. Este projeto teve como objetivo avaliar a utilização de *gramáticas regulares estocásticas* (GRE) como técnica alternativa às PWMs neste problema, uma vez que GREs são capazes de caracterizar dependências entre posições consecutivas dos sítios. Embora as diferenças de desempenho tenham sido sutis, GREs parecem mesmo ser mais adequadas do que PWMs na presença de valores mais altos de dependência de bases, e PWMs nos demais casos. Por fim, uma ferramenta de predição computacional de SLFTs foi criada baseada tanto em GREs quanto em PWMs.

Palavras-chaves: gramáticas regulares; PWM; fator de transcrição; sítios de ligação de fatores de transcrição; *enhancer*; módulos cis-regulatórios; CRM; motivos.

Abstract

FERRÃO, Antonio Ferrão Neto. **Computational Prediction of Transcription Factor Binding Sites Based on Stochastic Regular Grammars**. 2017. 161 p. Master's Dissertation in Bioinformatics – Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2017.

Transcription factors (FT) are proteins that bind to specific and well-conserved sequences of nucleotides in the DNA, called *transcription factor binding sites* (TFBS), contained in regions of gene regulation known as *cis-regulatory modules* (CRM). By recognizing TFBS, the transcription factor binds to that site and positively or negatively influence the gene transcription. There are experimental procedures for the identification of TFBS in a genome such as **footprinting**, **ChIP-chip** or **ChIP-Seq**. However, the implementation of these techniques involves high costs and time. Alternatively, one may utilize the TFBS sequences already known for a particular transcription factor and applying computational supervised learning techniques to create a computational model for that site and then perform the computational prediction in the genome. However, most existing software tools for this purpose considers independence between nucleotide positions in the site - such as those based on PWMs (*position weight matrix*) - which is not necessarily true. This project aimed to evaluate the use of *stochastic regular grammars* (SRG) as an alternative technique to PWMs in this problem, since SRGs are able to characterize dependencies between consecutive positions in the sites. Although differences in performance have been subtle, SRGs appear to be more suitable than PWMs in the presence of higher base dependency values, and PWMs in other cases. Finally, a computational TFBS prediction tool was created based on both SRGs and PWMs.

Keywords: *regular grammars*; PWM; transcription factor; transcription factor binding sites; *enhancer*; *cis-regulatory modules*; CRM; motifs.

Lista de figuras

Figura 1 – Diversos níveis de controle gênico em uma célula eucariota.	22
Figura 2 – Fatores de transcrição e suas ligações nos sítios de ligação dos fatores de transcrição em módulos reguladores em um gene.	25
Figura 3 – Ligação dos Fatores de Transcrição ao DNA	26
Figura 4 – Exemplo da derivação de uma PWM.	31
Figura 5 – Exemplo de utilização de uma PWM para verificar se um dado sítio pode ou não ser considerado um SLFT.	33
Figura 6 – Árvore da gramática para o exemplo da Figura 4.	38
Figura 7 – Produções que definem a gramática representada pela árvore da Figura 6.	39
Figura 8 – Matriz de confusão.	41
Figura 9 – Espaço do gráfico ROC.	43
Figura 10 – Exemplo de gráfico ROC	45
Figura 11 – Validação cruzada.	48
Figura 12 – Fluxo resumizando o estágio da seleção.	53
Figura 13 – Visão geral do processo de geração das amostras de teste e treinamento para a validação cruzada.	68
Figura 14 – Exemplo de simulação de uma amostra artificialmente construída com 21 sequências de 6 bases e com 100% de dependência entre as bases. . .	76
Figura 15 – Exemplo de simulação de uma amostra artificialmente construída com 21 sequências de 6 bases e com as bases distribuídas de maneira totalmente aleatória.	77
Figura 16 – Escores calculados para simulações de 11 amostras artificialmente cons- truídas com diferentes níveis de dependência, de 100% a 0%.	78
Figura 17 – Gráfico mostrando o número de arquivos para cada quantidade de sequências encontradas no banco de dados Jaspar 2016 , para todos os FTs, incluindo várias espécies.	80
Figura 18 – Mesmo gráfico da Figura 17, mas mostrando apenas os números de arquivos com quantidades de sequências inferiores a 101.	80
Figura 19 – Faixas de escores de níveis de dependência calculados para dados reais do banco Jaspar 2016 em relação aos dados das simulações produzidas artificialmente.	81

Figura 20 – Área sob a curva ROC média sobre todas as amostras	83
Figura 21 – Área sob a curva ROC média sobre todas as amostras (<i>zoom</i>)	84
Figura 22 – Área sob a curva ROC (AUC) por Escore de Dependência. A tabela abaixo do gráfico mostra as percentagens de dependência para cada faixa de escore.	85
Figura 23 – Área sob a curva ROC (AUC) por tamanho do SLFT	86
Figura 24 – Gráfico da Precisão (critério ROC01) por tamanho do SLFT, para os três modelos nulos, PWM e GRE.	86
Figura 25 – Área sob a curva ROC (AUC) por tamanho da amostra	88
Figura 26 – Gráfico da Precisão (critério ROC01) por tamanho da amostra, para os três modelos nulos, PWM e GRE.	89
Figura 27 – Correlações entre o escore de dependência e a AUC, para os três modelos nulos (N1, N2 e N3), tanto para as GREs quanto para as PWMs.	90
Figura 28 – Correlações entre os escores de dependência e as medidas de desempenho, para os três modelos nulos, tanto para as GREs quanto para as PWMs, para os dois critérios de limiar adotados (ROC01 e Youden).	91
Figura 29 – Gráfico de dispersão dos pontos relacionando o escore de dependência e a medida AUC da gramática (GRE) para o modelo N1.	114
Figura 30 – Exemplo mostrando o formato do arquivo de saída da validação cruzada para a PWM.	119
Figura 31 – Exemplo mostrando o formato do arquivo de saída da validação cruzada para a GRE.	120
Figura 32 – Gráfico da Sensibilidade (ou Taxa de Verdadeiros Positivos-TPR) (critério ROC01) por tamanho do SLFT, para os três modelos nulos, PWM e GRE.	121
Figura 33 – Gráfico da Sensibilidade (ou Taxa de Verdadeiros Positivos-TPR) (critério Youden) por tamanho do SLFT, para os três modelos nulos, PWM e GRE.	121
Figura 34 – Gráfico da Especificidade (1-FPR) (critério ROC01) por tamanho do SLFT, para os três modelos nulos, PWM e GRE.	122
Figura 35 – Gráfico da Especificidade (1-FPR) (critério Youden) por tamanho do SLFT, para os três modelos nulos, PWM e GRE.	122

Figura 36 – Gráfico da Precisão (critério ROC01) por tamanho do SLFT, para os três modelos nulos, PWM e GRE.	123
Figura 37 – Gráfico da Precisão (critério Youden) por tamanho do SLFT, para os três modelos nulos, PWM e GRE.	123
Figura 38 – Gráfico da Medida-F (critério ROC01) por tamanho do SLFT, para os três modelos nulos, PWM e GRE.	124
Figura 39 – Gráfico da Medida-F (critério Youden) por tamanho do SLFT, para os três modelos nulos, PWM e GRE.	124
Figura 40 – Gráfico do Erro Global (critério ROC01) por tamanho do SLFT, para os três modelos nulos, PWM e GRE.	125
Figura 41 – Gráfico do Erro Global (critério Youden) por tamanho do SLFT, para os três modelos nulos, PWM e GRE.	125
Figura 42 – Gráfico da Sensibilidade (ou Taxa de Verdadeiros Positivos - TPR) (critério ROC01) por tamanho da amostra, para os três modelos nulos, PWM e GRE.	126
Figura 43 – Gráfico da Sensibilidade (ou Taxa de Verdadeiros Positivos - TPR) (critério Youden) por tamanho da amostra, para os três modelos nulos, PWM e GRE.	126
Figura 44 – Gráfico da Especificidade (1-FPR) (critério ROC01) por tamanho da amostra, para os três modelos nulos, PWM e GRE.	127
Figura 45 – Gráfico da Especificidade (1-FPR) (critério Youden) por tamanho da amostra, para os três modelos nulos, PWM e GRE.	127
Figura 46 – Gráfico da Precisão (critério ROC01) por tamanho da amostra, para os três modelos nulos, PWM e GRE.	128
Figura 47 – Gráfico da Precisão (critério Youden) por tamanho da amostra, para os três modelos nulos, PWM e GRE.	128
Figura 48 – Gráfico da Medida-F (critério ROC01) por tamanho da amostra, para os três modelos nulos, PWM e GRE.	129
Figura 49 – Gráfico da Medida-F (critério Youden) por tamanho da amostra, para os três modelos nulos, PWM e GRE.	129
Figura 50 – Gráfico do Erro Global (critério ROC01) por tamanho da amostra, para os três modelos nulos, PWM e GRE.	130

Figura 51 – Gráfico do Erro Global (critério Youden) por tamanho da amostra, para os três modelos nulos, PWM e GRE.	130
Figura 52 – Gráfico da Sensibilidade (ou Taxa de Verdadeiros Positivos - TPR) (critério ROC01) por faixas de Escore de Dependência, para os três modelos nulos, PWM e GRE.	131
Figura 53 – Gráfico da Sensibilidade (ou Taxa de Verdadeiros Positivos - TPR) (critério Youden) por faixas de Escore de Dependência, para os três modelos nulos, PWM e GRE.	131
Figura 54 – Gráfico da Especificidade (1-FPR) (critério ROC01) por tamanho da amostra, para os três modelos nulos, PWM e GRE.	132
Figura 55 – Gráfico da Especificidade (1-FPR) (critério Youden) por faixas de Escore de Dependência, para os três modelos nulos, PWM e GRE.	132
Figura 56 – Gráfico da Precisão (critério ROC01) por faixas de Escore de Dependência, para os três modelos nulos, PWM e GRE.	133
Figura 57 – Gráfico da Precisão (critério Youden) por faixas de Escore de Dependência, para os três modelos nulos, PWM e GRE.	133
Figura 58 – Gráfico da Medida-F (critério ROC01) por tamanho da amostra, para os três modelos nulos, PWM e GRE.	134
Figura 59 – Gráfico da Medida-F (critério Youden) por faixas de Escore de Dependência, para os três modelos nulos, PWM e GRE.	134
Figura 60 – Gráfico do Erro Global (critério ROC01) por faixas de Escore de Dependência, para os três modelos nulos, PWM e GRE.	135
Figura 61 – Gráfico do Erro Global (critério Youden) por faixas de Escore de Dependência, para os três modelos nulos, PWM e GRE.	135
Figura 62 – Exemplo de utilização de uma janela deslizante.	136
Figura 63 – Arquivo de entrada <code>lista_gen_freq.txt</code>	138
Figura 64 – Trechos do arquivo de entrada <code>tf_parameters.txt</code>	139
Figura 65 – Trechos do arquivo de entrada <code>gre_tool_parameters.txt</code>	140
Figura 66 – Exemplo de GRE inferida pelo <i>framework</i> GrammarLab	141
Figura 67 – Exemplo de PWM inferida durante a validação cruzada	142
Figura 68 – Exemplo de arquivo de parâmetros (" <code>exe_parameters.txt</code> ").	142
Figura 69 – Exemplo de trecho de arquivo de saída (" <code>found_tfbs.txt</code> ")	148

Lista de tabelas

Tabela 1 – Síntese dos artigos analisados.	54
Tabela 2 – Validação Cruzada Clusterizada	67
Tabela 3 – Relação de espécies utilizadas no banco de dados Jaspar 2016 e suas respectivas frequências genômicas	71
Tabela 4 – Número Sequências por Quantidades de Arquivos - Jaspar 2016	81
Tabela 5 – Resultados das métricas de desempenho médias do Jaspar 2014 (An- tigo) e os resultados das métricas de desempenho ponderadas para todos as 341 amostras utilizadas do Jaspar 2016 (Novo).	92
Tabela 6 – Escore de dependência de dinucleotídeos dos arquivos do Jaspar 2016 juntamente com arquivos "dep_xxx" criados artificialmente.	109
Tabela 6 – (continuação)	110
Tabela 6 – (continuação)	111
Tabela 6 – (continuação)	112
Tabela 6 – (continuação)	113
Tabela 7 – Resultados dos cálculos dos coeficientes de correlação de Pearson e o seu respectivo p-valor para cada medida de desempenho.	115

Lista de abreviaturas e siglas

AUC	<i>Area Under the Curve</i> - Área sob a curva. Essa área sob a curva ROC é muito utilizada para medir o desempenho dos classificadores.
BNF	<i>Backus Normal Form</i> - Metassintaxe usada para expressar gramáticas. Neste trabalho, “BNF” define o formato dos arquivos que contém as gramáticas inferidas pelo GrammarLab.
FASTA	<i>Formato de arquivos</i> - Formato de arquivos contendo dados de sequências de DNA.
GRE	Gramática Regular Estocástica.
<i>Motif</i>	O mesmo que motivo.
Motivo	Padrão de sequências de nucleotídeos relativamente bem conservadas no DNA, mas que permite algumas variações.
N1, N2 e N3	<i>Modelos Nulos</i> - Modelos nulos adotados para o cálculo da frequência de fundo do DNA.
PCM	<i>Position Count Matrix</i> - contagem inteira que descreve o número de vezes em que cada nucleotídeo foi observado em cada posição.
PFM	<i>Position Frequency Matrix</i> - cada coluna totaliza o valor 1.0, com a representação da frequência fracional de cada base, em cada posição da matriz. PCMs podem ser convertidas para PFMs dividindo cada coluna pela soma da contagem de nucleotídeos nesta coluna.
PWM	<i>Position Weight Matrix</i> - apresenta cada frequência observada em relação a um modelo de <i>background</i> da distribuição dos nucleotídeos. Os valores zerados são considerados como valores próximos a zero e os valores são expressos em termos de <i>log-odds</i> .
ROC01	<i>Critério ROC 01</i> - Critério para a obtenção de um limiar ideal na curva ROC.

ROC	<i>Receiver Operating Characteristics</i> - Gráficos ROC são gráficos bidimensionais nos quais a taxa de verdadeiros positivos (TVP) é plotada no eixo vertical (ordenada) e a taxa de falsos positivos (TFP) é plotada no eixo horizontal (abscissa).
SLFT	Sítio de Ligação de Fatores de Transcrição - trechos de DNA, usualmente com comprimento de 5 a 30 pb, onde os Fatores de Transcrição (FT) usualmente se ligam.
SRG	<i>Stochastic Regular Grammar</i> - o mesmo que GRE.
TFBS	<i>Transcription Factor Binding Site</i> - o mesmo que SLFT.
TF	<i>Transcription Factor</i> - Fatores de Transcrição são proteínas que se ligam em sítios específicos no DNA e influenciam na Regulação da Transcrição Gênica, positivamente ou negativamente.
Youden	<i>Critério Youden</i> - Outro critério para a obtenção de um limiar ideal na curva ROC.

Lista de símbolos

N	Conjunto de símbolos não-terminais.
Σ	conjunto de símbolos terminais da gramática.
R	o conjunto de regras de substituições ou produções da gramática.
S	o símbolo inicial da gramática.
P	função $P : R \rightarrow [0, 1]$ que associa a cada produção em R uma probabilidade p .

Sumário

1	Introdução	21
2	Conceitos Fundamentais	24
2.1	Fatores de Transcrição	24
2.2	Técnicas experimentais de identificação de sítios de ligação dos fatores de transcrição (SLFT)	26
2.3	Motivo e sequência consenso	28
2.4	<i>Position Weight Matrix</i> (PWM)	29
2.4.1	Derivação das PWMs	29
2.4.2	Cálculo do escore de uma PWM sobre um sítio no DNA	32
2.5	Dependência entre as bases	33
2.6	Gramáticas Regulares Estocásticas	35
2.6.1	Derivação de GREs	36
2.6.2	Cálculo do escore de uma GRE sobre um sítio no DNA	39
2.7	Desempenho de um classificador	40
2.7.1	Gráficos ROC	43
2.7.2	Critérios para a escolha de um limiar ótimo para classi- ficação	45
2.7.3	Validação cruzada	47
3	Objetivo	49
3.1	Objetivo Geral	49
3.2	Objetivos Específicos	49
4	Revisão Bibliográfica Sistemática	50
4.1	Materiais e Métodos da Revisão Sistemática	50
4.1.1	Planejamento	51
4.1.2	Condução e Extração de Dados	52
4.2	Síntese da Extração de Dados	54
4.2.1	Análise Global	57
4.2.2	Melhorias na predição dos SLFTs	58

4.3	Discussões da Revisão Sistemática	62
4.4	Conclusão da Revisão Sistemática	63
5	Materiais e Métodos	64
5.1	Criação da base de amostras de SLTFs	64
5.1.1	Obtenção das amostras positivas	64
5.1.2	Amostra Negativa	66
5.2	Preparação de amostras de treinamento e teste para validação cruzada baseada em agrupamentos	66
5.2.1	Cálculo das similaridades entre as sequências e agrupa- mentos	69
5.2.2	Preparação dos agrupamentos para teste	70
5.3	Comparação entre PWMs e GREs na identificação de sítios de ligação de fatores de transcrição	70
5.3.1	Modelos nulos para o teste das PWMs e GREs	70
5.3.2	Processamento das validações cruzadas com as PWMs e GREs	71
5.3.3	O gráfico ROC e a área sob a curva ROC (AUC)	72
5.4	Determinação do nível de dependência existente entre as bases	74
6	Resultados e Discussões	79
6.1	Criação da base de amostras positivas de SLFTs	79
6.1.1	Escore de dependência dos dados do banco Jaspar 2016	79
6.2	Comparação de desempenho de PWMs e GREs	82
6.2.1	Desempenhos médios gerais	82
6.2.2	Desempenhos médios por faixa de dependência	83
6.2.3	Desempenhos médios por tamanho de sítio	84
6.2.4	Desempenhos médios por faixa de tamanho de amostra	87
6.3	Correlação entre as medidas de desempenho e o escore de dependência	87
6.4	Comparação com resultados anteriores	90
6.5	Ferramenta de Predição de SLFTs	92

6.6	Considerações finais	93
7	Conclusão	96
7.1	Principais contribuições deste trabalho	96
7.2	Trabalhos futuros	99
	Referências ¹	101
	APÊNDICES	107
	Apêndice A–Escore de dependência por arquivo	108
	Apêndice B–Correlações entre escores de dependência e as medidas de desempenho	114
	Apêndice C–Códigos	116
	Apêndice D–Exemplo de formato de saída da va- lidação cruzada	118
	Apêndice E–Gráficos das medidas de desempenho	121
E.1	Gráficos das medidas de desempenho por tamanho do SLFT	121
E.2	Gráficos das medidas de desempenho por tamanho da amostra	126
E.3	Gráficos das medidas de desempenho por faixas de Escore de Dependência	131
	Apêndice F–Ferramenta de predição computaci- onal baseada em gramáticas regula- res estocáticas	136
F.1	A ideia central da ferramenta de predição	136
F.2	GREs, PWMs e arquivos auxiliares	137
F.3	Parâmetros do programa de predição de SLFTs	143
F.4	Detalhes sobre o programa de predição de SLFTs	144

¹ De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

F.5	Como executar o programa de predição de SLFTs . . .	161
------------	--	------------

1 Introdução

Nos últimos anos, tem havido um grande esforço da comunidade científica em todo o mundo no sentido de interpretar a grande quantidade de informação que os sequenciadores de DNA obtêm a um custo cada vez mais reduzido. Atualmente, os maiores esforços se concentram no entendimento do seu funcionamento, por exemplo, entender de que forma os genes interagem entre si constituindo uma rede integrada e coerente. Um passo fundamental para a compreensão desses mecanismos é conhecer os mecanismos da Regulação Gênica.

A Figura 1 apresenta diversos níveis conhecidos para a regulação gênica¹ (ALBERTS et al., 2009). E dentre todos esses níveis, um dos mais importantes é o controle transcricional, que é o controle sobre a transcrição para a geração do RNA.

Este trabalho está relacionado com o mecanismo de início da transcrição nos eucariotos. Portanto, trata do controle transcricional, que é regido por proteínas reguladoras conhecidas como *fatores de transcrição* (FTs).

Os FTs são proteínas que medeiam o processo transcricional ligando-se em trechos específicos do DNA. Esses trechos de sequências de DNA onde os FTs se ligam são conhecidos por sítios de ligação de fatores de transcrição (SLFTs). Esses SLFTs possuem tipicamente de cinco a trinta nucleotídeos e representam sequências de nucleotídeos relativamente bem conservadas, mas que permitem algumas variações (STEWART; HANNENHALLI; PLOTKIN, 2012). Esses trechos de sequências são conhecidos como *motivos* (D'HAESELEER, 2006). Os SLFTs fazem parte dos módulos cis-reguladores da transcrição dos genes, também conhecidos por *enhancers* (LEON; DAVIDSON, 2009). Ao se ligarem nos SLFTs, os FTs modulam a expressão gênica, podendo estimular ou inibir o processo da transcrição, interagindo com o complexo transcricional. Os SLFTs normalmente aparecem próximos uns aos outros, formando agrupamentos chamados *clusters*, constituindo os módulos cis-regulatórios (CRMs - *cis-regulatory modules*)

Existem ferramentas computacionais para a predição de SLFTs. A técnica mais utilizada para a caracterização dos motivos é a PWM, mas ela ignora dependências entre as bases. Uma técnica alternativa, que leva em consideração a dependência entre as bases, são as *gramáticas regulares estocásticas* (GRE). Assim, este trabalho teve como objetivo comparar essas duas técnicas.

¹ Nesta Figura 1 estão citados apenas alguns controles, podendo ainda haver outros já conhecidos e ainda outros a serem descobertos.

Figura 1 – Diversos níveis de controle gênico em uma célula eucariota. A regulação pode ocorrer (a) na ativação e desativação cromossômica; (b) na transcrição da informação contida no DNA para o RNA transcrito; (c) no processamento do RNA transcrito: splicing, adição de CAP-5' e da cauda Poli-A; (d) durante o transporte do RNAm para o citosol: ação de ribonucleases e RNA de interferência; (e) na tradução do RNAm em proteína: regulado pelos fatores de iniciação da tradução e (f) na ativação da proteína recém sintetizada: dobramento auxiliado pelas chaperonas, fosforilação, acetilação, alquilação, formação de pontes dissulfeto, clivagem, glicosilação, sulfatação, metilação, isoprenilação e ubiquitinação.

Fonte: Antonio Ferrão Neto, 2017

O restante deste documento está organizado da seguinte forma: o capítulo 2 descreve os principais conceitos necessários para o entendimento desse trabalho; o capítulo 3 define mais detalhadamente os objetivos propostos; o capítulo 4 traz uma revisão sistemática da literatura acerca das ferramentas computacionais de predição de SLFTs; o capítulo 5 aborda os materiais e métodos utilizados para a execução deste projeto; o capítulo 6 traz os resultados obtidos e uma discussão dos mesmos e, finalmente, o capítulo 7 faz uma conclusão do trabalho. Além disso, os apêndices contém as seguintes informações: o apêndice A apresenta os escores de dependência por arquivo **Jaspar 2016**; o apêndice B apresenta os resultados numéricos das correlações entre os escores de dependência e as medidas de desempenho; o apêndice C apresenta alguns trechos de código de exemplo; o apêndice D apresenta um exemplo do arquivo de saída das validações cruzadas e o apêndice E apresenta os gráficos das medidas de desempenho, tanto em função dos tamanhos dos SLFTs, quanto em função das faixas de tamanho de amostras.

2 Conceitos Fundamentais

Esta seção trata de alguns conceitos fundamentais, necessários para o entendimento do projeto.

2.1 Fatores de Transcrição

Os fatores de transcrição (FTs) são proteínas que afetam a transcrição do gene por meio de suas ligações ao DNA em regiões conservadas de sequências de bases no DNA, conhecidas como *Sítios de Ligação de Fatores de Transcrição* (SLFT). A ligação dos FTs aos respectivos SLFTs afeta a frequência da transcrição do gene, aumentando ou diminuindo a quantidade do produto expresso nos diferentes tecidos, em diferentes fases do desenvolvimento do organismo.

A Figura 2 mostra esquematicamente, para um gene genérico “X”, os elementos que participam da transcrição. Os FTs se ligam ao DNA tanto a montante quanto a jusante e, embora o desenho não mostre, é possível também a ligação dos FTs em sítios que se localizam nos *introns* do gene em questão (LI et al., 2012).

Os Fatores Gerais da Transcrição, diferentemente dos FTs, ligam-se na região promotora do gene (ALBERTS et al., 2009) e sempre participam da formação do Complexo Transcricional, juntamente com a enzima RNA polimerase II (RNAPol II)¹. Na ausência dos Fatores Gerais de Transcrição, nos organismos eucariotos, não há como haver transcrição.

¹ Os núcleos das células eucariotas possuem três tipos de RNA-polimerases: RNA-polimerase I, RNA-polimerase II e RNA-polimerase III. As três polimerases são estruturalmente similares entre si e compartilham algumas subunidades, mas transcrevem diferentes tipos de genes. As RNAPolimerases I e III transcrevem os genes que codificam o tRNA, o rRNA e vários pequenos RNAs. A RNAPolimerase II transcreve a grande maioria dos genes, inclusive todos aqueles que codificam proteínas (ALBERTS et al., 2009).

Os módulos cis-reguladores², que estão representados na Figura 2 constituem-se de regiões no DNA onde se verificam agrupamentos de SLFTs. Ou seja, são regiões onde diversos FTs se ligam, eventualmente competindo pelo mesmo SLFT e estabelecendo a dinâmica regulatória do gene.

Figura 2 – Fatores de transcrição e suas ligações nos sítios de ligação dos fatores de transcrição em módulos reguladores em um gene.

Fonte: (ALBERTS et al., 2009), adaptado por Antonio Ferrão Neto, 2017

Os FTs podem se ligar a SLFTs que estão próximos ou muito distantes do início da transcrição do gene, podendo estar a milhares de pares de bases do ponto de início da transcrição. A Figura 3 mostra a hipótese que explica como os FTs podem agir à distância sobre o complexo transcricional: a molécula de DNA se dobra formando laços, permitindo assim que os FTs, que estão a muitos pares de bases da região promotora, interajam com o complexo transcricional.

² Muitos autores utilizam a expressão *enhancers* como sinônimo de módulo cis-reguladores e consideram que os *enhancers* possam conter sítios de ligação para proteínas repressoras. Por exemplo, “Many enhancers also contain binding sites for sequence-specific repressors, which exclude expression in inappropriate tissues” (LEVINE, 2010). Por outro lado, alguns autores fazem a distinção entre *enhancers* e outros tipos de elementos reguladores como, por exemplo, “[...] almost exclusive focus of regulatory mutations on distal enhancers reflects our inability to assay functionally other types of regulatory elements in the genome. Yet other classes of regulatory elements, such as insulators, repressors and matrix attachment regions [...]”(PENNACCHIO et al., 2013). Já outros autores consideram que os elementos cis-reguladores compreendem os “Promoters”, “Enhancers”, “Silencers”, “Insulators” e outros tipos de regiões regulatórias: “locus control regions” (LCRs) e “matrix attachment regions” (MARs) (NARLIKAR; OVCHARENKO, 2009). Neste trabalho, consideramos que os módulos cis-reguladores não incluem as regiões promotoras (“Promoters”).

Figura 3 – A molécula de DNA se dobra, permitindo que os Fatores de Transcrição influenciem o Complexo de Transcrição, mesmo estando a milhares de bases distantes do início da transcrição.

Fonte: (ALBERTS et al., 2009), adaptado por Antonio Ferrão Neto, 2017

2.2 Técnicas experimentais de identificação de sítios de ligação dos fatores de transcrição (SLFT)

Os locais no DNA onde se encontram os SLFTs podem ser identificados utilizando técnicas experimentais que investigam a interação entre proteínas específicas e a molécula de DNA, como por exemplo, **footprinting**, **ChIP-chip** e **ChIP-seq**. Neste caso, as proteínas específicas são os FTs e as regiões genômicas específicas de nosso interesse são os SLFTs.

Na técnica de **footprinting**, as amostras são amplificadas por reação em cadeia da polimerase (*Polymerase Chain Reaction* - PCR) e marcada radioativamente ou por fluorescência uma das extremidades do DNA³. As amostras são misturadas (*in vitro*) a uma solução contendo a proteína de interesse, que irá se ligar ao DNA (ligações covalentes cruzadas por formaldeído). Em seguida, adiciona-se uma enzima que cliva o DNA, por exemplo, a enzima DNase I, que tem um pequeno grau de especificidade, cortando o DNA uniformemente (alternativamente, pode-se utilizar a técnica de sonicação para fragmentar o DNA), exceto na região exata onde a proteína está ligada, pois a DNase I não terá acesso ao DNA no sítio onde a proteína está ligada. Posteriormente, as proteínas são

³ O protocolo especifica que a marcação deve ser realizada em apenas uma das extremidades (3' ou 5') para que os fragmentos sejam produzidos com diferentes tamanhos em relação a apenas uma das extremidades. E a marcação é feita para que possam, posteriormente, sensibilizar um filme em uma radiografia do gel de poliacrilamida. A adição deste marcador em uma das extremidades é anterior à adição da proteína para que sejam criadas duas soluções: uma com a proteína e outra sem, para comparação posterior.

retiradas dos fragmentos de DNA e esses fragmentos são separados por eletroforese em gel de poliacrilamida. As bandas resultantes são visualizadas por radiografia e uma interrupção no padrão de bandas indica o local de proteção por parte da proteína, isto é, o local onde a proteína se encontra ligada e o DNA está protegido do corte por esta enzima. Este local corresponde ao que se chama *pegada* ou *footprinting* da proteína sobre uma determinada sequência de DNA.

Nas técnicas baseadas em Imunoprecipitação da Cromatina (*Chromatin Immunoprecipitation* ou *ChIP*), os FTs associados ao DNA são temporariamente ligados covalentemente, formando um complexo proteína-DNA, dentro das células ou tecidos vivos. As células são lisadas e o DNA é mecanicamente fragmentado, por exemplo, por sonicação, em trechos com comprimentos de cerca de 500 bp. Estes fragmentos conterão trechos de DNA com e sem as proteínas associadas. Esses fragmentos recebem anticorpos (imunoglobulinas) especificamente direcionados para as proteínas alvos de estudo. Os fragmentos que contenham as proteínas com os anticorpos a elas ligadas serão precipitados e assim esse material é purificado, isolando-se apenas os trechos de DNA que contenham as proteínas ligadas ao DNA e estas aos anticorpos. O produto da imunoprecipitação poderá então ser analisado pela técnica ChIP-chip, que utiliza microarranjos (*microarrays* ou *chips*) ou pela técnica ChIP-seq onde esses trechos de DNA são simplesmente sequenciados em “sequenciadores de próxima geração” (*next generation sequencers*). Estes métodos permitem então a identificação dos sítios de ligação no genoma que são ocupados *in vivo* pelos fatores de transcrição.

A execução dessas técnicas, porém, implica em custos elevados, além de demandar tempo. Uma alternativa a essas técnicas experimentais é utilizar as sequências de SLFTs já conhecidas para um determinado FT (obtidas, por exemplo, pelas técnicas experimentais mencionadas acima e disponibilizadas em bases de dados) e aplicar técnicas de aprendizado computacional supervisionado para criar um modelo para os sítios de tal FT e então realizar a predição computacional dos sítios no genoma de interesse. Estes modelos podem ser baseados, por exemplo, em matrizes PWMs ou gramáticas regulares estocásticas, modelos estes comparados neste trabalho e descritos brevemente nas próximas seções.

SELEX (*Systematic Evolution of Ligands by Exponential Enrichment*) *Evolução Sistemática de Ligantes por Enriquecimento Exponencial* (GOLD et al., 1995), é uma técnica *in vitro* para produção de fragmentos de DNA de dupla ou simples fita ou RNA que se ligam a um ou mais ligantes alvos, que no escopo deste texto, são os fatores de transcrição.

Inicialmente são sintetizados fragmentos de DNA, compondo uma imensa biblioteca de sequências de comprimento fixo, contendo bases aleatoriamente geradas, delimitadas por terminadores constantes nas extremidades 5' e 3', que servem como iniciadores (*primers*). Essa solução, composta por essa enorme quantidade de fragmentos de DNA, é então misturada com o fator de transcrição que irá se ligar aos sítios que eventualmente existam entre esse imenso número de sequências. A quantidade de possíveis diferentes sequências em um trecho de DNA de comprimento n é 4^n , ou seja, 4 possibilidades A, C, G e T em cada posição. Uma vez que a quantidade de sequências é muito grande, é possível que o fator de transcrição encontre sítios onde ele se ligue a esses fragmentos. As sequências que não se ligam são então removidas, usualmente por cromatografia de afinidade. Os fragmentos que se ligaram passam por um processo de eluição⁴ e são a seguir amplificados por PCR (Reação em Cadeia da Polimerase), preparando os fragmentos para o próximo ciclo, no qual todo esse ciclo se repete mas com condições de lavagem mais restritivas para identificar as sequências de ligações mais fortes.

2.3 Motivo e sequência consenso

O motivo representa um padrão que os SLFTs precisam ter para que os FTs apresentem afinidade ao sítio e se liguem a ele. Isso não significa que os SLFTs tenham que ser sempre idênticos, com todas as suas bases sendo sempre iguais, para que os FTs reconheçam esses sítios. Existe uma certa variação aceitável nas bases que compõem os Motivos e essa variação é o que define o Motivo e ela depende do FT. Então, para cada FT existe um Motivo (D'HAESELEER, 2006).

Uma das formas de se representar um motivo é por meio de uma sequência consenso. Uma sequência consenso mostra quais bases são conservadas e quais bases são variáveis. Por exemplo, a sequência consenso T[AC]RY{G}N define um Motivo de seis bases no qual o primeiro T significa que um T é encontrado nesta primeira posição, [AC] significa que ou um A ou um C é encontrado na segunda posição, R significa que qualquer purina (A ou G) pode ser encontrada nesta terceira posição, Y significa que qualquer pirimidina (T ou C) pode ser encontrada nesta quarta posição, {G} significa que podem ser encontradas quaisquer

⁴ Eluição é um processo utilizado para lavar os componentes de uma mistura através de uma coluna de cromatografia, consistindo na remoção de um material adsorvido existente em um adsorvente lavando-o em um líquido. Neste caso, o material removido são os fatores de transcrição que estavam ligados aos fragmentos de DNA.

bases nesta quinta posição, exceto um G e um N significa que podem ser encontradas quaisquer bases nesta sexta posição. Nesta notação, [AC] não permite que sejam fornecidas as frequências relativas de A e C.

2.4 *Position Weight Matrix* (PWM)

A maneira mais frequente de se caracterizar ou representar um Motivo (seção 2.3) é por meio de uma *Position Weight Matrix* (PWM).

As PWMs foram criadas pelo geneticista Gary Stormo (STORMO et al., 1982) como uma alternativa às sequências consenso. Como descrito na próxima seção, as PWMs não só representam as frequências relativas de cada nucleotídeo em cada posição como também levam em consideração um modelo nulo que representa a distribuição dos nucleotídeos fora do Motivo. Nesta distribuição, a soma das probabilidades para cada nucleotídeo (q_α) deve que ser igual a um, ou seja:

$$\sum_{\alpha \in \{A,T,C,G\}} q_\alpha = 1 \quad (1)$$

Exemplos de modelos nulos são o modelo nulo uniforme, no qual todos os nucleotídeos possuem a mesma probabilidade, e o modelo nulo baseado em frequência genômica, no qual q_α é a frequência relativa do nucleotídeo α no genoma da espécie em questão. Modelos nulos são utilizados não só em PWMs mas em vários modelos probabilísticos, explícita ou implicitamente no momento de tomada de decisão, a exemplo do descrito posteriormente na seção 2.6.2.

2.4.1 Derivação das PWMs

As PWMs são frequentemente derivadas de um conjunto de sequências alinhadas, supostamente relacionadas aos locais onde os FTs se ligam, ou seja, aos SLFTs (KULAKOVSKIY; MAKEEV, 2013). Essas sequências são chamadas de sequências de treinamento⁵.

⁵ O termo *amostra de treinamento* é amplamente utilizado na área de Aprendizado Computacional para designar os exemplos que são utilizados para treinar ou derivar um modelo.

A Figura 4 mostra como uma PWM pode ser derivada de um conjunto de sequências de fragmentos de DNA, obtidos, por exemplo, pela técnica da Imunoprecipitação da Cromatina (*ChIP - Chromatin Immunoprecipitation*) (seção 2.2).

A derivação da PWM, conforme exemplificado na Figura 4, ocorre da seguinte maneira: a) dados de fragmentos de DNA são obtidos por imunoprecipitação da cromatina (ChIP), onde se sabe, *a priori*, que existem sítios de ligação de um mesmo fator de transcrição nesses fragmentos; b) uma ferramenta de software para Descoberta de Motivos identifica nessas sequências subsequências similares, que seriam supostamente os sítios de ligação do referido fator; c) as sequências dos sítios descobertos são alinhadas com alinhamento múltiplo sem *gaps*; d) as sequências dos SLFTs podem ser representadas em uma matriz; e) as bases dos SLFTs são contadas em cada posição, produzindo-se uma matriz de contagens (PCM - *Position Count Matrix*); f) a matriz de frequências (PFM - *Position Frequency Matrix*) ou a matriz de estimativas de probabilidades (PPM - *Position Probability Matrix*) é calculada dividindo-se cada elemento da PCM pela soma de cada coluna; g) o Logo deste Motivo pode ser desenhado, onde o tamanho das letras é proporcional ao tamanho das frequências da PFM ou à entropia nesta distribuição e h) as frequências relativas são transformadas em pesos, que consideram também o modelo nulo sendo considerado, ou seja, a matriz PWM pode ser calculada simplesmente aplicando-se a equação 2 (LIFANOV et al., 2003) a cada elemento da PCM:

$$M_{\alpha,j} = \log_2 \left(\frac{x_{\alpha,j} + cq_{\alpha}}{(N + c)q_{\alpha}} \right) \quad (2)$$

na qual $M_{\alpha,j}$ é peso para o elemento na posição j do motivo para a letra α , sendo α a letra que representa um dos nucleotídeos A, T, C ou G, ou seja, $\alpha \in \{A, T, C, G\}$; N é o total de sequências da amostra; $x_{\alpha,j}$ é o elemento da matriz PCM para o nucleotídeo α na posição j ; q_{α} é a probabilidade do nucleotídeo α no modelo nulo considerado e c é um pseudocontador, que é um valor arbitrário utilizado para impedir que os valores das probabilidades inferidas a partir das contagens sejam iguais a zero. Os valores dos elementos de uma PWM podem ser negativos ou positivos, dependendo da razão entre a frequência relativa de um nucleotídeo no Motivo e sua probabilidade no modelo nulo. Devido à utilização da função *log*, se essa razão for entre 0 e 1, o valor será negativo, se for maior que 1, o valor será positivo.

Figura 4 – Exemplo da derivação de uma PWM.

A PWM pode se representada por uma matriz M $4 \times N$, sendo N é o tamanho do sítio, mas é também possível encontrar representações nas quais essa matriz é transposta. Este trabalho assume a primeira representação.

2.4.2 Cálculo do escore de uma PWM sobre um sítio no DNA

Uma vez que se tenha uma matriz PWM, pode-se utilizá-la para calcular a pontuação (escore) de uma sequência de nucleotídeos do DNA (um sítio ou candidato a sítio). Conforme a equação 3 (KULAKOVSKIY; MAKEEV, 2013), essa pontuação pode ser obtida pela soma dos valores dos correspondentes nucleotídeos (linhas) nas correspondentes posições (colunas) da matriz PWM em relação ao sítio fornecido. Ou seja, dada uma matriz PWM M , o escore de uma sequência s de tamanho l é:

$$Escore(s|M) = \sum_{j=1}^l M[s[j], j] \quad (3)$$

sendo $s[j]$ o nucleotídeo presente na j -ésima posição da sequência s .

Os escores calculados com as PWMs podem resultar em valores negativos ou positivos. Um limiar L deve ser utilizado para classificar uma sequência s como sendo um SLFT representado pela PWM M ou não. Ou seja, se $Escore(s|M) > L$, s é considerada um SLFT, caso contrário não. Um valor pouco conservador mas razoável do ponto de vista teórico é 0, que indica que a probabilidade da sequência dado o modelo do Motivo (isto é, baseada nas frequências relativas dos nucleotídeos no Motivo) é maior que a probabilidade dado o modelo nulo. A Figura 5 apresenta um exemplo de cálculo do escore de dois sítios, sendo que, para um valor de limiar de 0.0, um deles é considerado um SLFT e outro não.

A PWM agrega algumas vantagens sobre a sequência consenso, pois além da matriz PWM ter a capacidade de levar em conta as frequências relativas das bases e de considerar as frequências de um modelo nulo, no cálculo dos seus elementos são utilizados pseudocontadores para que os valores estimados de probabilidades nunca fiquem zerados e também, o fato dos elementos da matriz serem expressos em termos de logaritmos transforma o cálculo de produtos em somas, o que melhora a precisão dos cálculos, reduzindo assim a propagação dos erros. E em termos de custo computacional, a utilização da PWM possui praticamente a mesma eficiência que a sequência consenso.

Figura 5 – Exemplo de utilização de uma PWM para verificar se um dado sítio pode ou não ser considerado um SLFT. Dada uma sequência de DNA (um sítio), uma PWM e um limiar, calcula-se o escore do sítio somando-se os pesos de cada nucleotídeo em cada posição na PWM. Se o valor do escore for maior que o valor do limiar, que neste exemplo é 0.0, a sequência é considerada como um SLFT. Caso contrário, não será aceita como um SLFT. A princípio, este procedimento pode ser repetido ao longo de todo o DNA para a predição dos SLFT.

Fonte: Antonio Ferrão Neto, 2017

Embora as PWMs apresentem diversas vantagens sobre a utilização de sequências consenso e a maioria das ferramentas para a predição de Motivos utilize PWMs para a caracterização dos Motivos, elas não são capazes de representar as possíveis dependências estatísticas entre as posições dos nucleotídeos, pois na PWM não há como anotar a probabilidade condicional de ocorrência de um determinado nucleotídeo em uma posição dado um nucleotídeo em outra posição.

2.5 Dependência entre as bases

A possível existência de dependência entre as bases de um SLFT é apontada por diversos autores (LU; YUAN; CHEN, 2008; BULYK; JOHNSON; CHURCH, 2002; TO-

MOVIC; OAKELEY, 2007; KULAKOVSKIY et al., 2013; MATHELIER; WASSERMAN, 2013; BI et al., 2011) e é considerada em diversas ferramentas de predição computacional de sítios de ligação de fatores de transcrição, como descrito na seção 4.2.2 da revisão sistemática. Nos modelos mais simples de SLFTs, como as PWMs, assume-se que as bases contribuem de forma independente para a ligação, de modo que a energia total da interação seja a mera soma das energias dos contatos individuais (BENOS; BULYK; STORMO, 2002).

A dependência entre as bases também é evidenciada por métodos experimentais bioquímicos (ZHAO et al., 2012; MAN; STORMO, 2001) ou em métodos baseados em energia de interação, podendo ser representada como *Z-scores* (TOMOVIC; OAKELEY, 2007; AHMAD et al., 2006; GROMIHA et al., 2004; KONO; SARAI, 1999) ou métodos baseados em entropia da informação (ERILL; O'NEILL, 2009) ou métodos estruturais, que levam em consideração a geometria das moléculas (GUO; LOFGREN; FARREL, 2014; CHIU; KOLODZIEJCZAK, 1991; BAIN et al., 2007; SIGGERS; HONIG, 2007)

Enquanto as tecnologias experimentais e os métodos computacionais baseados em sequências podem responder às questões de *onde* (localização do genoma) e *quais* (a sequência de ligação), as abordagens baseadas em estrutura também podem fornecer explicações sobre o *porquê* e a *forma* como eles se vinculam nesses locais (SHANE; ALVIN et al., 2015).

Os métodos que utilizam experimentos bioquímicos possuem a vantagem de utilizar como modelo o próprio material biológico, sem a necessidade de abstrações ou modelos que os represente. Mas os experimentos de bancada costumam utilizar protocolos complexos, demorados, caros e nem sempre assertivos.

Já os métodos baseados em sequências, como é o caso deste trabalho, na qual apenas as sequências de bases do DNA a serem investigadas são fornecidas, tem a vantagem de serem computacionalmente muito mais econômicos, exigindo recursos computacionais menores e apresentando velocidade de processamento maior que os métodos baseados em estruturas ou bioquímicos.

2.6 Gramáticas Regulares Estocásticas

Gramáticas Regulares Estocásticas (GREs) são modelos estatísticos e linguísticos capazes de representar dependências entre posições vizinhas. Por esse motivo, as GREs representam uma alternativa às PWMs para representar tais dependências.

Nesta seção, apresentaremos algumas definições sobre gramáticas utilizadas neste trabalho. Descrições mais detalhadas podem ser encontradas em textos sobre linguagens formais (RAMOS; NETO; VEJA, 2009; HOPCROFT; ULLMAN; MOTWANI, 2002; MENEZES, 1998; SIPSER; QUEIROZ, 2007).

As gramáticas constituem sistemas formais baseados em regras de substituição, por meio dos quais é possível representar o conjunto das cadeias que compõem uma determinada linguagem.

Formalmente, uma gramática estocástica G pode ser definida como sendo uma quintupla $G = (N, \Sigma, R, S, P)$ sendo que: N é o conjunto de símbolos não-terminais⁶; Σ é o conjunto de símbolos terminais⁷ da gramática; R é o conjunto de regras de substituições ou produções da gramática; S é o símbolo inicial da gramática ($S \in N$); P é uma função $P : R \rightarrow [0, 1]$ que associa a cada produção em R uma probabilidade p .

Uma gramática estocástica é regular se suas produções são todas da forma $\alpha \rightarrow \beta$, com $\alpha \in N$ e ($\beta \in \Sigma$ ou $\beta \in \Sigma N$), ou todas da forma $\alpha \rightarrow \beta$, com $\alpha \in N$ e ($\beta \in \Sigma$ ou $\beta \in N\Sigma$).

Gramáticas regulares são capazes de caracterizar as dependências entre símbolos vizinhos de uma sentença justamente com as produções $\alpha \rightarrow \beta$ com $\beta \in \Sigma N$ (ou alternativamente $\beta \in N\Sigma$), nas quais o símbolo não terminal (que representa a próxima posição a ser analisada) depende do símbolo terminal à sua esquerda (ou alternativamente à sua direita). Cada conjunto de produções com o mesmo lado esquerdo possui uma distribuição de probabilidades, que deve totalizar 1. A figura 7 mostra um exemplo de gramática regular estocástica, na qual $\Sigma = \{a, c, g, t\}$, N é o conjunto de símbolos S e $S1$ a $S25$, e S é o símbolo inicial. O símbolo ε representa a cadeia vazia.

⁶ N corresponde ao conjunto dos símbolos auxiliares utilizados na estruturação e na geração das sentenças sem, no entanto, fazer parte das mesmas.

⁷ Σ corresponde ao conjunto dos símbolos que podem ser justapostos para compor as sentenças da linguagem que se está definindo.

2.6.1 Derivação de GREs

Existem alguns algoritmos de aprendizado supervisionado para derivar ou treinar GREs, como **Lapfa** (RON; SINGER; TISHBY, 1998), **Amnesia** (RON; SINGER; TISHBY, 1996), **Alergia** (CARRASCO; ONCINA, 1994) e **RPNI** (ONCINA; GARCIA, 1992). Neste trabalho será utilizado o **Lapfa** (do inglês “Acyclic Probabilistic Finite Automata”), por ter obtido o melhor resultado em um trabalho de caracterização de sítios de *splicing* (KASHIWABARA et al., 2007).

O **Lapfa** é um algoritmo de aprendizado de distribuições geradas por autômatos finitos acíclicos determinísticos estocásticos (SIPSER; QUEIROZ, 2007), que baseia-se em um “modelo de aprendizado de distribuições de probabilidades”. E, sendo o objetivo “aprender distribuições”, a definição do erro da hipótese deve se basear em uma métrica de “distância entre distribuições”. A “distância” escolhida para esse algoritmo foi a divergência de Kullback-Leibler, definida da seguinte forma:

$$D_{KL}[P_M^N][P_{M'}^N] \stackrel{def}{=} \sum_{r \in \sum^N} P_M^N(r) \log \frac{P_M^N(r)}{P_{M'}^N(r)} \quad (4)$$

sendo que, P_M^N e $P_{M'}^N$ são distribuições de probabilidades de cadeias de tamanho N geradas pelos autômatos finitos determinísticos estocásticos M e M' (VIEIRA, 2004). O algoritmo **Lapfa**, que possui o objetivo de modelar sequências curtas, que é exatamente o caso dos sítios de ligação dos fatores de transcrição, que são sequências curtas, de 5 a 25 bases.

O algoritmo **Lapfa** destina-se a aprender autômatos finitos determinísticos **acíclicos** estocásticos (AFDA estocásticos), com a seguinte propriedade: o gráfico subjacente de cada PFA nesta subclasse é acíclico. A **profundidade** de um AFDA estocástico é definida como sendo o comprimento do caminho mais longo, desde o estado inicial q_0 até o estado final q_f .

O **Lapfa** considera AFDA estocásticos em níveis. Em um AFDA estocásticos em níveis, cada estado pertence a um único nível d , onde o estado inicial, q_0 é o único estado no nível 0 e o estado final, q_f , é o único estado no nível D , na qual D é a profundidade do AFDA. Todas as transições de um estado no nível d são para estados no nível $d + 1$, exceto para as transições rotuladas pelo símbolo final ζ , que podem ir de qualquer estado para o estado final. O conjunto de estados pertencentes ao nível d , é denotado por Q_d . Pode-se

demonstrar (RON; SINGER; TISHBY, 1998), que cada APFA pode ser transformado em um APFA nivelado menor.

O algoritmo utiliza então um AFDA estocástico de níveis, como estrutura para realizar o processo de inferência. Inicialmente o algoritmo constrói uma *árvore de prefixos*, (SEGEWICK; WAYNE, 2011) representando todos os prefixos das sequências de treinamento. Nesta árvore, cada caminho a partir da raiz até uma folha da árvore representa um exemplo do conjunto de sequências de treinamento. Para cada nó interno da árvore, pode-se atribuir um prefixo de um exemplo da amostra. Cada aresta da árvore está associada com um contador que indica o número de prefixos que foram utilizados no caminho.

O algoritmo de aprendizagem do **Lapfa** recebe três parâmetros:

1. ϵ : um parâmetro de precisão (parâmetro p1 do **GrammarLab**);
2. μ : um parâmetro de distinção (parâmetro p2 do **GrammarLab**) e
3. δ : um parâmetro de confiança (parâmetro p3 do **GrammarLab**).

A Figura 6 apresenta o exemplo de uma árvore T correspondente ao exemplo apresentado na Figura 4. A Figura 7 mostra as produções que definem essa gramática.

Apesar do fato dessa árvore T corresponder a um autômato finito (e, por conseguinte, a uma gramática regular) que representa todos os elementos da amostra de treinamento, ela não constitui uma boa hipótese para o problema, uma vez que a árvore descreve apenas as sequências que pertencem à amostra de treinamento ou que constituem o prefixo dos elementos da amostra de treinamento. Em outras palavras, é preciso enfrentar o problema de *overfitting* (DIETTERICH, 1995). Para evitar esse problema, o **Lapfa** aplica um processo de generalização, onde os estados similares do autômato são unidos em um único novo estado, conforme descrito a seguir.

Para cada nível do autômato sendo construído, o **Lapfa** testa todos os pares i e j de estados do nível em análise para verificar se esses possuem a seguinte propriedade: “Dado um limiar $m_0 = p1$ (parâmetro ϵ de entrada), ambas as contagens m_i e m_j devem ser maiores ou iguais a m_0 , sendo que os estados devem ser similares⁸”. Caso essa propriedade seja verificada, o algoritmo junta i e j e todos os pares de estados que eles alcançam.

⁸ A grosso modo, dois nós são considerados similares se as estatísticas, de acordo com a amostra das cadeias que podem ser vistas como geradas a partir desses nós, são semelhantes (RON; SINGER; TISHBY, 1998).

Quando o nível D é atingido, o último gráfico, G_N , é transformado em $G_N + 1$ ⁹. O gráfico final, $G_N + 1$, é então transformado em um APFA enquanto suaviza as probabilidades de transição⁹.

Figura 6 – Árvore da gramática para o exemplo da Figura 4.

Fonte: Antonio Ferrão Neto, 2016

⁹ Conforme descrito nas rotinas AddSlack e GraphToPFA em (RON; SINGER; TISHBY, 1998), ou, respectivamente, nas rotinas junta_{LAPFA} e adiciona_folga_{LAPFA} em (VIEIRA, 2004).

Figura 7 – Produções que definem a gramática representada pela árvore da Figura 6.

$S \rightarrow cS1$	[0,38]	$S9 \rightarrow tS15$	[0,88]
$S \rightarrow tS9$	[0,62]	$S15 \rightarrow aS16$	[0,86]
$S1 \rightarrow tS2$	[1,00]	$S16 \rightarrow tS17$	[1,00]
$S2 \rightarrow aS3$	[1,00]	$S17 \rightarrow cS18$	[1,00]
$S3 \rightarrow tS4$	[1,00]	$S18 \rightarrow aS19$	[0,33]
$S4 \rightarrow aS5$	[0,60]	$S18 \rightarrow cS20$	[0,17]
$S4 \rightarrow cS7$	[0,40]	$S18 \rightarrow tS21$	[0,50]
$S5 \rightarrow aS6$	[1,00]	$S19 \rightarrow \varepsilon$	[1,00]
$S7 \rightarrow tS8$	[1,00]	$S20 \rightarrow \varepsilon$	[1,00]
$S6 \rightarrow \varepsilon$	[1,00]	$S21 \rightarrow \varepsilon$	[1,00]
$S8 \rightarrow \varepsilon$	[1,00]	$S15 \rightarrow gS22$	[0,14]
$S9 \rightarrow gS10$	[0,13]	$S22 \rightarrow tS23$	[1,00]
$S10 \rightarrow gS11$	[1,00]	$S23 \rightarrow aS24$	[1,00]
$S11 \rightarrow tS12$	[1,00]	$S24 \rightarrow aS25$	[1,00]
$S12 \rightarrow cS13$	[1,00]	$S25 \rightarrow \varepsilon$	[1,00]
$S13 \rightarrow aS14$	[1,00]		
$S14 \rightarrow \varepsilon$	[1,00]		

Fonte: Antonio Ferrão Neto, 2016

2.6.2 Cálculo do escore de uma GRE sobre um sítio no DNA

Ao contrário das PWMs, as GREs normalmente não incluem a consideração de modelo nulo. Assim, o modelo nulo deve ser levado em consideração durante o cálculo do escore das sequências. O escore de uma sequência s é calculado com a equação *log-odd*:

$$SG(s) = \log P(s|G) - \log P(s|N) \quad (5)$$

na qual $\log P(s|G)$ é o log da probabilidade da sequência s dada pela gramática G e $\log P(s|N)$ é o log da probabilidade da sequência s dado o modelo nulo N , representado por uma distribuição **i.i.d.** (independente e identicamente distribuída). Definido um limiar L , a sequência s é considerada como pertencente ao modelo caracterizado por G se $SG(s) > L$.

2.7 Desempenho de um classificador

Esta seção descreve alguns conceitos de Aprendizado Computacional utilizados neste projeto.

A abordagem utilizada para a predição computacional de SLFTs, seja utilizando as PWMs ou as GREs, é a de classificador binário, que considera o problema de classificar um objeto (no contexto deste projeto, uma sequência) em apenas duas classes, atribuindo-se a ele uma classificação “positiva” (que significaria ser um sítio de ligação do fator de transcrição em questão) ou “negativa” (que significa não ser tal sítio).

Tal classificador é aprendido a partir de uma amostra de treinamento, que no caso de PWMs ou GREs é o conjunto das sequências utilizadas para derivá-las, as quais sabe-se que são verdadeiramente sequências de sítios de ligação do fator de transcrição de interesse, e por isso ditas sequências *positivas*¹⁰. Da mesma, para avaliar o desempenho do classificador aprendido, é necessário testar o classificador sobre sequências cuja classificação é conhecida, ou seja, sequências que sejam ou não sejam SLFT, chamadas de *positivas* e *negativas*, respectivamente. Estas sequências utilizadas para estimar o desempenho do classificador compõem a *amostra de teste*.

Formalmente, cada instância de teste possui um rótulo $r \in \{p, n\}$ representando se ela é positiva ou negativa. O *modelo de classificação* (ou *classificador*) é um mapeamento dessas instâncias de testes para as classes preditas. Para distinguir entre a classe real e a classe prevista, usamos letras maiúsculas nos rótulos $\{P, N\}$ para as previsões de classe produzidas pelo modelo.

Dado um classificador e uma instância de teste, existem quatro possíveis resultados: Se a instância de teste for positiva e classificada como positiva, é contada como verdadeira positiva (VP); se for classificada como negativa, é contada como uma falsa negativa (FN). Se a instância for negativa e for classificada como negativa, ela é contada como uma verdadeira negativa (VN); se for classificada como positiva, é contada como uma falsa positiva (FP).

Dado um classificador e um conjunto de várias instâncias de teste (o conjunto de teste), pode-se construir uma matriz 2x2, chamada “matriz de confusão” (também chamada de tabela de contingência), representando as disposições do conjunto de instâncias. Esta matriz, representada na Figura 8, constitui a base para diversas métricas.

¹⁰ Esse tipo de aprendizado, no qual conhece-se previamente a classificação da amostra de treinamento, é chamado *aprendizado supervisionado*

Figura 8 – Matriz de confusão.

Fonte: Antonio Ferrão Neto, 2016

Neste trabalho serão utilizadas as seguintes métricas: sensibilidade, especificidade, precisão, medida-F e erro global.

Por definição, a **sensibilidade**¹¹ é a porcentagem das sequências positivas que o classificador consegue reconhecer corretamente. No caso deste trabalho, a sensibilidade é a capacidade do teste prever corretamente os SLFTs que realmente são SLFTs.

Por definição, a **especificidade**¹² é a porcentagem das sequências negativas que o classificador consegue reconhecer corretamente¹³. No caso deste trabalho, a especificidade é a capacidade do teste prever corretamente os SLFTs que não são SLFTs.

Por definição, a **precisão**¹⁴ é a porcentagem das sequências positivas que o classificador consegue reconhecer entre todas as sequências consideradas positivas pelo teste. No caso deste trabalho, a precisão é a probabilidade de um sítio ser realmente um SLFT,

¹¹ Também chamada de “Taxa de Verdadeiros Positivos” ou “*Recall*”.

¹² Também chamado de “Taxa de Verdadeiros Negativos”.

¹³ Em Bioinformática, a determinação dos sítios de DNA que devem ser considerados é, normalmente, uma tarefa difícil, trabalhosa e cara, necessitando experimentos adicionais para a determinação de que um SLFT predito como negativo é ou não realmente negativo. Além disso, devido ao fato do número de sítios negativos ser muito maior que o número de sítios positivos, o cálculo da especificidade resulta em valores numericamente muito baixos. Por esse motivo, as outras medidas de desempenho, que não consideram as ocorrências negativas, são preferidas.

¹⁴ Também conhecido como “Valor Preditivo Positivo” (VPP).

quando o resultado do teste predisser que o sitio é um SLFT, mesmo quando realmente ele não for.

A **Medida-F** é uma medida de desempenho que considera em seu cálculo a precisão e a sensibilidade. É a média harmônica entre a precisão e o *recall*, atingindo o seu melhor valor quando for igual a 1 e o pior quando for igual a zero.

O **Erro Global** é porcentagem das sequências que o classificador reconheceu erroneamente em relação a todas as sequências que foram testadas (e que foram preditas corretamente ou não). No caso deste trabalho, o Erro Global será o número de SLFTs preditos incorretamente, sobre o total de SLFTs testados. Quanto menor for seu valor, quanto mais próximo de zero, melhor será o desempenho do teste.

Estas métricas são calculadas a partir do número de sequências positivas (SLFTs) corretamente classificadas como tal (VP - verdadeiros positivos), do número de sequências positivas (SLFTs) erroneamente classificadas como negativas (FN - falsos negativos), do número de sequências negativas corretamente classificadas como tal (VN - verdadeiros negativos) e do número de sequências negativas erroneamente classificadas como positivas (FP - falsos positivos):

1. *Sensibilidade* = $VP / (VP + FN)$
2. *Especificidade* = $VN / (VN + FP)$
3. *Precisão* = $VP / (VP + FP)$
4. *Medida-F* = $(2 * precisão * sensibilidade) / (precisão + sensibilidade)$
5. *Erro Global* = $(FP + FN) / (FP + FN + VP + VN)$

A medida TPR (“True Positive Rate”) é o mesmo que a Sensibilidade:

6. $TPR = Sensibilidade$

E a medida FPR (“False Positive Rate”) é o complemento da Especificidade:

7. $FPR = 1 - Especificidade$

Para classificadores baseados em um limiar, cada valor distinto de limiar resultará em uma matriz de confusão distinta.

2.7.1 Gráficos ROC

Os gráficos ROC, ou simplesmente curva ROC, (*Receiver operating characteristics*) são gráficos bidimensionais nos quais a taxa de verdadeiros positivos (TVP) é plotada no eixo vertical (ordenada) e a taxa de falsos positivos (TFP) é plotada no eixo horizontal (abcissa), ao considerar-se diferentes valores de limiares ou de parâmetros. A Figura 9 mostra o espaço de um gráfico ROC.

Figura 9 – Espaço do gráfico ROC.

Fonte: Antonio Ferrão Neto, 2016

Diferentes limiares de classificação produzem diferentes pares (*taxa de verdadeiros positivos, taxa de verdadeiros negativos*), e cada par corresponde a um ponto na chamada curva ROC (FAWCETT, 2006). Como descrito a seguir, essa curva pode auxiliar na comparação de diferentes algoritmos classificadores assim como na determinação de um limiar adequado para a classificação.

Vários pontos no espaço do gráfico ROC são importantes a serem observados e representam diferentes significados de acordo com as estratégias de escolha do limiar. Na Figura 9, o ponto inferior esquerdo A(0,0), representa a estratégia de nunca existir uma

classificação positiva; Esse classificador não comete erros falsos positivos, mas também não produz nenhum positivo verdadeiro. A estratégia oposta, de produção incondicional de classificações positivas, é representada pelo ponto superior direito $C(1,1)$. O ponto $B(0,1)$ representa a classificação perfeita. O desempenho do ponto B é perfeito, como mostrado na figura, reconhecendo todos os positivos e negativos como tal.

Um classificador será tanto melhor quanto maior for a sua taxa de verdadeiros positivos e menor for a sua taxa de falsos positivos, ou seja, quanto mais próximo esse ponto estiver do ponto $B(0,1)$. Portanto, em uma comparação entre diferentes técnicas ou algoritmos de classificação, serão melhor sucedidos aqueles que possuírem mais pontos próximos do canto superior esquerdo, que é o ponto $(0,1)$. Seguindo esse princípio, o classificador representado pelo ponto D da Figura 9 é mais eficiente que aquele representado pelo ponto E. O classificador representado pelo ponto F e quaisquer outros repousados sobre a linha diagonal tracejada no gráfico, ligando os pontos $(0,0)$ e $(1,1)$, representam uma classificação aleatória, por acaso, ou seja, a ineficácia absoluta da classificação. O ponto G, que representaria uma “classificação pior que o acaso” pode representar um erro na interpretação do resultado da classificação: bastaria interpretar o ponto G de maneira invertida (invertendo os rótulos “positivo” e “negativo” reportados por ele) e ele poderia se tornar um “bom classificador”.

Uma maneira de se quantificar a eficiência de uma técnica ou algoritmo de classificação é pela área sob a curva ROC (AUC, *Area Under the Curve*): a melhor técnica ou algoritmo é aquele cuja AUC seja maior.

A Figura 10 apresenta um exemplo do aspecto de uma curva ROC (linha em azul) e a área hachurada representa a área sob a curva ROC (AUC). Quanto maior for a AUC, melhor será a eficiência da técnica ou do algoritmo de classificação.

Portanto, as curvas ROC são úteis para comparar classificadores e avaliar de maneira visual o desempenho de diferentes técnicas ou algoritmos de classificação (FAWCETT, 2006).

Figura 10 – Exemplo de gráfico ROC. A curva ROC é a linha azul e representa todos os possíveis classificadores obtidos para uma mesma técnica ou algoritmo de classificação. A área sob a curva (AUC) é área hachurada. O critério ROC01 é ponto da curva ROC que apresenta a mínima distância r ao ponto $(0,1)$. O critério Youden é aquele ponto que apresenta a máxima altura J , medido na vertical (mesma abscissa), entre esse ponto P na curva ROC e o ponto Q, sobre a “diagonal do acaso”, que liga $(0,0)$ a $(1,1)$.

Fonte: Antonio Ferrão Neto, 2016

2.7.2 Critérios para a escolha de um limiar ótimo para classificação

Um classificador discreto necessita de dois elementos: uma técnica ou algoritmo¹⁵ e um critério de corte para aceitar um determinado item da amostra como “aceito” (rotulado como positivo) ou “não aceito” (rotulado como negativo). O primeiro elemento irá definir uma curva ROC e o segundo elemento irá definir um ponto na curva ROC, onde o classificador irá funcionar. Torna-se necessário então adotar um ponto na curva ROC para a classificação.

¹⁵ Essa técnica pode ou não ser computacional. Por exemplo, uma classificação pode ser efetuada por meio de inspeções manuais de medidas de peças produzidas por uma máquina em uma indústria, com o uso de uma régua. Já o algoritmo aqui presume a utilização algum tipo de processamento computacional, com ou sem automação.

A escolha de um “ponto ótimo de corte” ou limiar de classificação pode seguir critérios subjetivos ou objetivos.

Os critérios subjetivos não dependem apenas dos dados coletados, mas também de informações e critérios externos ao experimento¹⁶.

Os critérios objetivos dependem essencialmente dos dados e são escolhidos previamente. Existem muitos possíveis critérios ou abordagens para a escolha do limiar ótimo¹⁷, incluindo métodos baseados em análises de custo-benefício e medidas de precisão do teste de diagnóstico (sensibilidade/especificidade, valores preditivos e razões de verossimilhança diagnóstica). Dois desses critérios são “ROC01” e “Youden”.

Critério ROC01:

Este critério define como limiar ótimo o ponto na curva ROC mais próximo do ponto (0,1), isto é, o canto superior esquerdo do quadrado de lado unitário (METZ, 1978; VERMONT et al., 1991), que representa taxa 0 (mínima) de falsos positivos e taxa 1 (máxima) de . A Figura 10 mostra essa distância mínima representada pela letra r .

Critério do índice de Youden:

O índice de Youden, denotado pela letra J e definido pela equação: $J = \text{sensibilidade} + \text{especificidade} - 1$, é definido para todos os pontos de uma curva ROC e o valor máximo do índice pode ser usado como critério para selecionar o limiar ideal. O índice é representado graficamente como a altura acima da linha da aleatoriedade, e também é equivalente à área sob a curva (AUC) subtendida por um único ponto de operação (POWERS, 2011). A Figura 10 mostra esse valor máximo representado pela letra J , como sendo a máxima

¹⁶ Por exemplo, em um exame clínico para a detecção de uma doença muito grave, a taxa de falsos positivos necessita ser muito baixa antes que o paciente, ou a sua família, sejam informados sobre a doença. Eventualmente, se o teste der positivo para a doença, o paciente pode ser levado a fazer um outro teste, com uma taxa de falsos positivos muito menor, mas esse exame pode ser muito caro, ou não estar disponível naquela clínica ou hospital, o teste pode ser dolorido ou aumentar o risco de óbito, etc. Enfim, neste exemplo e em muitos outros, existem fatores externos, subjetivos, que podem influenciar no limiar do classificador.

¹⁷ O pacote do software R, “OptimalCutpoints” (LÓPEZ-RATÓN et al., 2014a), disponível em <https://cran.r-project.org/web/packages/OptimalCutpoints/OptimalCutpoints.pdf> - acessado em 25/06/2017, apresenta 32 diferentes critérios, a saber: “CB”, “MCT”, “MinValueSp”, “MinValueSe”, “ValueSe”, “MinValueSpSe”, “MaxSp”, “MaxSe”, “MaxSpSe”, “MaxProdSpSe”, “ROC01”, “SpEqualSe”, “Youden”, “MaxEfficiency”, “Minimax”, “MaxDOR”, “MaxKappa”, “MinValuePPV”, “ValueNPV”, “ValuePPV”, “MinValueNPVPPV”, “PROC01”, “NPVEqualPPV”, “MaxNPVPPV”, “MaxSumNPVPPV”, “MaxProdNPVPPV”, “ValueDLR.Negative”, “ValueDLR.Positive”, “MinPvalue”, “ObservedPrev”, “MeanPrev” e “PrevalenceMatching”.

diferença, medida na vertical, entre um ponto “P” da curva ROC e o correspondente ponto “Q” na “diagonal do acaso”.

2.7.3 Validação cruzada

A validação cruzada, às vezes chamada de estimativa de rotação, é uma técnica de validação de modelo para avaliar como os resultados de uma análise estatística serão generalizadas para um conjunto de dados independentes. Ela é usada principalmente em situações onde o objetivo é a previsão, desejando-se estimar com precisão como o modelo preditivo irá se comportar na prática. Em um problema de previsão, o modelo é geralmente obtido para um conjunto de dados fornecidos, sobre o qual o treinamento é executado (conjunto de dados de treinamento) e um conjunto de dados desconhecidos (ou dados não observados anteriormente) contra os quais o modelo é testado.

O objetivo da validação cruzada é definir um conjunto de dados para “testar” o modelo na fase de treinamento (isto é, o conjunto de dados de validação), a fim de limitar problemas como *overfitting*, dando uma visão sobre como o modelo irá generalizar a um conjunto de dados independente (isto é, um conjunto de dados desconhecido, por exemplo de um problema real), etc. Uma rodada de validação cruzada envolve a partição de uma amostra de dados em subconjuntos complementares, realizando a análise em um subconjunto (denominado conjunto de treinamento) e validando a análise no outro subconjunto (chamado conjunto de validação ou conjunto de testes). Para reduzir a variabilidade, várias rodadas de validação cruzada são realizadas usando diferentes partições, e os resultados de validação são calculados em média ao longo das rodadas.

Assim, a validação cruzada *k-fold* consiste em dividir a amostra em k partes (*folds*) mutuamente exclusivas e de tamanhos similares, e então executar k etapas de treinamento e teste. Em cada etapa, uma das k partes é utilizada para o teste do modelo aprendido (ou treinado) com as $k - 1$ partes restantes. Essa técnica está representada na Figura 11.

O desempenho do modelo a ser aprendido com a amostra completa é então estimado pela média do desempenho dos k modelos gerados e testados.

Figura 11 – Validação cruzada.

Fonte: Antonio Ferrão Neto, 2017

3 Objetivo

3.1 Objetivo Geral

O objetivo geral deste trabalho é comparar as técnicas baseadas em *Position Weight Matrix* e em Gramáticas Regulares Estocásticas no problema de identificação de sítios de ligação de fatores de transcrição. A hipótese a ser testada é que, na presença de dependência entre bases dos sítios, as gramáticas possuem potencial para apresentar melhores desempenhos que as PWMs, uma vez que são capazes de representar dependências entre bases vizinhas.

3.2 Objetivos Específicos

1. Realizar uma revisão bibliográfica sistemática do estado da arte das técnicas computacionais publicadas para a caracterização e predição de Sítios de Ligação de Fatores de Transcrição;
2. Criar uma base de amostras positivas de SLFTs baseada em bancos de dados públicos;
3. Comparar o desempenho da identificação dos SLFTs utilizando PWMs e GREs.

4 Revisão Bibliográfica Sistemática

Neste capítulo foi realizada uma revisão bibliográfica sistemática para a determinação do estado da arte das técnicas computacionais publicadas para a caracterização e predição de Sítios de Ligação de Fatores de Transcrição.

Existem diversas abordagens utilizadas para a predição de Sítios de Ligação de Fatores de Transcrição (SLFT), com a adoção de diversas ferramentas experimentais biológicas, bioquímicas, computacionais e estatísticas. Esta seção apresenta uma revisão sistemática dos artigos que abordam a previsão dos SLFT, apontando as características de cada abordagem. Embora a comparação entre as diversas ferramentas ou abordagens tenha sido prejudicada pela ampla multidisciplinaridade das abordagens empregadas e a ausência de um padrão único utilizado pelos autores, esta revisão sistemática torna-se fundamental para todos os pesquisadores interessados no estado da arte das técnicas de predição dos SLFT.

4.1 Materiais e Métodos da Revisão Sistemática

Nesta revisão sistemática foi utilizada uma metodologia de pesquisa bibliográfica que objetiva identificar estudos primários e secundários, relacionados com um tópico particular de pesquisa, o que permite a avaliação de todas as pesquisas relevantes em um assunto (KITCHENHAM, 2004).

Uma Revisão Sistemática pode ser dividida em três partes bem definidas: planejamento, condução e análise. Na fase de planejamento, um protocolo é definido, especificando questões de investigação e a metodologia a ser empregada na condução da revisão.

Além disso, este protocolo define expectativas para revisão sistemática, fontes de referência, critérios para a inclusão ou exclusão de estudos primários, palavras-chave e outros tópicos de interesse. Na fase de condução, a pesquisa bibliográfica é levada a cabo. Nesta fase, os estudos são selecionados de acordo com os critérios de inclusão e exclusão definidos previamente. Finalmente, na análise de resultados, a extração dos dados é executada, e os resultados são comparados.

O fato de existir um protocolo estabelecido, permite que a revisão sistemática possa ser auditada e reproduzida.

4.1.1 Planejamento

Primeiramente, foi definida a seguinte questão:

- Quais são as técnicas computacionais para a caracterização e predição de Sítios de Ligação de Fatores de Transcrição?

Uma análise exploratória sobre a predição de SLFTs foi conduzida sobre o banco de dados PubMed. Esse estudo preliminar ajudou na obtenção de uma fonte de referência de controle (MATHELIER; WASSERMAN, 2013) e na definição de algumas palavras-chave utilizadas na revisão sistemática. Os seguintes banco de dados foram utilizados nesta revisão sistemática:

- PubMed¹
- Teses da USP²
- Banco de teses da Capes³
- Portal Periódicos da Capes⁴

Os mecanismos de busca dos bancos “Teses da USP”, “Banco de teses da Capes” e “Portal Periódicos da Capes” são muito mais limitados que o mecanismo de busca do “PubMed”. Apenas no banco “Teses da USP” foi encontrado um trabalho, que foi incluído manualmente. Os demais bancos de dados foram descartados. Os critérios de busca no PubMed foram os seguintes:

Que tivessem a expressão “transcription factor” ou o seu plural e que também tivessem as expressões “prediction” e “binding site” ou o seu plural, no título ou no resumo, com a restrição de conterem pelo menos uma das seguintes expressões em quaisquer campos do artigo: “computational”, “program”, “in silico”, “in silicon”, “tool”, “algorithm”. E desta seleção, foram excluídos os artigos que contenham em seu título ou resumo as expressões: “micrornas”, “mirna”, “mitochondrial dna”, “mtdna”, “bacterial”, “transcriptome” e “prokaryotic cells”. A composição desses termos, traduzidos para o site de busca de artigos PubMed fica, então:

¹ Disponível em <<<http://www.ncbi.nlm.nih.gov/pubmed>>> - acesso em 12/03/2016.

² Disponível em <<<http://www.teses.usp.br/>>> - acesso em 12/03/2016.

³ Disponível em <<<http://servicos.capes.gov.br/capesdw/>>> - acesso em 12/03/2016.

⁴ Disponível em <<<http://novo.periodicos.capes.gov.br/>>> - acesso em 12/03/2016.

("transcription factor"[Title/Abstract] OR "transcription factors"[Title/Abstract]) AND "prediction"[Title/Abstract] AND ("binding site"[Title/Abstract] OR "binding sites"[Title/Abstract]) AND ("computational"[All Fields] OR "program"[All Fields] OR "in silico"[All Fields] OR "in silicon"[All Fields] OR "tool"[All Fields] OR "algorithm"[All Fields]) NOT ("micrnas"[Title/Abstract] OR "mirna"[Title/Abstract]) NOT ("mitochondrial dna"[Title/Abstract] OR "mtdna"[Title/Abstract]) NOT bacterial[Title/Abstract] NOT "transcriptome"[Title/Abstract] NOT "prokaryotic cells"[Title/Abstract] AND English[lang]

Para selecionar somente os artigos que são relevantes para o estudo desta revisão sistemática, os seguintes critérios de inclusão e exclusão foram estabelecidos:

Critério de Inclusão:

- trabalhos que caracterizassem ou predissessem os SLFTs.

Crítérios de Exclusão:

-
- trabalhos que abordassem a predição de SLFTs, mas que não fossem predições computacionais (p. ex. predições bioquímicas);
- artigos que não abordassem a predição dos SLFTs na íntegra (p. ex. excluir apresentações ou posteres de conferências);
- artigos que tratassem de organismos procariotos (por ex. bactérias);
- artigos que tratassem de RNA mitocondrial;
- artigos que tratassem de transcriptoma e
- artigos que tratassem de DNA mitocondrial.

4.1.2 Condução e Extração de Dados

As pesquisas foram conduzidas entre os dias 15 e 30 de junho de 2015 pelo site do PubMed, resultando em 330 artigos. Deste total, 102 artigos foram analisados, 40 foram rejeitados e 62 foram selecionados segundo os critérios de inclusão/exclusão e 20 foram aceitos após a leitura dos artigos selecionados. A Figura 12 mostra um fluxo da seleção dos artigos.

Figura 12 – Fluxo resumizando o estágio da seleção.

Fonte: Antonio Ferrão Neto, 2016

4.2 Síntese da Extração de Dados

A Tabela 1 mostra uma síntese dos artigos analisados.

Tabela 1 – Síntese dos artigos analisados.

Artigos	Descrição	Método de Caracterização dos SLFT	Natureza do Artigo				Bancos de Dados	Interdependência entre bases	Análise dos resultados	Nome do programa	Outros programas utilizados
			Estática	Computacional	Experimental	Benchmark					
[Xu et al. 2015]	Considera os níveis de metilação e o artigo demonstra o poder preditivo dos níveis de metilação do DNA.	PWM CHIP-seq peaks	X	X			JASPAR, RegulonDB	Sim	Sensibilidade (recall), Precisão, Acurácia, Especificidade, Curva ROC	Methylphet (RF algorithm)	MATCH, CENTIPEDE, CROSeD, MEME
[Broin et al. 2015]	Esse artigo descreve um método com uma abordagem genética "k-medoids", que permite os motivos se moverem livremente entre os clusters em qualquer ponto no processo de clusterização.	k-mer frequency vector (KFV).		X			JASPAR, TRANSFAC	Não	Validação cruzada	não fornecido	MoSta STAMP
[Pujato et al. 2014]	O método utiliza modelo estrutural em uma das fases da construção, o que representa uma dificuldade adicional para a predição.	PWM	X	X	X	X	JASPAR, UniPROBE, RefSeq, Modeller, TF2DNA database	Sim	p-Valor, z-score	TF2DNA	RosettaDNA MAST
[Navarro et al. 2014]	Utiliza uma abordagem para detectar "cis-regulatory modules" (CRM) por meio de uma abordagem de mineração "fuzzy itemset".	PWM	X	X	X		JASPAR, TRANSFAC	Sim	p-Valor, Patser-score	CisMiner	
[Talebzadeh e Zare-Mirakabad 2014]	Utiliza abordagem epigenética. Este artigo utiliza dados de padrões alinhados de nucleossomos modificados para melhorar a performance da predição dos sítios de ligação dos fatores de transcrição (SLFT). Neste artigo, os autores propuseram duas abordagens: "modified nucleosomes neighboring" e "modified nucleosomes occupancy", para reduzir os falsos positivos na descoberta e predição de SLFT.	PWM	X	X	X			Sim	Curva ROC	não fornecido	FANTOM 4
[Eichner et al. 2013]	Workflow de classificação em quatro passos para uma dada sequência de proteína: (1) discrimina o Fator de Transcrição de outras proteínas; (2) determina a superclasse estrutural dos Fatores de Transcrição; (3) identifica o domínio do Fator de Transcrição de ligação ao DNA e (4) prediz seu motivo de cis-atuação no DNA. Basicamente, parte-se da estrutura primária da proteína (sequência de aminoácidos) e procura-se predizer o motivo.	PWM	X	X			JASPAR, TRANSFAC, MatBase, InterPro, UniProt	Sim	Precisão, Validação cruzada, Curva ROC	TFpredict e SABINE	DNAbinder, InterProScan
[Zhong et al. 2013]	Abordagem de predição de ligações de FT que leva em consideração os contextos biológicos, como os estágios do desenvolvimento e tipos de tecido. Utiliza uma abordagem que integra Protein Binding Microarrays (PBMs) "in vitro", conservação de sequências e informação epigenética específica de tecido. Segundo os autores, a análise PBM baseada em k-mer pode eficientemente inferir as especificidades das ligações dos FTs, levando-se em consideração a interdependência entre as bases.	max E-score of k-mers (Max E-score)	X	X	X		JASPAR, TRANSFAC, ENCODE project data	Não	Curva ROC, p-Valor, E-score	PIPES	Seed-and-Wobble algorithm (denoted as S&W PWMs) CENTIPEDE HOMER on ChIPseq data

(continua)

(continuação)

Artigos	Descrição	Método de Caracterização dos SLFT	Natureza do Artigo				Bancos de Dados	Interdependência entre bases	Análise dos resultados	Nome do programa	Outros programas utilizados
			Estatística	Computacional	Experimental	Benchmark					
[Guturu et al. 2013]	Leva em consideração fatores tridimensionais das moléculas.	3DPWM (definido no artigo).	X	X	X		JASPAR, TRANSFAC, UNIPROBE	Sim	p-Valor, z-score, taxa de falsos positivos ("false discovery rate" - FDR)	não fornecido	PRISM framework
[Dror et al. 2014]	Covariação: variação correlacionada de duas variáveis. Neste artigo os autores investigam a covariação entre a sequência de aminoácidos na proteína e a sequência e a forma dos seus alvos no DNA, que são os SLFT.	PBM PFM	X		X			Sim	Coefficiente de Correlação, Validação cruzada, Curva ROC, p-Valor, E-score, MI mutual information score	não fornecido	BLOSUM45, DNASHape
[Piper et al. 2013]	O método relaciona dados de footprinting com dados de CHIP-seq, utilizando métodos bioquímicos em conjunto com métodos computacionais de terceiros para, finalmente, executar o algoritmo Wellington, que leva em conta a interação proteína-DNA.	DNase I Hypersensitive Sites (DHSs) e Digital Genomic Footprinting (DGF)		X	X		ENCODE	Sim	Curva ROC, p-Valor	Wellington pyDNase	DNase-seq, CENTIPEDE, HOMER
[Mathelier e Wasserman 2013]	Neste artigo os autores introduzem um modelo alternativo à PWM ("Position Weight Matrix"), ao qual eles denominam "Transcription Factor Flexible Models" (TFFMs) para representar as propriedades de ligação dos Fatores de Transcrição ao DNA. Este modelo é baseado em Modelos Ocultos de Markov, sendo capazes de modelar tanto a interdependência das bases dentro dos Sítios de Ligação dos Fatores de Transcrição, quanto os variáveis comprimentos de motivos dentro de um único padrão. Essa abordagem pode ser equivalente ao uso das GRE.	PWM HMM TFFM (transcription factor flexible models: conceito definido e introduzido no artigo) DWM (dinucleotide weight matrices)	X	X	X		JASPAR, ENCODE, MafK, MAPPER, PAZAR	Não	Coefficiente de Correlação, Validação cruzada, Curva ROC, p-Valor	TFFM-framework	MEME, MITOMI, MAST
[Morozov e Ioshikhin 2013]	Este artigo é uma extensão de um estudo prévio dos autores que utilizava uma abordagem de refinamento das PWMs, mostrando a capacidade das PWMs refinadas para a predição de novos SLFTs.	PWM		X	X		JASPAR, TRANSFAC, UniPROBE	Não	E-score, z-score, Matthews correlation coefficient	não fornecido	
[He et al. 2012]	O método apresentado aqui, CTF, é um novo método integrado para predizer SLFT que combina vários recursos utilizando campos aleatórios condicionais como uma estrutura subjacente. Para isso, utiliza uma classe de modelagem estatística chamada "Conditional Random Fields" (CRF).	PWM	X	X	X		JASPAR	Sim	Curva ROC	CTF	Chromia, Cluster-Buster

(continua)

(continuação)

Artigos	Descrição	Método de Caracterização dos SLFT	Natureza do Artigo				Bancos de Dados	Interdependência entre bases	Análise dos resultados	Nome do programa	Outros programas utilizados
			Teórica	Computacional	Experimental	Benchmark					
[Tan et al. 2012]	Este artigo discute a interdependência entre as bases na caracterização dos SLFTs e apresenta uma abordagem onde a informação contida na PWM é enriquecida por uma técnica de captura e transferência de informação.	PWM Transferred Information (TI) e Minimum Transferred Information (MTI) - conceitos definidos no artigo.	X	X	X			Não	Performance média de site	tfbsInfoScanner	SOMBRERO, MEME, AlignACE
[Hooghe et al. 2012]	Neste artigo, os autores levam em conta a dependência posicional dos nucleotídeos ("positional dependencies of nucleotides" - NPDs) e a estrutura dependente da sequência do DNA. Eles empregam um algoritmo de "Random Forests" para explorar com flexibilidade ambas as informações.	PWM					TRANSFAC, RegulonDB	Não	Curva ROC	RF algorithm	MATCH, RF algorithm, CROSeD, MEME
[Wright et al. 2011]	Neste artigo, são utilizadas técnicas de aprendizado de máquina: "Bayesian Network" (BN) e "Support Vector Machine" (SVM) para quatro conjunto de dados de SLFTs e são analisadas as suas performances.	PWM	X	X			TRANSFAC	Sim	Validação cruzada, Curva ROC	não fornecido	
[Bi et al. 2011]	Utiliza uma abordagem baseada em árvores para especificar as interdependências entre as bases. Os autores comentam a possibilidade de dependência entre as bases não contíguas (a longa distância), devido à estrutura tridimensional do complexo de ligação entre as proteínas e os SLFTs. Esta abordagem se assemelha à utilização de uma Gramática Regular,	Árvore TPWM (Tree-based PWM) definida no artigo.	X	X			JASPAR, TRANSFAC	Não	Validação cruzada, Curva ROC, Matthew Correlation Coefficient (MCC), chi-square test	TPD	MEME, HMS, ChipMunk
[Fazius et al. 2011]	Abordagem que emprega o conjunto de motivos de entrada como um template de busca para varrer uma sentença de consulta.	SITaR score	X	X	X		JASPAR, TRANSFAC	Sim	Sensibilidade (recall), Especificidade, Medida F, Validação cruzada	SITaR (Site Tracking and Recognition)	Alibaba2, P-Match, Match
[Bauer et al. 2010]	Os autores desenvolveram um método chamado SiteSleuth, na qual aplica-se a predição da estrutura do DNA, química computacional e aprendizado de máquina para o desenvolvimento de SLFTs.	Supervised learning approach, which we call SiteSleuth, involves training a linear SVM classifier to distinguish TF binding sites.	X	X	X			Não	Validação cruzada, Curva ROC	SiteSleuth	QPMEME, BvH (method of Berg and von Hippel), Match, MATRIX SEARCH

(continua)

(continuação)

Artigos	Descrição	Método de Caracterização dos SLFT	Natureza do Artigo				Bancos de Dados	Interdependência entre bases	Análise dos resultados	Nome do programa	Outros programas utilizados
			Estatística	Computacional	Experimental	Benchmark					
[Lahdesmaki et al. 2008]	Este trabalho tem uma abordagem estatística e computacional que permite considerar evidências de múltiplas fontes para enriquecer a predição. Essas fontes podem ser diversas, tais como modelos estatísticos múltiplos (motivos) dos Fatores de Transcrição, conservação evolucionária, potencial regulatório, ilhas CpG, posicionamento dos nucleossomos, sítios de hipersensibilidade à DNase, segmentos de ligação de CHIP-chip e outros conhecimentos prévios biológicos baseados na sequência de nucleotídeos. A abordagem estatística emprega métodos tanto probabilísticos (likelihood), quanto Bayesianos. Um algoritmo de Cadeias de Markov Monte Carlo (MCMC) é empregado para o método Bayesiano.	PWM PBM (protein binding microarray)	X	X			JASPAR, TRANSFAC, ABS, ORegAnno, UCSC, genome browser	Sim	Sensibilidade (recall), Especificidade, Validação cruzada, Curva ROC, p-Valor	ProBTf (ferramenta web)	ESPERR

Fonte: Antonio Ferrão Neto, 2016

A seguir, foi realizada uma análise global, levando-se em consideração tópicos relevantes para identificar o estado da arte. Cada artigo foi dividido em categorias e analisadas nas seções seguinte. A divisão nas categorias considerou o método de caracterização dos SLFTs, a natureza do artigo, o banco de dados empregado, se o método empregado considera ou não a dependência entre as bases, a forma de análise dos resultados, o nome do programa desenvolvido e outros programas utilizados.

4.2.1 Análise Global

Conforme podemos observar na Tabela 1, a maioria dos artigos utilizam a PWM como um dos métodos de caracterização dos SLFTs, embora em vários artigos exista um crescente interesse na utilização de métodos de predição que incorporem alguma informação adicional além daquela presente na PWM, para melhorar a eficiência da predição. Somente com a utilização das PWMs, muitos falsos positivos são obtidos na predição (TALEBZADEH; ZARE-MIRAKABAD, 2014). Existem também outras abordagens com o objetivo melhorar a eficiência da predição. Essas abordagens, estão classificadas na tabela 1 quanto à natureza do artigo, que podem ser:

1. Estatística: nestas abordagens, os autores procuram melhorar as predições com o uso de técnicas estatísticas. Algumas técnicas estão presentes em quase todos os artigos, principalmente na validação das ferramentas desenvolvidas, mas em alguns artigos, esta abordagem se torna a principal, como é o caso do artigo Dror (DROR et al., 2014), que trata de uma análise de covariação entre os domínios dos FT e a estrutura

3D do DNA. Nenhum programa é fornecido neste caso e a análise estatística é o foco do artigo;

2. Computacional: neste caso, para melhorar a eficiência da predição, os autores empregam diversas técnicas conhecidas na Ciência da Computação, como Modelos Ocultos de Markov, *Support Vector Machine*, árvores, Lógica Fuzzy, entre outras;
3. Experimental: neste caso, dados provenientes de experimentos biológicos, como dados de ChIP-chip, experimentos com a DNase I, dados de experimentos *in vitro* de *Protein Binding Microarray* (PBM), dados estruturais das proteínas ou do DNA, obtidos pelas técnicas de cristalografia de difração de Raios-X ou espectroscopia de Ressonância Nuclear Magnética (RNM), entre outros, são utilizados em conjunto com a PWM para enriquecer a capacidade preditiva e reduzir o número de falsos positivos;
4. *Benchmark*: A maioria dos autores empregam técnicas estatísticas, até mesmo para validar os resultados obtidos de uma ferramenta desenvolvida, confrontando esses resultados com os dados obtidas por ferramentas rivais. Mas existem autores que apresentam essa comparação de maneira bastante extensa, tornando o *benchmark* a característica principal do artigo. é o caso do artigo de Pujato (PUJATO et al., 2014), que efetuou um *benchmark* da predição da ferramenta desenvolvida (TF2DNA) com 311 modelos de PWM previstos experimentalmente e armazenados nos bancos de dados *Jaspar* (MATHELIER et al., 2015) e *UniPROBE* (NEWBURGER; BULYK, 2009).

Observamos também na Tabela 1, que os bancos de dados mais utilizados pelos autores desses artigos que tratam de predição computacional de SLFTs são os bancos *Jaspar* (13 artigos = 65%) e *TRANSFAC* (11 artigos = 60%).

4.2.2 Melhorias na predição dos SLFTs

Predizer o local exato onde se localizam os trechos onde os fatores de transcrição se ligam no DNA, ou seja, os SLFTs, é um grande desafio, tanto devido às grandes dimensões do DNA, mesmo considerando-se apenas as regiões cis-regulatórias, quanto pela grande quantidade de falsos positivos que são encontrados quando se utiliza apenas a PWM.

Devido a isso, nos últimos anos tem havido uma tendência no sentido de se buscar técnicas para melhorar a eficiência na predição dos SLFTs, conforme discutido a seguir.

Melhoria da predição com dados biológicos

As abordagens de melhorias com dados biológicos procuram levar em conta informações coletadas de experimentos biológicos *in vivo* ou *in vitro* e com esses dados, em conjunto com dados computados previamente e registrados, por exemplo, nas PWMs, reduzir o número de falsos positivos que seria obtido apenas com a utilização das PWMs, sem informações biológicas adicionais.

1. Níveis de metilação:

A informação biológica considerada aqui está nas alterações do nível de metilação do DNA, que sabidamente influencia no relacionamento do FT com o DNA. Essa abordagem está apresentada em (XU et al., 2015), onde os autores desenvolveram um programa em R, chamado **Methylphet**, que para utilizar perfis de metilação com recursos genômicos. A idéia é baseada no fato de que eventos de ligação de FT *in vivo* ocorre com níveis de metilação alterados. Então, neste método, ao invés de analisar os dados “in silico”, este método dados de metilação coletados experimentalmente.

2. Abordagem genética:

Nessa abordagem, apresentada, por exemplo, em (BROIN; SMITH; GOLDEN, 2015), ao invés de se considerar a PWM, utiliza-se o conceito de “perfil de ligação familiar”(FBP - “Familial Binding Profile”), que é uma espécie de média obtida entre proteínas de uma mesma família. E também o conceito de vetor de frequências *k-mer* (KFV - “k-mer Frequency Vector”), um vetor de dimensão 4 elevado a *k*, composto pela frequência de cada *k-mer* possível em um dado Motivo, permitindo determinar similaridades pelo uso de alguma métrica.

3. Abordagem epigenética:

Aqui, a informação biológica utilizada é a modificação das histonas. Em (TALEB-ZADEH; ZARE-MIRAKABAD, 2014), os autores descobriram que a utilização das posições genômicas dos nucleossomos modificados podem ser informativos para prever a localização das ligações dos FT.

4. Estágios biológicos específicos de tecido:

Contextos biológicos, tais como os estágios do desenvolvimento e tipos de tecidos podem influenciar, ou são influenciados, pelas ligações dos FTs nos seus SLFTs. Essa informação é utilizada em (ZHONG; HE; BAR-JOSEPH, 2013) em um programa, chamado PIPES, para prever ligações de FTs específicos do tecido.

5. Inclusão de dados de ChIP-seq e *footprinting*:

Em (PIPER et al., 2013), os autores descrevem a ocorrência de um desequilíbrio nas informações de alinhamento específicas da cadeia de DNA dos dados de DNase-seq em torno das interações proteína-DNA, permitindo uma previsão precisa dos SLFTs. Eles apresentam um novo algoritmo, chamado Wellington, que considera o desequilíbrio nestas informações específicas da cadeia para identificar de forma eficiente as “pegadas” de DNA (“DNA footprints”). Este algoritmo melhora significativamente a especificidade, reduzindo a proporção de falsos positivos e requer um número significativamente menor previsões em relação aos métodos baseados apenas em PWM.

Melhoria da predição com dados estruturais da proteína e/ou do DNA

Dados estruturais das proteínas e do DNA podem melhorar a capacidade de predição dos SLFTs. Em (GUTURU et al., 2013), os autores adotam essa abordagem e utilizam uma adaptação das PWMs para caracterizar os Motivos dos SLFTs. Eles chamam essa “nova PWM” de 3DPWM, onde eles combinaram sequências descritas na forma de PWMs com dados tridimensionais obtidos das estruturas das ligações dos FT com o DNA.

Em (PUJATO et al., 2014) também são utilizados dados estruturais.

Melhoria da predição com o uso de técnicas computacionais

As técnicas computacionais, normalmente conhecidas como métodos “in silico”, são utilizadas recorrentemente para o tratamento dos dados obtidos dos sequenciadores, principalmente na era pós genômica e mais ainda após a o surgimento dos “sequenciadores de próxima geração”, que reduziu bastante o custo do sequenciamento.

Atualmente, dados genômicos e de proteínas encontram-se disponíveis em grande quantidade em diversos sites na internet, como o Ensembl 2016 (YATES et al., 2016), GenBank (BENSON et al., 2012), FlyBase (ATTRILL et al., 2015), entre outros. Este

fato estimula a utilização de métodos que sejam apenas computacionais. Por outro lado, também torna-se desejável a utilização de métodos que sejam mais eficientes do que aqueles que utilizam somente as PWMs.

Considerar a dependência entre as bases é uma das maneiras de se melhorar a eficiência da predição sem que seja necessária a introdução de mais informações, além daquelas já fornecidas. Existe apenas um custo computacional adicional, mas com a custo cada vez menor dos computadores digitais, o custo computacional tende a ser reduzido.

Dentre os 20 artigos analisados, 12 artigos, ou seja, 60% dos artigos consideram a dependência das bases em suas diferentes abordagens. Observa-se uma tendência crescente no uso desta abordagem.

Dentre os artigos pesquisados, nenhum apresentou uma abordagem que utilize Gramáticas Regulares Estocásticas (GREs) para caracterizar os SLFTs. Entretanto, uma abordagem semelhante que utiliza árvores foi encontrada, com a mesma finalidade, que é a de se considerar a dependência entre as bases.

A seguir estão listados alguns métodos computacionais para melhorar a eficiência da predição, disponíveis na literatura.

Melhoria da predição com modelos alternativos à PWM

1. Abordagem que emprega o conjunto de motivos de entrada como um template de busca para varrer uma sentença de consulta (FAZIUS; SHELEST; SHELEST, 2011);
2. Abordagem baseada em árvores para especificar as dependências entre as bases (BI et al., 2011);
3. Um modelo de PWM refinado com a utilização de dinucleotídeos. (MOROZOV; IOSHIKHES, 2013);
4. Utilização de Random Fields (HE et al., 2012);
5. Utilização de Random Forests (HOOGHE et al., 2012);
6. Utilização de Modelos Ocultos de Markov (MATHELIER; WASSERMAN, 2013);
7. Utilização de Support Vector Machines (SVM) (WRIGHT et al., 2011) e
8. Detecta os "cis-regulatory modules"(CRM) por meio de uma abordagem de mineração "fuzzy itemset"(NAVARRO et al., 2014).

Melhoria da predição com técnicas estatísticas

1. Modelos de Covariação (DROR et al., 2014);
2. Random Forests (HOOGHE et al., 2012);
3. Modelos Bayesianos (WRIGHT et al., 2011);
4. Support Vector Machines (SVM) (WRIGHT et al., 2011);
5. Técnica de captura e transferência de informação. (TAN et al., 2012) e
6. Técnica que aglutina um conjunto de Motivos em um template. (BAUER et al., 2010).

Melhoria da predição com a utilização de várias técnicas combinadas

1. Workflow empregando várias técnicas (EICHNER et al., 2013) e
2. Abordagem que emprega métodos tanto probabilísticos quanto bayesianos para considerar evidências de diversas fontes. (LAHDESMAKI; RUST; SHMULEVICH, 2008).

4.3 Discussões da Revisão Sistemática

A caracterização dos SLFT com baixa taxa de falsos positivos permanece um desafio ainda não solucionado, principalmente em uma abordagem que utilize somente dados de sequenciamento “in silico”.

A Tabela 1 também apresenta as diversas formas que os autores utilizaram para analisar os seus resultados. Pode-se observar que não existe um consenso entre os autores, no sentido de se definir um padrão único que permita uma comparação direta e objetiva entre os diversos programas computacionais, métodos ou abordagens apresentados. Assim, é possível que uma mesma ferramenta computacional, desenvolvida pelos autores, possa ser considerada boa ou ruim, dependendo da forma de análise adotada.

Essas abordagens são muito diferentes, tanto em relação à natureza do artigo, quanto na forma de análise dos resultados e isso também dificulta muito a comparação entre as abordagens. Algumas nem mesmo são computacionais, embora também tratem da predição de SLTF.

Além disso, o fato de não existir nenhuma ferramenta de referência ou um padrão único de comparação entre todas as ferramentas desenvolvidas, prejudica a comparação entre os métodos.

Outro aspecto, que não foi apresentado na Tabela 1 por ser subjetivo, é o nível de esforço que precisa ser feito em relação ao ganho de eficiência da predição. Esse nível de esforço pode se referir tanto ao tempo adicional de processamento (CPU, memória, etc.), no caso de um programa de computador, quanto ao esforço experimental necessário (ou mesmo custo financeiro) para que dados *in vivo* ou *in vitro* sejam obtidos, ou dados cristalográficos da estrutura de uma proteína, ou o esforço de uma análise estatística, etc. Algumas abordagens conseguem um ganho de eficiência às custas de um esforço tão elevado, que pode não valer a pena adotar essa abordagem, até mesmo porque, algumas vezes, as condições necessárias para a realização dos experimentos, podem não estar disponíveis.

Além destes problemas, a extrema multidisciplinaridade das abordagens apresentadas exige diferentes competências, conhecimentos e técnicas, o que dificulta o entendimento dos artigos e a reprodução de suas abordagens.

Apesar dessas dificuldades, a categorização das abordagens apresentada neste artigo, deverá muito ser útil para o leitor, facilitando o estudo e a seleção das abordagens relacionadas.

4.4 Conclusão da Revisão Sistemática

A revisão sistemática foi conduzida com o objetivo de identificar o estado da arte da predição de SLFTs e verificamos também que existem tendências de se aperfeiçoar a predição dos SLFTs utilizando-se uma abordagem que incluía na análise mais informação além daquela existente nas PWMs ou que, analisando os mesmos trechos de DNA que originam as PWMs, consigam extrair mais informações além daquelas obtidas diretamente das sequências de bases consideradas independentes, sem nenhuma consideração biológica, estrutural, computacional, estatística ou alguma combinação de todas as abordagens.

Observamos, por meio da análise de vários artigos, que existem limitações no uso das PWMs e que outros métodos podem melhorar a eficiência da predição. Não foi encontrado nenhum trabalho que aborde este problema de predição de SLFT com o uso de gramáticas e, por essa razão, este projeto traz algo inovador para esta área.

5 Materiais e Métodos

Este capítulo descreve os materiais e métodos utilizados a fim de alcançar os objetivos propostos. As seções 5.1 e 5.2 descrevem como foram obtidas as amostras de SLTFs e como elas foram utilizadas para avaliar os classificadores por meio de uma validação cruzada adaptada. A seção 5.3 descreve como os classificadores baseados em PWMs e GREs foram treinados e avaliados. A seção 5.4 aborda como foi estimada a dependência entre as bases nos SLTFs.

5.1 Criação da base de amostras de SLTFs

Como já mencionado na Seção 2.7, os classificadores utilizados neste trabalho, tanto os baseados em PWMs quanto em GREs, são treinados a partir de sequências positivas (reconhecidas como SLFT), chamada de amostra de treinamento, e avaliados utilizando-se amostras de sequências positivas e amostras negativas. Esta seção descreve como essas amostras foram obtidas ou criadas.

5.1.1 Obtenção das amostras positivas

Foram pesquisados bancos de dados públicos para a obtenção de dados para as amostras que contivessem sequências de sítios de ligação de fatores de transcrição que fossem validados experimentalmente ou que tivessem sido identificados por meio de algum experimento bioquímico, por exemplo, técnicas de SELEX, ChIP-chip, ChIP-seq ou *footprinting*. Como critérios de busca, os dados para a criação dessas bases foram obtidos por meio de uma pesquisa exploratória na internet, partindo-se da *string* de busca: “*transcription factor binding site database*”, que resultou em sites que apontam para links de outras páginas e assim sucessivamente.

Vários bancos foram encontrados, muitos porém com poucas sequências. Pensou-se em fazer uma união desses bancos, porém a maioria deles continha sequências sem localização genômica (provavelmente obtidas por SELEX). Logo, não seria possível misturar sequências alvo de um mesmo fator de transcrição e espécie de bancos distintos com a certeza de que são de *loci* diferentes, a fim de não duplicar dados e invalidar a estimação dos parâmetros probabilísticos.

Desta forma, optou-se por utilizar apenas uma fonte de SLFTs, o banco de dados **Jaspar 2016**¹ (MATHELIER et al., 2015), por ser o que possuía maior número de sítios para o maior número de fatores de transcrição, para diferentes espécies. Arquivos² no formato FASTA desses sítios foram obtidos por meio do link “DOWNLOAD”, disponibilizado na página inicial do site. Cada arquivo, referente a uma amostra positiva refere-se a um conjunto de sítios de ligação de um mesmo fator de transcrição em uma determinada espécie.

O banco **Jaspar 2016** disponibiliza 594 arquivos de sítios de diferentes fatores de transcrição e espécies. Dentre esses 594 arquivos, quatro (POL001.1.sites, POL005.1.sites, POL011.1.sites e POL013.1.sites) são de sítios de RNA polimerase II, portanto não são objetos deste estudo.

Além disso, existem 174 arquivos, nomeados como “PFXXXX.1.sites”, que possuem muitas sequências que ou são idênticas ou a diversidade das sequências é muito pequena. Por exemplo, algumas amostras possuem apenas três tipos de sequências. A amostra pode possuir centenas de sequências, mas todas iguais a uma dessas três sequências. Após serem eliminadas as repetições, as poucas sequências que restam não formariam os cinco clusters, necessários para o processo da validação cruzada.

Alguns arquivos utilizam caracteres diferentes de A, C, G ou T. Provavelmente seguem o padrão IUPAC para a notação de nucleotídeos, mas por estarem fora do padrão da maioria dos demais arquivos do **Jaspar 2016**, foram descartados.

Outros arquivos apresentam algumas inconsistências, que foram reportadas aos responsáveis pelo **Jaspar 2016**, como sequências diferentes do tamanho do motivo e linhas vazias. Esses poucos arquivos também foram descartados.

Finalmente, dos 594 arquivos “.sites” fornecidos pelo banco **Jaspar 2016**, restaram 341 arquivos que puderam ser aproveitados pois formaram pelo menos cinco clusters.

Os arquivos foram tratados para que todas as sequências contivessem apenas as bases dos sítios onde os fatores de transcrição se ligaram. Nos arquivos do extttJaspar 2016, essas bases estão destacadas das demais com letras maiúsculas. As bases restantes, que não pertencem aos sítios onde o fator se ligou, foram deletadas.

¹ Disponível em (<http://jaspar.genereg.net/>) - acesso em 12/03/2017.

² Esses arquivos possuem a extensão “.sites”, mas o formato dos arquivos é FASTA.

5.1.2 Amostra Negativa

É comum em Bioinformática não ser possível obter uma amostra realmente negativa, ou seja, sequências que com certeza não são positivas. O que é feito, então, é utilizar sequências aleatórias. Neste trabalho, a amostra negativa (referente a cada amostra positiva) é composta por sequências aleatórias contendo o mesmo tamanho e distribuição de nucleotídeos das sequências da amostra positiva a que se refere.

Isto foi feito simplesmente embaralhando-se os nucleotídeos de cada sequência da amostra positiva, gerando assim as sequências negativas. Para cada sequência positiva foram geradas até dez mil sequências aleatórias³. Por fim, foram excluídas da amostra negativa as sequências repetidas e aquelas sequências que eventualmente já existiam na amostra positiva. Ou seja, a preservação da composição da frequência é feita com um algoritmo que troca as posições dos nucleotídeos nas sequências, sem retirar ou acrescentar nucleotídeos de cada sequência, apenas trocando-o de lugar com outro nucleotídeo escolhido aleatoriamente na sequência e repetindo isso para todas as posições da sequência.

5.2 Preparação de amostras de treinamento e teste para validação cruzada baseada em agrupamentos

Para avaliar os classificadores baseados em PWMs e GREs, será utilizada uma estratégia de validação cruzada para cada fator de transcrição e espécie presentes nas amostras obtidas do banco de dados *Jaspar* 2016. Porém, uma adaptação será feita. Quando a validação cruzada convencional é utilizada na análise de sequências, é muito comum que as sequências repetidas sejam removidas, permanecendo no arquivo apenas uma sequência de cada, sem repetições. Isso é feito para que se garanta que não existam sequências repetidas no treinamento e no teste. No entanto, 1) isso não garante que sequências muito similares estejam uma na amostra de treinamento e outra na amostra de teste e 2) quando as sequências repetidas são removidas da amostra de treinamento, a distribuição das sequências originais é alterada, comprometendo assim o aprendizado do classificador.

³ Arquivos pequenos foram embaralhados várias vezes até o total de sequências se aproximar de dez mil. Arquivos grandes, com mais de dez mil sequências foram embaralhados apenas uma vez.

No intuito de atacar esses dois problemas, este trabalho seguiu a abordagem descrita em (NETO, 2016)⁴, cujas conclusões e resultados principais estão apresentados na Tabela 2.

Tabela 2 – Resultados do estudo comparativo de validações cruzadas para a predição de sítios de ligação de fatores de transcrição (NETO, 2016).

Em uma validação cruzada tradicional, as sequências que aparecem mais de uma vez são consideradas uma única vez e as demais, redundantes, são removidas do conjunto de treinamento. Entretanto, as repetições removidas podem conter informações valiosas para o treinamento. Se uma sequência repetir muitas vezes, o peso de suas bases deve influenciar significativamente a distribuição de probabilidades que se deseja obter para a PWM na fase de treinamento.

O objetivo deste trabalho de conclusão de curso foi comparar as duas abordagens apenas sobre PWMs, uma convencional, com os “folds” criados sem repetições e outra com os “clusters” construídos com repetições para o treinamento e sem repetições para os testes.

As comparações, realizadas com essas medidas de desempenho, mostram uma significativa melhoria na classificação para as medidas de desempenho relacionadas aos falsos positivos (especificidade e precisão), quando as repetições são consideradas no treinamento para a construção das PWMs.

VALIDAÇÃO CRUZADA	Sensibilidade	Especificidade	Precisão	Medida-F	Erro global
CONVENCIONAL					
(sem repetições):					
MÉDIA TOTAL	97%	90%	59%	70%	10%
DESVIO PADRÃO	10%	13%	29%	25%	11%
CLUSTERIZADA					
(com repetições):					
MÉDIA TOTAL	97%	92%	66%	75%	7%
DESVIO PADRÃO	10%	11%	66%	75%	10%

Fonte: Antonio Ferrão Neto, 2016 (TCC).

Resumidamente, a fim de realizar uma validação cruzada 5-vezes, (5-*fold*) cada amostra positiva foi dividida em 5 partes (cada uma correspondendo a um *fold*), cada parte agrupando sequências mais similares (potencialmente idênticas) entre si. Somente quando o *fold* é utilizado como teste é que as sequências idênticas são removidas. A Figura 13 apresenta uma visão geral deste processo. Os detalhes de como esse procedimento foi executado são descritos a seguir.

A amostra negativa não foi particionada. Ou seja, enquanto os testes e os treinamentos foram aplicados alternadamente em cada passo da validação 5-*fold* para a amostra

⁴ Trabalho de conclusão de curso de graduação em Ciências Biológicas do próprio autor (Antonio Ferrão Neto), disponível como “Project”, no site **ResearchGate**: <https://www.researchgate.net> - acessado em 13/11/2017.

Figura 13 – Visão geral do processo de geração das amostras de teste e treinamento para a validação cruzada.

Fonte: Antonio Ferrão Neto, 2017

positiva, a amostra negativa foi testada integralmente em cada *fold*, conforme a Figura 11.

5.2.1 Cálculo das similaridades entre as sequências e agrupamentos

A fim de dividir cada amostra positiva em cinco partes, foi utilizado um algoritmo de agrupamento hierárquico, tendo como métrica de distância entre duas sequências a dissimilaridade entre elas. Para isso, foi necessário calcular a similaridade entre as sequências, duas a duas, utilizando-se o algoritmo de alinhamento Smith-Waterman (SW) (SETUBAL; MEIDANIS, 1997) produzindo-se assim uma matriz, que pode ser triangular⁵, para minimizar a ocupação de memória.

Existe um pacote disponível para a linguagem de programação Python 2.7, chamado `swalign`, que implementa o cálculo do alinhamento SW. Utilizamos os parâmetros de `score 2` para *match* e `-1` para *mismatch* (conforme o código C.1 do Apêndice C).

Assim, dadas duas sequências x e y , a distância entre elas é dada por:

$$d(x, y) = 1 - \text{identidade}(x, y) \quad (6)$$

sendo $\text{identidade}(x, y)$ a porcentagem de identidade entre as sequências x e y em escala de 0 a 1.

Calculadas as distâncias entre cada par de sequências, o agrupamento das mesmas foi realizado por meio da função `hclust` do software R utilizando o método de aglomeração *Ward* (parâmetro `ward.D`) e realizando o corte para definição de 5 grupos (*clusters*). Alguns testes foram realizados e foi verificado que, na criação de *clusters*, o melhor método é o “ward.D” ((MURTAGH; LEGENDRE, 2014)), entre todos os oferecidos na versão mais atual do R. Os demais métodos de aglomeração produziam grupos muito desbalanceados, muitas vezes não conseguindo formar 5 grupos. A sequência de comandos utilizada está exemplificada no Apêndice C, código C.2.

Mesmo utilizando o parâmetro `ward.D`, 70 amostras foram agrupadas em menos de 5 grupos, sendo estas descartadas do teste. Desta forma, um total de 341 arquivos do banco *Jaspar 2016* foram utilizados para a comparação de PWMs e GREs. aproveitados. A tabela 6 no apêndice A relaciona esses 341 arquivos.

⁵ Se a amostra possui N sequências, essa matriz possuirá $N \times N$ elementos, sendo que N pode ser da ordem de milhares. A ocupação de memória é portanto bastante crítica e uma vez que essa matriz é simétrica, somente os elementos acima ou abaixo da diagonal principal necessitam ser armazenados. Teremos assim uma matriz triangular com $N(N-1)/2$ elementos.

5.2.2 Preparação dos agrupamentos para teste

Como já mencionado no início desta seção, nos arquivos de teste não existirão repetições, pois não faria sentido testar duas sequências idênticas em um mesmo teste, o que alteraria a estatística de desempenho, pelo fato de se testar mais de uma vez sequências que são idênticas.

Portanto, para cada grupo de sequências obtido na etapa de agrupamento, foi gerada uma versão repetições para ser utilizada como teste na validação cruzada, substituindo cada um dos *folds* de teste, em cada passo da validação, como representado na Figura 11.

Existem, portanto, duas versões de cada *cluster*: uma para testes, que não possui repetições, e a versão de treinamento, que possui repetições. Essas duas versões de arquivos foram utilizadas na mesma validação cruzada.

5.3 Comparação entre PWMs e GREs na identificação de sítios de ligação de fatores de transcrição

Foi criado um programa em linguagem Python para processar as validações cruzadas para cada um dos 341 conjuntos de arquivos das amostras que foram produzidas a partir do Jaspar 2016, tanto para as PWMs, quanto para as GREs. Esta seção descreve como os classificadores foram treinados e avaliados por esse programa.

5.3.1 Modelos nulos para o teste das PWMs e GREs

Neste trabalho foram utilizados três modelos nulos, denominados N1, N2 e N3:

1. Modelo nulo N1: considera a distribuição uniforme de bases, ou seja, 0,25 (25%) para A, C, G e T.
2. Modelo nulo N2: considera a frequência de bases do genoma da espécie em questão. A tabela 3, construída com dados obtidos do NCBI⁶, apresenta uma relação de frequências para cada espécie dos fatores de transcrição utilizados nas amostras.
3. Modelo nulo N3: considera a frequência de bases do próprio sítio que está sendo testado. Neste caso, a PWM não pode ser construída previamente. Ela precisa ser

⁶ National Center for Biotechnology Information (NCBI)[Internet]. Bethesda (MD): National Library of Medicine (US), National Center for Biotechnology Information; [1988] – [cited 2017 Aug 15]. Available from: <https://www.ncbi.nlm.nih.gov/>

reconstruída a cada teste pois as frequências das bases segundo modelo nulo faz parte do cálculo que define a PWM (equação 2).

Tabela 3 – Relação de espécies utilizadas no banco de dados **Jaspar** 2016 e suas respectivas frequências genômicas. As frequências marcadas com asterisco (*) não foram encontradas, então foram estimadas como 25%.

Espécie	A	C	G	T
<i>Antirrhinum majus</i> *	0,2500	0,2500	0,2500	0,2500
<i>Arabidopsis thaliana</i>	0,3170	0,1830	0,1830	0,3170
<i>Bos taurus</i>	0,2906	0,2094	0,2094	0,2906
<i>Caenorhabditis elegans</i>	0,3227	0,1773	0,1773	0,3227
<i>Drosophila melanogaster</i>	0,2893	0,2107	0,2107	0,2893
<i>Gallus gallus</i>	0,2854	0,2146	0,2146	0,2854
<i>Halocynthia roretzi</i> *	0,2500	0,2500	0,2500	0,2500
<i>Homo sapiens</i>	0,2947	0,2053	0,2053	0,2947
<i>Hordeum vulgare</i>	0,2785	0,2215	0,2215	0,2785
<i>Mus musculus</i>	0,2869	0,2131	0,2131	0,2869
<i>Nicotiana sp.</i>	0,3045	0,1955	0,1955	0,3045
<i>Oncorhynchus mykiss</i> *	0,2500	0,2500	0,2500	0,2500
<i>Oryctolagus cuniculus</i>	0,2797	0,2203	0,2203	0,2797
<i>Petunia x hybrida</i> *	0,2500	0,2500	0,2500	0,2500
<i>Pisum sativum</i> *	0,2500	0,2500	0,2500	0,2500
<i>Rattus norvegicus</i>	0,2882	0,2118	0,2118	0,2882
<i>Rattus rattus</i>	0,2882	0,2118	0,2118	0,2882
<i>Triticum aestivum</i>	0,2825	0,2175	0,2175	0,2825
<i>Xenopus (Silurana) tropicalis</i>	0,2973	0,2027	0,2027	0,2973
<i>Xenopus laevis</i>	0,2988	0,2012	0,2012	0,2988
<i>Zea mays</i>	0,2675	0,2325	0,2325	0,2675

Fonte: Antonio Ferrão Neto, 2017.

5.3.2 Processamento das validações cruzadas com as PWMs e GREs

Conforme discutido na seção 2.7.3, o processamento das validações cruzadas consiste no treinamento do modelo com os dados de quatro dos cinco *folds* e no teste do modelo sobre o quinto *fold* que não foi utilizado para treinamento. E esse procedimento é efetuado cinco vezes, alternando o *fold* de teste entre os *folds* de treinamento.

O treinamento das PWMs e cálculo de escore de cada sequência de teste foram efetuados de acordo com os procedimentos descritos na seção 2.4.1 (equações 2 e 3, respectivamente).

As GREs foram treinadas utilizando o algoritmo Lapfa (RON; SINGER; TISHBY, 1998; KASHIWABARA et al., 2007), um algoritmo para inferência gramatical, mencionado

na seção 2.6 que já se encontra implementado na linguagem de programação C++ no *framework* GrammarLab (MACHADO-LIMA, 2002; VIEIRA, 2004).

Esse procedimento de treinamento da GRE produz um programa executável em linha de comando do sistema operacional, que é compilado especificamente para cada treinamento, portanto gerando um executável para cada conjunto de treinamento. Esse executável recebe como argumento de linha de comando do Linux uma sequência de bases e retorna o escore da GRE para essa sequência.

O teste da GRE é então realizado tomando-se cada sequência do *fold* de teste, uma a uma, e submetendo-a a esse executável que foi previamente treinado com os outros quatro *folds* que haviam sido destinados para o treinamento, nesta rodada da validação cruzada.

Depois dos testes terem sido realizados com as amostras positivas, eles foram também realizados com as amostras negativas, neste caso, com todas as sequências negativas (um único *fold*), utilizando a mesma GRE que foi treinada para as sequências positivas. Então, para cada *fold* de teste utilizado para testar as sequências positivas, foram realizados testes sobre todas as sequências negativas (ver Figura 11).

Para cada *fold* da validação cruzada, 100 GREs foram treinadas variando-se os parâmetros de generalização do Lapfa, conforme realizado em (KASHIWABARA et al., 2007). O parâmetro $p1$ foi fixado em 5, o parâmetro $p2$ variado de 0.1 a 1.0 (em incrementos de 0.1) e o parâmetro $p3$ variado de 0.01 a 0.1 (em incrementos de 0.01).

Esse mesmo procedimento foi executado para os três modelos nulos em conjunto, pelo mesmo programa.

Em resumo, para cada um dos 341 conjuntos de dados (cada conjunto referente a um fator de transcrição e espécie), foram realizadas 303 validações cruzadas: 3 referentes a PWMs utilizando os 3 modelos nulos, e 300 referentes às 100 GREs (estimadas com diferentes conjuntos de parâmetros) utilizando os 3 modelos nulos.

O Apêndice D apresenta exemplos de saídas para os dados das validações cruzadas, tanto para a PWM, quanto para a GRE.

5.3.3 O gráfico ROC e a área sob a curva ROC (AUC)

O desempenho das PWMs GREs obtidas nas validações cruzadas foram comparados por meio de curvas ROC. Para cada um dos cinco *folds* de cada validação cruzada foi

calculada uma curva ROC. Posteriormente foram calculados os valores médios, dentre os cinco *folds*, da AUC e das medidas de desempenho nos pontos ótimos, assim como seus respectivos desvios padrões. A seguir são descritos os detalhes desse procedimento.

Um programa **Python** foi criado para ler os dados de saída das validações cruzadas e produzir a curva ROC. As curvas ROC de cada amostra foram calculadas utilizando-se um pacote do software estatístico **R** chamado **ROCR** (SING et al., 2005). As chamadas às funções do pacote **ROCR** foram efetuadas a partir do **Python**. Essa integração entre a linguagem **Python** e o pacote estatístico **R** foi feita com o uso do pacote **rpy2** do **Python**⁷.

As funções do pacote **ROCR** chamadas para o cálculo da curva ROC são **prediction** e **performance**. A primeira função, **prediction**, apenas prepara os dados de entrada, que foram obtidos da validação cruzada, para o formato que a função **performance** necessita. Esses dados de entrada são os valores dos escores da PWM ou da GRE contidos em uma lista do **Python** e as rotulações, positivas ou negativas, em outra lista. A função **performance** retorna, além da AUC, 34 medidas de desempenho para cada ponto da curva ROC.

Determinação do limiar de escore para classificação

O pacote **ROCR** não fornece nenhuma função que retorne um limiar “ótimo”, conforme algum critério adotado. Então, o que se tem como saída é uma grande quantidade de medidas de desempenho (34 medidas de desempenho), para cada ponto da curva ROC (dezenas ou centenas de pontos), mas nenhum ponto pode ser escolhido objetivamente, automaticamente, conforme um critério previamente estabelecido.

Outro pacote do **R**, o **OptimalCutpoints** (LÓPEZ-RATÓN et al., 2014b), já citado na seção 2.7.2, faz exatamente o que o pacote **ROCR** não faz: dada uma curva ROC, a função **optimal.cutpoints()** (configurada pela função **control.cutpoints()**) do pacote **OptimalCutpoints**, calcula qual é o limiar ótimo, para cada critério adotado. Este pacote foi testado no ambiente **R** e funcionou sem problemas mas não funcionou quando se tentou fazer a sua chamada pelo **Python**, através do pacote **rpy2**.

A estratégia foi então implementar o cálculo de apenas dois critérios diretamente utilizando a própria linguagem **Python**. Foram escolhidos dois critérios: ROC01 e Youden, pelas suas popularidades e simplicidade de cálculo. A Figura 10 ilustra como esses pontos

⁷ Pacote **rpy2**: https://rpy2.readthedocs.io/en/version_2.8.x/ - acesso em 12/08/2017.

são determinados a partir da curva ROC. Os trechos de código para cálculo desses limiares encontram-se no Apêndice C, códigos C.3 e C.4.

5.4 Determinação do nível de dependência existente entre as bases

Foi visto que as PWMs (seção 2.4) não são capazes de considerar a dependência entre as bases, enquanto que as GREs (seção 2.6) podem ser modeladas e treinadas para considerar essa dependência entre as bases.

Portanto, torna-se necessário analisar o nível de dependência existente nos dados da amostra, fornecida no banco **Jaspar 2016**, para que seja verificada a eficácia das GREs sobre as PWMs, especificamente nas amostras onde a dependência entre as bases se verifique ser mais acentuada.

A princípio, para um dado motivo de tamanho n , todas as n bases do motivo podem depender de todas as $(n - 1)$ demais bases. E essa dependência pode acontecer entre 1 base e as $(n - 1)$ demais bases, entre 2 bases e as $(n - 2)$ demais bases, entre 3 bases e as $(n - 3)$ bases e assim sucessivamente, em quaisquer posições e ordens. Entretanto, neste trabalho, por razões de simplicidade, o cálculo de um escore para determinar o nível de dependência entre as bases levará em consideração apenas entre dinucleotídeos.

Neste trabalho, será utilizada a abordagem de informação mútua (CHIU; KOLODZIEJCZAK, 1991; TOMOVIC; OAKELEY, 2007; YE et al., 2017), que é uma medida quantitativa em pares da covariância. Esta métrica quantifica a dependência mútua entre duas variáveis aleatórias discretas $X(X = [A, C, G, T])$ e $Y(Y = [A, C, G, T])$ e é definida como:

$$I(X; Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right) \quad (7)$$

na qual x ($x \in [A, C, G, T]$) e y ($y \in [A, C, G, T]$) representam os nucleotídeos de duas listas de nucleotídeos X e Y . Os valores $p(x)$ e $p(y)$ indicam as frequências das bases x e y . Os valores $p(x, y)$ definem as frequências dos dinucleotídeos (xy) de X e Y . A significância da dependência entre as bases nas duas posições pode ser avaliada usando a desigualdade de Chebyshev⁸.

⁸ Desigualdade de Chebyshev: para um conjunto de dados com média $\bar{x}$ e desvio padrão s :

1- no mínimo 75% dos valores dos dados estarão entre $\bar{x} - 2s$ e $\bar{x} + 2s$. E, no mínimo, 75% dos valores

Finalmente, uma vez que o escore desejado é para toda a amostra e não apenas das sequências individualmente, o escore de toda a amostra é calculado pela média de todos os escores de todas as sequências da amostra.

Assim, essa análise de dependência foi efetuada determinando-se um escore para cada arquivo no formato **FASTA** de cada amostra, utilizando a equação 7. Um programa em linguagem Python foi produzido para ler um arquivo no formato **FASTA** de um conjunto de sequências, calcular um escore de dependência para cada dinucleotídeo e retornar uma média aritmética de todos esses escores, para cada arquivo.

Com o intuito de se verificar a eficácia deste escore como sendo uma medida para se quantificar o nível de dependência entre as bases, foi produzida uma massa de testes criando-se artificialmente a dependência sobre os dados das sequências.

A geração de dados com dependência total de dinucleotídeos foi simulada simplesmente fazendo-se com que uma certa base seja sempre sucedida pela mesma base. Por exemplo, a Figura 14 mostra um conjunto de 21 sequências de bases, com tamanho 6, artificialmente construídas para que haja dependência total de bases de dinucleotídeos adjacentes⁹. Assim, sempre que a base na primeira posição for um “T”, a base na segunda posição será um “G”; sempre que a base na segunda posição for um “G”, a base na terceira posição será um “C” e assim por diante. A expectativa é que essa amostra adquira um escore alto.

Por outro lado, para um arquivo construído com bases totalmente escolhidas ao acaso, ou seja, aleatoriamente, como exemplificado na Figura 15, a expectativa é que o escore desta amostra seja baixo.

A seguir, foram construídas 11 amostras, denominadas “dep_XXX”, com 1000 sequências cada, combinando situações entre esses dois extremos. Então, por exemplo, no arquivo **dep_070**, 70% do arquivo foi construído com sequências totalmente dependentes,

dos dados terão um *z-score* entre -2 e 2 ;

2- no mínimo 89% dos valores dos dados estarão entre $\bar{x} - 3s$ e $\bar{x} + 3s$. E, no mínimo, 89% dos valores dos dados terão um *z-score* entre -3 e 3 e

3- em geral, no mínimo $(1 - 1/k^2) \times 100\%$ dos valores dos dados estarão entre $\bar{x} - ks$ e $\bar{x} + ks$, possuindo um *z-score* entre $-k$ e k , para $k > 1$ (KVANLI; PAVUR; KEELING, 2005).

⁹ Essa construção foi feita artificialmente desta maneira para tornar a sua verificação humanamente possível, quando a simulação for produzida com milhares de sequências. Mas em uma situação real, as dependências não são perfeitas, existindo exceções e as linhas não precisam estar ordenadas, com as bases perfeitamente alinhadas, como neste exemplo, mas isso não interfere no cálculo do escore, já que esses cálculos são feitos entre as posições (colunas) e não entre as linhas (sequências). Além disso, as dependências não precisam necessariamente ser adjacentes.

Figura 14 – Exemplo de simulação de uma amostra artificialmente construída com 21 sequências de 6 bases e com 100% de dependência entre as bases.

	1	2	3	4	5	6
1	T	G	C	A	G	T
2	T	G	C	A	G	T
3	T	G	C	A	G	T
4	T	G	C	A	G	T
5	T	G	C	A	G	T
6	T	G	C	A	G	T
7	A	T	A	C	A	C
8	A	T	A	C	A	C
9	A	T	A	C	A	C
10	A	T	A	C	A	C
11	A	T	A	C	A	C
12	C	C	G	T	C	A
13	C	C	G	T	C	A
14	C	C	G	T	C	A
15	C	C	G	T	C	A
16	C	C	G	T	C	A
17	C	C	G	T	C	A
18	C	C	G	T	C	A
19	G	A	T	G	T	G
20	G	A	T	G	T	G
21	G	A	T	G	T	G

Fonte: Antonio Ferrão Neto, 2017

como na Figura 14 e as sequências restantes, foram sequências totalmente aleatórias, como na Figura 15.

Essas 11 amostras artificiais foram produzidas conforme descrito a seguir:

1. dep_100: 100% de dependência;
2. dep_090: 90% de dependência e 10% de aleatoriedade;
3. dep_080: 80% de dependência e 20% de aleatoriedade;
4. dep_070: 70% de dependência e 30% de aleatoriedade;
5. dep_060: 60% de dependência e 40% de aleatoriedade;
6. dep_050: 50% de dependência e 50% de aleatoriedade;
7. dep_040: 40% de dependência e 60% de aleatoriedade;
8. dep_030: 30% de dependência e 70% de aleatoriedade;

9. dep_020: 20% de dependência e 80% de aleatoriedade;
10. dep_010: 10% de dependência e 90% de aleatoriedade e
11. dep_000: nenhuma dependência, totalmente aleatório;

Figura 15 – Exemplo de simulação de uma amostra artificialmente construída com 21 sequências de 6 bases e com as bases distribuídas de maneira totalmente aleatória.

	1	2	3	4	5	6
1	A	T	A	A	G	A
2	A	G	G	T	A	C
3	C	G	T	A	G	C
4	T	A	A	C	G	C
5	A	A	A	G	A	T
6	A	T	A	A	C	A
7	T	C	C	T	G	G
8	A	G	C	G	T	G
9	C	G	G	G	A	C
10	A	C	C	A	G	G
11	T	T	G	G	G	C
12	A	A	A	C	G	C
13	T	A	C	C	T	G
14	A	C	C	G	C	A
15	C	T	A	A	T	G
16	G	G	C	G	G	G
17	G	G	A	T	C	A
18	G	T	G	C	A	G
19	G	A	G	T	G	T
20	T	T	C	T	G	A
21	C	C	C	G	A	G

Fonte: Antonio Ferrão Neto, 2017

Além disso, as sequências dessas amostras artificiais foram produzidas com o tamanho 10 de bases.

Esses dados, gerados no formato de arquivos **FASTA**, foram submetidos ao programa de cálculo de escores. A Figura 16 mostra os escores calculados para essas 11 amostras (arquivos dep_100 a dep_000), juntamente com um gráfico, que mostra visualmente a dependência entre esses escores e os arquivos, nomeados conforme a porcentagem de dependência artificialmente inserida nos arquivos. Essa Figura mostra que o algoritmo

produziu escores exatamente como o previsto, permitindo-se ordenar os arquivos pelos seus escores, exatamente conforme as dependências foram construídas, sem nenhuma exceção observada entre esses 11 exemplos.

Esses testes servem para confirmar a validade dos escores calculados pelo programa.

Figura 16 – Escores calculados para simulações de 11 amostras artificialmente construídas com diferentes níveis de dependência, de 100% a 0%.

Fonte: Antonio Ferrão Neto, 2017

O escore máximo, calculado com essas amostras artificiais, foi 0,5987 e o escore mínimo foi 0,0027. O escore máximo não ficou mais próximo do valor máximo (1,0000) porque a amostra foi construída apenas com dependências de bases adjacentes. O escore teria sido maior se tivesse havido uma preocupação com a dependência entre bases não adjacentes, mas isso iria aumentar demais a complexidade do processamento, sem trazer resultados muito relevantes para esta análise. Mas, em uma situação real, é possível haver dependência entre bases não adjacentes (SHANE; ALVIN et al., 2015).

6 Resultados e Discussões

Neste capítulo serão apresentados e discutidos os resultados obtidos. A seção 6.1 aborda a base criada de amostras positivas de SLFTs. A seção 6.2 apresenta as medidas de desempenho apresentadas por PWMs e GREs. A seção 6.3 discute a correlação destas medidas com o escore de dependência das amostras. Tanto a seção 6.2 quanto a seção 6.3 apresentam os resultados baseados apenas no critério ROC01 de determinação de limiar ótimo, pois os resultados obtidos com o critério Youden foram muito semelhantes. Figuras relativas aos resultados utilizando os dois critérios são apresentadas no apêndice E. A seção 6.4 compara os resultados gerais com os que foram obtidos na fase inicial deste projeto. Finalmente a seção 6.6 traz as considerações finais deste capítulo, incluindo uma comparação dos resultados obtidos nesta etapa final do trabalho com os obtidos na fase inicial do projeto.

6.1 Criação da base de amostras positivas de SLFTs

Esta seção tem o objetivo de apresentar algumas informações básicas sobre a base criada, de forma a auxiliar a interpretação dos demais resultados.

A Figura 17 mostra o número de arquivos por número de sequências encontradas nesses arquivos, ou seja, o número de amostras de determinados tamanhos. A Figura 18 mostra um *zoom* dos primeiros 101 pontos do gráfico da Figura 17. As totalizações dos dados encontrados no banco de dados estão na tabela 4. Todos esses dados referem-se à versão 2016 do banco *Jaspar*.

6.1.1 Escores de dependência dos dados do banco *Jaspar* 2016

A seção 5.4 mostra uma metodologia para o cálculo de dependência com dados artificialmente gerados. Os dados das amostras reais, do banco *Jaspar* 2016, foram submetidos também a esse mesmo programa de cálculo de escore de dependência. Os resultados dos escores, tanto para os arquivos das amostras artificiais, quanto para as amostras reais, provenientes do banco *Jaspar* 2016, foram colocados em uma mesma tabela, ordenada pelo escore, a fim de determinar faixas de classes de dependência entre as bases das amostras reais. Essas faixas estão apresentadas na Figura 19, juntamente com

Figura 17 – Gráfico mostrando o número de arquivos para cada quantidade de sequências encontradas no banco de dados Jaspár 2016, para todos os FTs, incluindo várias espécies.

Fonte: Antonio Ferrão Neto, 2017

Figura 18 – Mesmo gráfico da Figura 17, mas mostrando apenas os números de arquivos com quantidades de sequências inferiores a 101.

Fonte: Antonio Ferrão Neto, 2017

Tabela 4 – Número Sequências por Quantidades de Arquivos - Jaspar 2016.

Número de Sequências	2016
6	1
7	1
8	1
9	6
10	7
11	3
12	6
13	6
14	4
15	7
16	7
17	8
18	10
19	3
20	16
21	23
22	20
23	18
24	6
25	7
26	3
27	2
28	4
29	3
30	5
Arquivos com mais de 30 sequências	417
Arquivos com mais de 1.000 sequências	168

Fonte: Antonio Ferrão Neto, 2017.

Figura 19 – Faixas de escores de níveis de dependência calculados para dados reais do banco Jaspar 2016 em relação aos dados das simulações produzidas artificialmente.

Fonte: Antonio Ferrão Neto, 2017

um gráfico. A tabela contendo todos os escores de dependência encontra-se no apêndice A.

Pode-se observar na Figura 19 que a maioria das amostras fornecidas pelo banco JASPAR 2016 não apresentam níveis muito elevados de dependências entre as bases: dos 341 arquivos, apenas quatro, que pertencem à classe F, possuem um nível de dependência superior a 50%.

6.2 Comparação de desempenho de PWMs e GREs

Nesta seção, os desempenhos médios das PWMs e GREs, utilizando os três modelos nulos (N1, N3 e N3), são comparados tanto na amostra inteira quanto por faixas de escore de dependência, tamanho de sítio e tamanho de amostra (número de sequências).

6.2.1 Desempenhos médios gerais

Uma comparação dos valores médios (de todas as amostras) das áreas sob a curva ROC (AUCs) para os três modelos nulos (N1, N3 e N3) é mostrada na Figura 20.

Esses valores das AUCs foram obtidos calculando-se a média, para todas as amostras, dos valores médios dos cinco *folds* da validação cruzada, para cada modelo nulo, tanto para as PWMs, quanto para as GREs.

Neste e em todos os demais gráficos das medidas de desempenho, as linhas verticais, no topo de cada barra, representam os mínimos e máximos valores das médias entre os cinco *folds* da validação cruzada.

Nota-se um ótimo desempenho das duas técnicas, utilizando os 3 modelos nulos, no caso geral, isto é, incluindo todas as amostras independente do número de sequências de cada amostra, tamanho dos sítios ou dependência.

A fim de visualizar as diferenças mais sutis entre esses 6 classificadores, a Figura 21 mostra um *zoom* da faixa superior de valores de AUC, utilizando uma escala que se inicia em 0,9955 e termina em 0,990, apresentando valores até a terceira casa decimal. Esta Figura evidencia que, embora com desempenhos similares: 1-) as PWMs apresentaram resultados ligeiramente melhores em média e 2-) os melhores resultados foram obtidos utilizando-se o modelo nulo N2, baseado nas frequências genômicas, seguido do modelo nulo N1, baseado na distribuição uniforme.

Figura 20 – Área sob a curva ROC média sobre todas as amostras. N1: modelo nulo uniforme; N2: modelo nulo baseado na frequência genômica; N3: modelo nulo baseado na frequência da sequência sendo analisada.

Fonte: Antonio Ferrão Neto, 2017.

6.2.2 Desempenhos médios por faixa de dependência

A Figura 22 apresenta as AUCs médias por faixa de escore de dependência. É possível observar que, enquanto as PWMs possuem melhores resultados (AUCs médias maiores que as GREs) nas faixas 0 a 4, as GREs apresentam os melhores resultados nas faixas 5 a 7. Ou seja, observa-se que o desempenho da GRE é superior ao da PWM exatamente quando o nível de dependência é maior. Tal fato é consistente com a hipótese deste trabalho de que as GREs teriam melhor desempenho nos casos em que houvesse dependência entre as bases dos sítios a serem caracterizados. Vale mencionar, no entanto, que as faixas de dependência 5 a 7 são aquelas com menor número de amostras (11 no total, ou seja 3% do total de amostras testadas). Em particular a faixa 7 possui apenas uma amostra (MA0079.2.sites). Logo, seria interessante que esses resultados fossem confirmados ao se analisar um número maior de amostras nestas faixas de dependência.

Figura 21 – Área sob a curva ROC média sobre todas as amostras (*zoom*). N1: modelo nulo uniforme; N2: modelo nulo baseado na frequência genômica; N3: modelo nulo baseado na frequência da sequência sendo analisada.

Fonte: Antonio Ferrão Neto, 2017.

6.2.3 Desempenhos médios por tamanho de sítio

A Figura 23 apresenta as AUCs médias por tamanho de sítio. Foi incluída uma barra a mais para representar o nível de dependência médio em cada agrupamento. O escore de dependência foi normalizado (ao valor máximo foi atribuído o valor 1,0 e aos demais a proporção em relação ao máximo), a fim de que os tamanhos das barras pudessem ser visualizados em conjunto com as demais barras. As barras do escore de dependência não parecem apresentar correlação evidente com os tamanhos dos SLFTs. No entanto, este gráfico mostra que, na maioria dos casos, as AUCs das PWMs são um pouco maiores que as das GREs, porém mais notadamente nas amostras com sítios menores. Isso indica que, além do nível de dependência entre as bases, o tamanho do sítio é um fator que influencia no desempenho das GREs neste problema. Também notou-se uma maior variação dos resultados obtidos utilizando os três modelos nulos N1, N2 e N3 quando os SLFT são pequenos, tanto para as PWMs, quanto para as GREs.

As mesmas conclusões sobre os desempenhos obtidos por tamanho de sítio foram semelhantes para a maioria das demais medidas de desempenho, cujas figuras estão no

Figura 22 – Área sob a curva ROC (AUC) por Escore de Dependência. A tabela abaixo do gráfico mostra as percentagens de dependência para cada faixa de escore.

Fonte: Antonio Ferrão Neto, 2017.

apêndice E (sensibilidade ou taxa de verdadeiros positivos - Figuras 32 e 33, especificidade - Figuras 34 e 35 e erro global - Figuras 40 e 41).

A análise da métrica de precisão, contudo, mostra algumas diferenças. Pode-se perceber na Figura 24 que as PWMs obtiveram melhores desempenhos para quase todos os tamanhos de sítios, com exceção dos tamanhos 16 e 17. Mesmo para os tamanhos 20 e 22, cujas amostras apresentam os dois valores médios mais altos de escore de dependência, as PWMs foram mais precisas que as GREs. O mesmo ocorreu para a medida F (Figuras 38 e 39 do apêndice E). Tal fato pode indicar que as GREs tendem a melhorar mais a sensibilidade (revocação) do que a precisão.

Figura 23 – Área sob a curva ROC (AUC) por tamanho do SLFT. Os valores no eixo horizontal representam os tamanhos dos SLFTs, que variam de 5 a 22. As dependências, normalizadas, foram incluídas para efeito de comparação.

Fonte: Antonio Ferrão Neto, 2017.

Figura 24 – Gráfico da Precisão (critério ROC01) por tamanho do SLFT, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

6.2.4 Desempenhos médios por faixa de tamanho de amostra

A Figura 25 apresenta as AUCs médias por faixa de tamanho de amostra. Assim como na Figura 23, as dependências normalizadas também foram representadas neste gráfico e neste caso, a figura mostra uma evidente correlação entre os tamanhos das amostras e o escore de dependência entre as bases. Tanto as PWMs quanto as GREs obtiveram as maiores AUCs nas faixas 6 em diante, ou seja, com amostras a partir de 100 sequências. O desempenho das PWMs e das GREs é semelhante na maioria dos casos. No entanto, é possível observar que a dependência entre as bases é mais impactante no melhor desempenho das GREs sobre as PWMs do que o tamanho das amostras, fato esse mais evidente ao se comparar as faixas 1, 2 e 9. Na faixa 1, apesar das amostras serem pequenas (compostas por 7 a 9 sequências), as GREs superam as PWMs, provavelmente devido ao alto escore de dependência dessas amostras. A diminuição dessa dependência na faixa 2 ocasiona uma diminuição das AUCs das GREs em relação às das PWMs, mesmo com um leve aumento no tamanho das amostras (10 a 19 sequências). Na faixa 9, mesmo com amostras com mais de 1000 sequências, a baixa dependência também favorece o melhor desempenho das PWMs sobre as GREs.

As mesmas observações podem ser aplicadas às métricas de taxa de verdadeiros positivos (Figuras 42 e 43), especificidade (Figuras 44 e 45) e erro global (Figuras 50 e 51), todas estas presentes no apêndice E. Mas novamente, considerando as métricas de precisão (Figura 26) e medida F (Figura 48 do apêndice E), as GREs obtiveram desempenhos inferiores em relação às PWMs, neste caso para todas as faixas de tamanho de amostra.

6.3 Correlação entre as medidas de desempenho e o escore de dependência

A fim de se analisar a variação das medidas de desempenho de PWMs e GREs de acordo com o escore de dependência das amostras de SLFTs, foi calculado o coeficiente de

Figura 25 – Área sob a curva ROC (AUC) por tamanho da amostra. As dependências, normalizadas, foram incluídas para efeito de comparação. Na parte inferior da figura estão os ranges de 1 a 9, os limites inferiores e superiores dos tamanhos das amostras, a contagem do número de arquivos de cada range, os valores das dependências não normalizadas e os valores das dependências normalizadas.

Range	Limites	Contagem	Dependência média por range	Dependência Normalizada
1	7 a 9	6	0,090102	1,000000
2	10 a 19	63	0,063288	0,702402
3	20 a 29	78	0,043186	0,479297
4	30 a 39	31	0,050997	0,565996
5	40 a 99	37	0,042049	0,466681
6	102 a 156	13	0,036534	0,405475
7	225 a 474	13	0,021292	0,236311
8	515 a 979	14	0,019172	0,212780
9	1010 a 99494	86	0,004385	0,048673
			0,090102	4,117613

Fonte: Antonio Ferrão Neto, 2017.

Figura 26 – Gráfico da Precisão (critério ROC01) por tamanho da amostra, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

correlação de Pearson¹ entre os escores de dependência entre as bases das amostras e os valores de desempenho obtidos nas mesmas para cada métrica considerada.

A Figura 27 apresenta os coeficientes de correlação entre os escores de dependência e a AUC (média entre os 5 *folds*) de cada amostra (apenas os coeficientes com valor de significância abaixo de 1% foram considerados). Assim como aconteceu com todas as demais métricas (Figura 28), os coeficientes de correlação são maiores, em valores absolutos, para as PWMs do que para as GREs, de forma consistente. Ou seja, as métricas das GREs são menos correlacionadas com as dependências entre as bases.

No caso da AUC, a correlação negativa indica que quanto maior for o escore de dependência, menor será a AUC. Para as GREs, verificamos uma diminuição, em valores absolutos, desta correlação negativa. Ou seja, para as GREs, a AUC será menos dependente do escore de dependência e isso é possível porque as GREs conseguem tratar a dependência entre as bases, ao contrário das PWMs.

Todos os valores dos coeficientes de correlação estão na Tabela 7 do Apêndice B).

¹ O coeficiente de correlação de Pearson mede o grau da correlação (e a direção dessa correlação - se positiva ou negativa) entre duas variáveis. Esse coeficiente assume valores apenas entre -1 e 1. Um valor deste coeficiente igual a 1 significa uma correlação perfeita positiva entre as duas variáveis; um valor igual a -1 significa uma correlação negativa perfeita entre as duas variáveis - isto é, se uma aumenta, a outra sempre diminui; um valor igual a 0 significa que as duas variáveis não dependem linearmente uma da outra. No entanto, poderia existir uma dependência não linear.

Figura 27 – Correlações entre o escore de dependência e a AUC, para os três modelos nulos (N1, N2 e N3), tanto para as GREs quanto para as PWMs.

Fonte: Antonio Ferrão Neto, 2016

6.4 Comparação com resultados anteriores

Foram realizados testes preliminares em 2014, com dados do banco de dados **Jaspar** 2014² para a produção da amostra positiva. Os dados, naquela época, eram compostos por amostras de 106 fatores de transcrição, cada um contendo de 9 a 54 sequências.

Naquela ocasião, não foi utilizado a curva ROC e nenhuma estimativa de um limiar ótimo. A sequência foi considerada como positiva tanto pela PWM quanto pela GRE (isto é, reconhecida como um SLFT) se seu escore fosse maior que zero, senão foi considerada negativa.

As duas primeiras linhas da Tabela 5 apresentam as medidas de desempenho médias sobre todos os fatores de transcrição testados para o banco **Jaspar** 2014. Os resultados destas duas linhas mostram que, embora as GREs apresentem menor sensibilidade que as PWMs, as GREs superaram as PWMs em especificidade, precisão, Medida-F e erro global. As duas últimas linhas da Tabela 5 apresentam as métricas de desempenho médias, incluindo a AUC, sobre todos os 341 fatores de transcrição testados do **Jaspar** 2016 e

² O banco de dados **Jaspar** é dinâmico, pois recebe continuamente contribuições de novas fontes de dados.

Figura 28 – Correlações entre os escores de dependência e as medidas de desempenho, para os três modelos nulos, tanto para as GREs quanto para as PWMs, para os dois critérios de limiar adotados (ROC01 e Youden).

que foram discutidos nas seções anteriores deste capítulo.

Tabela 5 – Resultados das métricas de desempenho médias do **Jaspar 2014** (Antigo) e os resultados das métricas de desempenho ponderadas para todos as 341 amostras utilizadas do **Jaspar 2016** (Novo).

Modelo	AUC	Sensibilidade	Especificidade	Precisão	Medida-F	Erro global
GRE (Antigo)	-	81%	86%	64%	70%	15%
PWM (Antigo)	-	85%	80%	56%	66%	19%
GRE (Novo)	98%	99%	96%	38%	45%	4%
PWM (Novo)	99%	99%	98%	47%	53%	2%

Fonte: Antonio Ferrão Neto, 2016.

Uma comparação dos dados antigos com os dados recentes desta tabela 5 mostra que o desempenho de classificação, tanto da GRE, quanto da PWM, calculados anteriormente foi bem menor que o desempenho de classificação calculados recentemente. Uma justificativa para essa diferença pode estar relacionada com os tamanhos das amostras, que eram bem menores na versão 2014 do banco, prejudicando o treinamento dos modelos.

Uma outra justificativa possível para essa diferença é o fato de ter sido utilizado um limiar fixado em zero na classificação antiga. A utilização de um limiar não ótimo implica em um maior erro de classificação.

Outra diferença metodológica que pode ter contribuído

6.5 Ferramenta de Predição de SLFTs

Uma ferramenta de predição de sítios de ligação de fatores de transcrição GREs e PWMs foi desenvolvida e seu código, acompanhado de uma documentação, encontra-se no apêndice F. A ideia é que outros pesquisadores possam aprimorar/adaptar o código, ou simplesmente a lógica do algoritmo, para assim desenvolverem seus próprios algoritmos, com suas próprias abordagens.

Basicamente, a ferramenta de predição recebe como entrada um arquivo, no formato FASTA contendo as sequências de DNA, um arquivo de parâmetros e arquivos “BNF” e “PWM”, contendo as definições das gramáticas e das PWMs e após a sua execução, como saída, o programa gera um arquivo (ou relatório) contendo todos os sítios preditos, tanto para PWMs, quanto para GREs.

A ferramenta de predição foi testada sobre dados genômicos de *Drosophila melanogaster*, obtidos do site Ensembl³ (ZERBINO et al., 2017). Este teste foi realizado sobre todo o genoma de *Drosophila melanogaster*, para todos os fatores de transcrição de *Drosophila melanogaster* encontrados no banco de dados Jaspar 2016⁴.

O número de sítios preditos foi de 137.547.944, para todos os 116 FTs de *Drosophila melanogaster* do Jaspar 2016. Esse grande número de predições pode indicar que grande parte dos sítios preditos não são realmente SLFTs, por razões bioquímicas⁵. É possível que muitos desses sítios sejam ativos em diferentes estágios do desenvolvimento ou em diferentes posições teciduais. A Rede de Regulação Gênica para o desenvolvimento da *Drosophila* define quando e onde cada um desses sítios serão utilizados ou não. De qualquer maneira, ainda assim, muitos desses sítios preditos podem ser apenas falsos positivos⁶. E o maior desafio das ferramentas de predição computacional é diminuir a taxa de falsos positivos, com o mínimo custo⁷ possível.

O arquivo `found_tfbs.txt` pode ser utilizado para confrontar os sítios preditos com os sítios conhecidos, publicados na literatura e que fazem parte dos módulos *cis*-reguladores.

6.6 Considerações finais

Os resultados apresentados neste capítulo mostram que as GREs apresentaram maiores AUC que as PWMs nos casos em que as amostras apresentaram alta dependência entre as bases. O mesmo não aconteceu com relação às medidas de precisão. Conforme observado na seção 5.4 e mostrado na Figura 19, a grande maioria das amostras do banco de dados Jaspar possuem baixo nível de dependência entre as bases. E, como foi discutido na seção 2.6, um baixo nível de dependência entre as bases pode não evidenciar que a utilização das GREs seja vantajosa em relação às PWMs.

³ O endereço do site do Ensembl, para o *download* do genoma de *Drosophila melanogaster*, é: http://www.ensembl.org/Drosophila_melanogaster/Info/Index - acesso em 17/12/2017. Somente o arquivo `Drosophila_melanogaster.BDGP6.dna.toplevel.fa` foi utilizado.

⁴ Esse processamento levou cerca de trinta horas para ser completado, em um dos servidores do IME/USP.

⁵ São diversos os mecanismos bioquímicos conhecidos para a inativação dos FTs, como por exemplo, a metilação do DNA e histonas e a abertura/fechamento da cromatina. E talvez existam mecanismos bioquímicos ainda desconhecidos.

⁶ Uma possível estratégia para verificar se esses sítios preditos são ou não falsos positivos, seria introduzir artificialmente alterações nessas bases que compõem o sítio predito, a fim de descaracterizá-lo e então verificar se ocorrem modificações no comportamento fenotípico dos genes.

⁷ Aqui, por “custo”, entenda-se “custo financeiro”; “custo laboratorial”; “custo computacional” medido, por exemplo, em Tflop/s; “dispêndio de energia”; “horas-homem de trabalho”; etc.

Nos demais casos, as PWMs obtiveram desempenho superior e, em muitos casos, o desempenho das PWMs foi muito próximo do desempenho de um classificador ideal ($AUC = 1$). Este desempenho excepcional das PWMs não condiz com diversos relatos na literatura, cujos autores afirmam que, embora as PWMs constituem-se em um instrumento computacionalmente performático para a predição de SLFTs, ela possui a desvantagem de apresentar uma alta taxa de falsos positivos (TALEBZADEH; ZARE-MIRAKABAD, 2014). Esse desempenho acentuado das PWMs talvez fosse menor se os testes tivessem sido realizados sobre dados genômicos e não sobre sequências que são sabidamente utilizadas exatamente para a produção de PWMs. Além disso, as sequências negativas foram obtidas a partir do embaralhamento das sequências positivas. Seria interessante testar o uso de sequências obtidas com um embaralhamento de dados genômicos.

Cabe mencionar também que este trabalho realizou uma comparação das abordagens PWM e GRE puramente na tarefa de caracterizar as sequências exatas dos sítios, sem nenhuma outra informação de contexto. A classificação ou predição das sequências de bases que estão digitalizadas na memória de um computador, por meio da utilização de alguma técnica computacional, não possui o mesmo significado que a classificação ou predição das sequências de bases *in vivo*, ou seja, as moléculas reais de bases nitrogenadas, que fazem parte do DNA, dentro dos núcleos das células do organismo sob estudo, com todo o seu contexto genômico. Mesmo que uma sequência de bases seja computacionalmente classificada como um SLFT, por ela possuir um “padrão” de bases que combine com o Motivo de um dado SLFT, é possível que a correspondente molécula deste fator de transcrição jamais se ligue em sítios de bases de DNA reais que sejam compostos exatamente por essa sequência em particular, pois existem outros fatores envolvidos na regulação do gene. Isso significa que esse sítio, embora tenha sido classificado computacionalmente como um SLFT, na verdade ele pode não ser, biologicamente, um SLFT genuíno.

Um outro fator que pode favorecer os resultados obtidos pelas PWMs é o possível viés introduzido pelas ferramentas de descoberta de motivos. As ferramentas de descoberta de motivos utilizam PWMs tanto internamente, para construir os motivos, quanto para reportar o motivo encontrado. Não existem ferramentas de descoberta de motivos que consigam descobrir motivos utilizando modelos mais complexos, como HMMs, GREs ou outros. Então, se os sítios que estão sendo preditos nos experimentos bioquímicos são exatamente aqueles reportados pelas ferramentas de descoberta de motivos, tal fato pode ser favorável às PWMs.

As GREs são modelos bem mais sofisticados do que as PWMs. Além do processo de aprendizado da estrutura gramatical ser mais complexo que a estrutura das PWMs, há mais parâmetros (valores de probabilidade) a serem estimados. Assim, considerando-se um tamanho de amostra fixo, o erro de estimação de uma GRE pode ser superior ao erro de estimação de uma PWM. Em casos em que a dependência entre as bases não for muito acentuada, pode não existir vantagens de se utilizar GREs em relação às PWMs.

7 Conclusão

Este trabalho apresenta uma comparação dos desempenhos de classificação de PWMs (*Position Weight Matrices*) e GREs (Gramáticas Regulares Estocásticas) no problema de identificação de sítios de ligação de fatores de transcrição utilizando três modelos nulos: distribuição uniforme, distribuição genômica e distribuição de bases da sequência analisada. Os desempenhos foram estimados por um método de validação cruzada baseado em agrupamentos utilizando amostras de sítios do banco de dados **Jaspar 2016**. A hipótese testada neste trabalho foi que, na presença de dependência entre bases dos sítios, as GREs apresentariam melhores desempenhos que as PWMs, uma vez que são capazes de representar dependências entre bases vizinhas.

Os resultados mostraram que as GREs obtiveram valores de desempenho ligeiramente maiores que os de PWMs para várias métricas (AUC, taxa de verdadeiros positivos, especificidade e erro global) para as amostras que apresentaram maior escore de dependência entre bases. Também foi mostrado que a correlação de todas as medidas com o escore de dependência foi maior para PWMs, sempre no sentido do desempenho piorar com o aumento da dependência. No entanto, os desempenhos das GREs quanto à precisão e medida F foram inferiores aos apresentados pelas PWMs em todos os casos. Este último resultado foi contrário ao que se esperava.

Embora esperava-se observar diferenças maiores de desempenho, os resultados deste trabalho corroboram a hipótese testada. Com base nisso, o que se pode concluir nesse momento, as GREs podem ser sim adequadas na presença de alta dependência de bases mas, caso contrário, as PWMs ainda são preferíveis. No entanto, mais experimentos, especialmente na presença de mais amostras com alta dependência entre as bases, poderiam confirmar ou não esses achados, já que a base testada não possui muitas amostras nessas condições.

7.1 Principais contribuições deste trabalho

As principais contribuições deste trabalho foram:

Estudo comparativo das PWMs x GREs

Este trabalho teve como foco principal a comparação entre as PWMs e as GREs. Além das conclusões dos resultados apresentadas no início deste capítulo, as técnicas utilizadas neste trabalho são úteis para quaisquer trabalhos futuros que utilizem GREs, seja para a predição de SLFTs ou seja para a predição de sítios de DNA semelhantes aos SLFTs.

Revisão Sistemática

A revisão sistemática apresenta de forma sintetizada os artigos que foram encontrados, dentro dos critérios estabelecidos, e que estão relacionados com as ferramentas de predição de SLFTs. Essa revisão representa uma ferramenta de pesquisa de grande valor para qualquer pesquisador que queira saber rapidamente quais são as ferramentas de predição de SLFTs disponíveis, quais são as suas características, vantagens e desvantagens, etc., seja para esse pesquisador escolher e utilizar essa ferramenta ou seja para ele criar uma nova ferramenta de predição utilizando uma nova abordagem que já não tenha sido utilizada anteriormente.

Validação Cruzada Baseada em Agrupamentos

Este trabalho utilizou uma abordagem de validação cruzada na qual os 5-*folds* foram criados com a utilização de técnicas de agrupamentos. O objetivo foi tanto permitir o aprendizado da real distribuição de sequências da amostra de treinamento quanto avaliar de maneira conservadora as amostras de teste. Os resultados mostraram-se muito bons¹ e essa técnica é bastante recomendável para futuros trabalhos de Bioinformática que utilizem validações cruzadas.

Cálculo da Dependência entre Bases

Os cálculos da dependência entre as bases contidas nas amostras representou, neste trabalho, uma valorosa ferramenta para as análises comparativas entre GREs e PWMs. Essa abordagem é altamente recomendável para futuras abordagens que venham utilizar GREs ou quaisquer outras ferramentas que sejam capazes de considerar a dependência

¹ Uma comparação entre essa abordagem clusterizada e a tradicional é apresentada na monografia "Estudo comparativo de validações cruzadas para a predição de sítios de ligação de fatores de transcrição"(NETO, 2016)

entre as bases.

Ferramenta de Predição

Uma ferramenta de predição de SLFTs, baseada em GREs e PWMs, foi criada e está descrita no Apêndice F. Entretanto, embora ela já seja plenamente funcional, ela pode ser muito aperfeiçoada, incorporando a ela as técnicas discutidas na Revisão Sistemática ou mesmo novas técnicas que venham a ser consideradas. Para isso, o código fonte foi fornecido, juntamente com uma detalhada documentação.

Pesquisas de Bancos de Dados

Um grande esforço foi realizado no sentido de se buscar bancos de dados na internet, além do *Jaspar* 2016, para agregar dados ao banco e enriquecer a análise. Mas, embora muitos dados tenham sido encontrados, eles não puderam ser utilizados pois não havia garantia de que esses dados eram curados e não redundantes. Um outro problema é que muitos bancos de dados apresentam apenas as PWMs já prontas e não apresentam as sequências que geraram as PWMs. Esses dados são fundamentais para o treinamento das GREs. Foram feitas tentativas de se resgatar esses dados do site do BDTNP², mas essas tentativas não prosperaram quando se percebeu que as PWMs publicadas não haviam sido geradas pelos mesmos dados de sequências de **ChIP-chip** que estão publicadas. Uma licença temporária do banco de dados **Transfac** foi obtida, mas foi verificado que esse banco de dados não fornece as sequências que compuseram as PWMs, mas somente as PWMs e diversas ferramentas de análise, todas baseadas nas PWMs. Mas embora todo esse esforço não tenha produzido um resultado prático direto, o fato desses dados não terem sido encontrados, disponíveis na internet, gratuitamente, também serve como uma lição aprendida para futuros projetos.

Contribuições menores

Os estudos realizados neste trabalho sobre a composição das PWMs e GREs poderá ser útil em trabalhos futuros. As pesquisas sobre as diversas técnicas de construção da curva ROC e como estimar o limiar ótimo, assim como o cálculo das medidas de desempenho neste ponto, nos conduziu ao uso de bibliotecas (ou pacotes), tanto no software estatístico

² *Berkeley Drosophila Transcription Network Project*: projeto multidisciplinar da *Berkeley University of California*

R quanto na linguagem `Python`, integrados pelo pacote `rpy2` do `Python`. Esse conhecimento pode vir a ser muito útil em trabalhos futuros.

7.2 Trabalhos futuros

Os excelentes resultados obtidos, tanto para as GREs, quanto para as PWMs, acaba prejudicando uma comparação completa entre esses dois modelos, sobre qual seriam os seus comportamentos em situações reais, *in vivo*. O ideal seria que um novo procedimento de testes fosse desenvolvido para que as GREs e as PWMs não apresentassem um desempenho tão elevado quanto os que foram obtidos neste trabalho. E, nessas condições, com a utilização de amostras onde o nível de dependência entre as bases seja acentuado, espera-se que as GREs apresentem desempenho significativamente superior às PWMs. Uma abordagem poderia ser utilizar, por exemplo, dados simulados ao invés de sítios reais. Ainda, outros métodos de predição podem ser desenvolvidos em combinação com as GREs para aumentar o desempenho da classificação em situações de aplicação real, *in vivo* quando o desempenho não for muito bom.

Os cálculos de dependência entre as bases, apresentados na na seção 5.4 apresentaram uma aparente forte correlação com o tamanho da amostra que, embora não calculada, está evidenciada na Figura 25. Um estudo sobre “o quanto varia a dependência em relação ao tamanho da amostra” pode se mostrar importante. Uma maneira de se fazer esse estudo, seria considerar um terceiro eixo “tamanho da amostra” e repetir toda a simulação variando os tamanhos das amostras artificiais entre, por exemplo, 10 a 1000 sequências³. Essa simulação produziria uma tabela e quando a dependência fosse calculada sobre dados reais, o escore seria corrigido acessando-se essa tabela⁴. A Figura 16 seria um gráfico tridimensional, do “escore de dependência”, calculado por simulação, em função do “nível de dependência” e do “tamanho da amostra”⁵. Já o gráfico da Figura 19, continuaria sendo bidimensional. Esse estudo também pode ajudar a determinar o quanto vale a pena

³ Essas variações poderiam ser como a seguir: 10 a 1000 sequências, sendo de 10 a 100 com passos de 10 e de 100 a 1000, com passos de 100 e, se a simulação mostrar que existe variação significativa de 900 a 1000, então a simulação poderia prosseguir, por exemplo, até 5000 com passos de 1000, e assim por diante. O tamanho dos passos e o número máximo de sequências deveriam ser obtidos experimentalmente a fim de não haver diferenças muito significativas nos valores de escores calculados entre os passos, mas para que também a tabela resultante não ficasse exageradamente grande.

⁴ Se o tamanho da amostra não fosse encontrado na tabela, uma interpolação linear poderia resolver.

⁵ Um quarto eixo também poderia ser adicionado para o tamanho do sítio.

se investir na produção de amostras maiores e, provavelmente, mais custosas de serem obtidas.

A ferramenta de predição pode ser aprimorada com a implementação de uma análise de *cluster*, a exemplo da ferramenta **ClusterDraw** (PAPATSENKO, 2007), no intuito de aumentar a precisão dos resultados (diminuição de falsos positivos). A ideia é que, por exemplo, se em um trecho de DNA de 1.000pb forem encontrados 20 SLFTs, pode-se assumir, com maior nível de confiança, que esses sítios são verdadeiros, do que se fossem encontrados apenas um ou dois SLFTs, neste mesmo trecho. Isso devido ao fato de que os FTs usualmente atuam em conjunto, ligando-se em trechos do DNA chamado *módulos cis-reguladores* e, portanto, é mais provável que eles estejam próximos uns dos outros, o que ajuda a definir o mecanismo da Regulação da Transcrição Gênica.

Conforme comentado na seção 4.1.2, a revisão sistemática poderá ser atualizada e os dados mais atualizados serão disponibilizados na internet. Além disso, nesse mesmo link⁶, pretende-se apresentar a revisão sistemática também em um formato diferente, chamado “Mapeamento Sistemático” (SILVA LEONARDO S.; CORREA, 2017. No prelo.; RODRIGUES DAVI SILVA; DELAMARO, 2017. No prelo.).

Uma nova versão do banco de dados, o **Jaspar 2018** (KHAN et al., 2017), foi disponibilizada quando este trabalho já estava sendo concluído. O processamento dos dados dessa nova versão deste banco de dados poderá evidenciar melhor a questão da dependência entre as bases.

⁶ <https://github.com/tonyfer2017/systematic-review/tree/master> - acesso em 17/12/2017.

Referências⁷

- AHMAD, S. et al. Readout: structure-based calculation of direct and indirect readout energies and specificities for protein–dna recognition. Nucleic acids research, Oxford University Press, v. 34, n. suppl.2, p. W124–W127, 2006. Citado na página 34.
- ALBERTS, B. et al. Biologia molecular da célula. [S.l.]: Artmed Editora, 2009. Citado 4 vezes nas páginas 21, 24, 25 e 26.
- ATTRILL, H. et al. Flybase: establishing a gene group resource for *Drosophila melanogaster*. Nucleic acids research, Oxford Univ Press, p. gkv1046, 2015. Citado na página 60.
- BAIN, D. L. et al. Nuclear receptor structure: implications for function. Annu. Rev. Physiol., Annual Reviews, v. 69, p. 201–220, 2007. Citado na página 34.
- BAUER, A. L. et al. Using sequence-specific chemical and structural properties of DNA to predict transcription factor binding sites. PLoS Comput. Biol., v. 6, n. 11, p. e1001007, 2010. Citado 2 vezes nas páginas 56 e 62.
- BENOS, P. V.; BULYK, M. L.; STORMO, G. D. Additivity in protein–dna interactions: how good an approximation is it? Nucleic acids research, Oxford University Press, v. 30, n. 20, p. 4442–4451, 2002. Citado na página 34.
- BENSON, D. A. et al. Genbank. Nucleic acids research, Oxford Univ Press, p. gks1195, 2012. Citado na página 60.
- BI, Y. et al. Tree-based position weight matrix approach to model transcription factor binding site profiles. PLoS ONE, v. 6, n. 9, p. e24210, 2011. Citado 4 vezes nas páginas 33, 34, 56 e 61.
- BROIN, P. O.; SMITH, T. J.; GOLDEN, A. A. Alignment-free clustering of transcription factor binding motifs using a genetic-k-medoids approach. BMC Bioinformatics, v. 16, p. 22, 2015. Citado 2 vezes nas páginas 54 e 59.
- BULYK, M. L.; JOHNSON, P. L.; CHURCH, G. M. Nucleotides of transcription factor binding sites exert interdependent effects on the binding affinities of transcription factors. Nucleic acids research, Oxford University Press, v. 30, n. 5, p. 1255–1261, 2002. Citado 2 vezes nas páginas 33 e 34.
- CARRASCO, R. C.; ONCINA, J. Learning stochastic regular grammars by means of a state merging method. In: Grammatical Inference and Applications. [S.l.]: Springer, 1994. p. 139–152. Citado na página 36.
- CHIU, D. K.; KOLODZIEJCZAK, T. Inferring consensus structure from nucleic acid sequences. Bioinformatics, Oxford University Press, v. 7, n. 3, p. 347–352, 1991. Citado 2 vezes nas páginas 34 e 74.
- D’HAESELEER, P. What are dna sequence motifs? Nature biotechnology, Nature Publishing Group, v. 24, n. 4, p. 423–425, 2006. Citado 2 vezes nas páginas 21 e 28.

⁷ De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

DIETTERICH, T. Overfitting and undercomputing in machine learning. ACM computing surveys (CSUR), ACM, v. 27, n. 3, p. 326–327, 1995. Citado na página 37.

DROR, I. et al. Covariation between homeodomain transcription factors and the shape of their DNA binding sites. Nucleic Acids Res., v. 42, n. 1, p. 430–441, Jan 2014. Citado 3 vezes nas páginas 55, 57 e 62.

EICHNER, J. et al. TFpredict and SABINE: sequence-based prediction of structural and functional characteristics of transcription factors. PLoS ONE, v. 8, n. 12, p. e82238, 2013. Citado 2 vezes nas páginas 54 e 62.

ERILL, I.; O'NEILL, M. C. A reexamination of information theory-based methods for dna-binding site identification. BMC bioinformatics, BioMed Central, v. 10, n. 1, p. 57, 2009. Citado na página 34.

FAWCETT, T. An introduction to roc analysis. Pattern recognition letters, Elsevier, v. 27, n. 8, p. 861–874, 2006. Citado 2 vezes nas páginas 43 e 44.

FAZIUS, E.; SHELEST, V.; SHELEST, E. SiTaR: a novel tool for transcription factor binding site prediction. Bioinformatics, v. 27, n. 20, p. 2806–2811, Oct 2011. Citado 2 vezes nas páginas 56 e 61.

GOLD, L. et al. Diversity of oligonucleotide functions. Annual review of biochemistry, Annual Reviews 4139 El Camino Way, PO Box 10139, Palo Alto, CA 94303-0139, USA, v. 64, n. 1, p. 763–797, 1995. Citado na página 27.

GROMIHA, M. M. et al. Intermolecular and intramolecular readout mechanisms in protein–dna recognition. Journal of molecular biology, Elsevier, v. 337, n. 2, p. 285–294, 2004. Citado na página 34.

GUO, J.-t.; LOFGREN, S.; FARREL, A. Structure-based prediction of transcription factor binding sites. Tsinghua Science and Technology, TUP, v. 19, n. 6, p. 568–577, 2014. Citado na página 34.

GUTURU, H. et al. Structure-aided prediction of mammalian transcription factor complexes in conserved non-coding elements. Philos. Trans. R. Soc. Lond., B, Biol. Sci., v. 368, n. 1632, p. 20130029, Dec 2013. Citado 2 vezes nas páginas 55 e 60.

HE, Y. et al. CTF: a CRF-based transcription factor binding sites finding system. BMC Genomics, v. 13 Suppl 8, p. S18, 2012. Citado 2 vezes nas páginas 55 e 61.

HOOGE, B. et al. A flexible integrative approach based on random forest improves prediction of transcription factor binding sites. Nucleic Acids Res., v. 40, n. 14, p. e106, Aug 2012. Citado 3 vezes nas páginas 56, 61 e 62.

HOPCROFT, J.; ULLMAN, J.; MOTWANI, R. INTRODUÇÃO A TEORIA DOS AUTOMATOS, LINGUAGENS: E COMPUTAÇÃO. [S.l.]: CAMPUS - RJ, 2002. ISBN 9788535210729. Citado na página 35.

KASHIWABARA, A. Y. et al. Splice site prediction using stochastic regular grammars. Genetics and Molecular Research, v. 6, n. 1, p. 105–115, 2007. Citado 3 vezes nas páginas 36, 71 e 72.

KHAN, A. et al. Jaspar 2018: update of the open-access database of transcription factor binding profiles and its web framework. Nucleic acids research, Oxford University Press, p. 1, 2017. Citado na página 100.

KITCHENHAM, B. Procedures for performing systematic reviews. Keele, UK, Keele University, v. 33, n. 2004, p. 1–26, 2004. Citado na página 50.

KONO, H.; SARAI, A. Structure-based prediction of dna target sites by regulatory proteins. Proteins: Structure, Function, and Bioinformatics, Wiley Online Library, v. 35, n. 1, p. 114–131, 1999. Citado na página 34.

KULAKOVSKIY, I. et al. From binding motifs in chip-seq data to improved models of transcription factor binding sites. Journal of bioinformatics and computational biology, World Scientific, v. 11, n. 01, p. 1340004, 2013. Citado 2 vezes nas páginas 33 e 34.

KULAKOVSKIY, I. V.; MAKEEV, V. J. Dna sequence motif: a jack of all trades for chip-seq data. Adv Protein Chem Struct Biol, v. 91, p. 135–171, 2013. Citado 2 vezes nas páginas 29 e 32.

KVANLI, A.; PAVUR, R.; KEELING, K. Concise managerial statistics. [S.l.]: Cengage Learning, 2005. Citado na página 75.

LAHDESMAKI, H.; RUST, A. G.; SHMULEVICH, I. Probabilistic inference of transcription factor binding from multiple data sources. PLoS ONE, v. 3, n. 3, p. e1820, 2008. Citado 2 vezes nas páginas 57 e 62.

LEON, S. B.-T. de; DAVIDSON, E. H. Modeling the dynamics of transcriptional gene regulatory networks for animal development. Developmental biology, Elsevier, v. 325, n. 2, p. 317–328, 2009. Citado na página 21.

LEVINE, M. Transcriptional enhancers in animal development and evolution. Current Biology, Elsevier, v. 20, n. 17, p. R754–R763, 2010. Citado na página 25.

LI, G. et al. Extensive promoter-centered chromatin interactions provide a topological basis for transcription regulation. Cell, Elsevier, v. 148, n. 1, p. 84–98, 2012. Citado na página 24.

LIFANOV, A. P. et al. Homotypic regulatory clusters in *Drosophila*. Genome research, Cold Spring Harbor Lab, v. 13, n. 4, p. 579–588, 2003. Citado na página 30.

LÓPEZ-RATÓN, M. et al. Optimalcutpoints: an r package for selecting optimal cutpoints in diagnostic tests. Journal of statistical software, Foundation for Open Access Statistics, v. 61, n. 8, p. 1–36, 2014. Citado na página 46.

LÓPEZ-RATÓN, M. et al. Optimalcutpoints: an r package for selecting optimal cutpoints in diagnostic tests. Journal of statistical software, Foundation for Open Access Statistics, v. 61, n. 8, p. 1–36, 2014. Citado na página 73.

LU, C.-C.; YUAN, W.-H.; CHEN, T.-M. Extracting transcription factor binding sites from unaligned gene sequences with statistical models. BMC bioinformatics, BioMed Central, v. 9, n. 12, p. S7, 2008. Citado 2 vezes nas páginas 33 e 34.

- MACHADO-LIMA, A. Laboratorio de geração de classificadores de sequencias. Dissertação (Mestrado) — Universidade de Sao Paulo - IME, 2002. Citado 2 vezes nas páginas 72 e 137.
- MAN, T.-K.; STORMO, G. D. Non-independence of mnt repressor–operator interaction determined by a new quantitative multiple fluorescence relative affinity (qumfra) assay. Nucleic acids research, Oxford University Press, v. 29, n. 12, p. 2471–2478, 2001. Citado na página 34.
- MATHELIER, A. et al. Jaspar 2016: a major expansion and update of the open-access database of transcription factor binding profiles. Nucleic acids research, Oxford Univ Press, p. gkv1176, 2015. Citado 2 vezes nas páginas 58 e 65.
- MATHELIER, A.; WASSERMAN, W. W. The next generation of transcription factor binding site prediction. PLoS Comput. Biol., v. 9, n. 9, p. e1003214, 2013. Citado 5 vezes nas páginas 33, 34, 51, 55 e 61.
- MENEZES, P. B. Linguagens formais e autômatos. [S.l.]: Sagra-Dcluzzato, 1998. Citado na página 35.
- METZ, C. E. Basic principles of roc analysis. In: ELSEVIER. Seminars in nuclear medicine. [S.l.], 1978. v. 8, n. 4, p. 283–298. Citado na página 46.
- MOROZOV, V. Y.; IOSHIKHES, I. P. Optimized position weight matrices in prediction of novel putative binding sites for transcription factors in the Drosophila melanogaster genome. PLoS ONE, v. 8, n. 8, p. e68712, 2013. Citado 2 vezes nas páginas 55 e 61.
- MURTAGH, F.; LEGENDRE, P. Ward’s hierarchical agglomerative clustering method: which algorithms implement ward’s criterion? Journal of Classification, Springer, v. 31, n. 3, p. 274–295, 2014. Citado na página 69.
- NARLIKAR, L.; OVCHARENKO, I. Identifying regulatory elements in eukaryotic genomes. Briefings in Functional Genomics and Proteomics, Oxford University Press, v. 8, n. 4, p. 215–230, 2009. Citado na página 25.
- NAVARRO, C. et al. CisMiner: genome-wide in-silico cis-regulatory module prediction by fuzzy itemset mining. PLoS ONE, v. 9, n. 9, p. e108065, 2014. Citado 2 vezes nas páginas 54 e 61.
- NETO, A. F. Estudo comparativo de validações cruzadas para a predição de sítios de ligação de fatores de transcrição. 39 p. Monografia (Trabalho de conclusão de curso) — Faculdade de Ciências da Saúde de São Paulo, São Paulo, 2016. Citado 2 vezes nas páginas 67 e 97.
- NEWBURGER, D. E.; BULYK, M. L. Uniprobe: an online database of protein binding microarray data on protein–dna interactions. Nucleic acids research, Oxford Univ Press, v. 37, n. suppl 1, p. D77–D82, 2009. Citado na página 58.
- ONCINA, J.; GARCIA, P. Inferring regular languages in polynomial update time. 1992. Citado na página 36.
- PAPATSENKO, D. Clusterdraw web server: a tool to identify and visualize clusters of binding motifs for transcription factors. Bioinformatics, Oxford Univ Press, v. 23, n. 8, p. 1032–1034, 2007. Citado na página 100.

PENNACCHIO, L. A. et al. Enhancers: five essential questions. Nature Reviews. Genetics, NIH Public Access, v. 14, n. 4, p. 288, 2013. Citado na página 25.

PIPER, J. et al. Wellington: a novel method for the accurate identification of digital genomic footprints from DNase-seq data. Nucleic Acids Res., v. 41, n. 21, p. e201, Nov 2013. Citado 2 vezes nas páginas 55 e 60.

POWERS, D. M. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. Bioinfo Publications, 2011. Citado na página 46.

PUJATO, M. et al. Prediction of DNA binding motifs from 3D models of transcription factors; identifying TLX3 regulated genes. Nucleic Acids Res., v. 42, n. 22, p. 13500–13512, Dec 2014. Citado 3 vezes nas páginas 54, 58 e 60.

RAMOS, M. V. M.; NETO, J. J.; VEJA, Í. S. Linguagens Formais: teoria, modelagem e implementação. [S.l.]: Bookman Editora, 2009. Citado na página 35.

RODRIGUES DAVI SILVA; DELAMARO, M. E. C. G. N. F. L. S. Using genetic algorithms in test data generation: a critical systematic mapping. 2017. No prelo. Citado na página 100.

RON, D.; SINGER, Y.; TISHBY, N. The power of amnesia: Learning probabilistic automata with variable memory length. Machine learning, Springer, v. 25, n. 2-3, p. 117–149, 1996. Citado na página 36.

RON, D.; SINGER, Y.; TISHBY, N. On the learnability and usage of acyclic probabilistic finite automata. J. Comput. Syst. Sci., v. 56, p. 133–152, 1998. Citado 4 vezes nas páginas 36, 37, 38 e 71.

SEDGEWICK, R.; WAYNE, K. Algorithms. [S.l.]: Addison-Wesley Professional, 2011. Citado na página 37.

SETUBAL, J. C.; MEIDANIS, J. Introduction to computational molecular biology. [S.l.]: PWS Pub., 1997. Citado na página 69.

SHANE, L.; ALVIN, F. et al. Structure-based prediction of transcription factor binding sites. Tsinghua Science and Technology, v. 19, n. 6, p. 568–577, 2015. Citado 2 vezes nas páginas 34 e 78.

SIGGERS, T. W.; HONIG, B. Structure-based prediction of c2h2 zinc-finger binding specificity: sensitivity to docking geometry. Nucleic acids research, Oxford University Press, v. 35, n. 4, p. 1085–1097, 2007. Citado na página 34.

SILVA LEONARDO S.; CORREA, C. G. N. R. N. F. L. S. A systematic mapping of temporal data visualization: Categories, approaches and methods. 2017. No prelo. Citado na página 100.

SING, T. et al. Rocr: visualizing classifier performance in r. Bioinformatics, v. 21, n. 20, p. 7881, 2005. Disponível em: <<http://rocr.bioinf.mpi-sb.mpg.de>>. Citado na página 73.

SIPSER, M.; QUEIROZ, R. J. G. B. de. Introdução à teoria da computação. [S.l.]: Thomson Learning, 2007. Citado 2 vezes nas páginas 35 e 36.

- STEWART, A. J.; HANNENHALLI, S.; PLOTKIN, J. B. Why transcription factor binding sites are ten nucleotides long. Genetics, Genetics Soc America, v. 192, n. 3, p. 973–985, 2012. Citado na página 21.
- STORMO, G. D. et al. Use of the ‘perceptron’ algorithm to distinguish translational initiation sites in e. coli. Nucleic Acids Research, Oxford Univ Press, v. 10, n. 9, p. 2997–3011, 1982. Citado na página 29.
- TALEBZADEH, M.; ZARE-MIRAKABAD, F. Transcription factor binding sites prediction based on modified nucleosomes. PLoS ONE, v. 9, n. 2, p. e89226, 2014. Citado 4 vezes nas páginas 54, 57, 59 e 94.
- TAN, M. et al. An information transmission model for transcription factor binding at regulatory DNA sites. Theor Biol Med Model, v. 9, p. 19, 2012. Citado 2 vezes nas páginas 56 e 62.
- TOMOVIC, A.; OAKELEY, E. J. Position dependencies in transcription factor binding sites. Bioinformatics, Oxford University Press, v. 23, n. 8, p. 933–941, 2007. Citado 3 vezes nas páginas 33, 34 e 74.
- VERMONT, J. et al. Strategies for graphical threshold determination. Computer methods and programs in biomedicine, Elsevier, v. 35, n. 2, p. 141–150, 1991. Citado na página 46.
- VIEIRA, D. Geração de Classificadores de Sequências Genéticas Utilizando Inferência de Linguagens Regulares. Dissertação (Mestrado) — Universidade de São Paulo - IME, 2004. Citado 4 vezes nas páginas 36, 38, 72 e 137.
- WRIGHT, H. et al. Occupancy classification of position weight matrix-inferred transcription factor binding sites. PLoS ONE, v. 6, n. 11, p. e26160, 2011. Citado 3 vezes nas páginas 56, 61 e 62.
- XU, T. et al. Base-resolution methylation patterns accurately predict transcription factor bindings in vivo. Nucleic Acids Res., v. 43, n. 5, p. 2757–2766, Mar 2015. Citado 2 vezes nas páginas 54 e 59.
- YATES, A. et al. Ensembl 2016. Nucleic Acids Research, v. 44, n. D1, p. D710–D716, 2016. Citado na página 60.
- YE, Z. et al. Circularlogo: A lightweight web application to visualize intra-motif dependencies. BMC bioinformatics, BioMed Central, v. 18, n. 1, p. 269, 2017. Citado na página 74.
- ZERBINO, D. R. et al. Ensembl 2018. Nucleic Acids Research, p. gkx1098, 2017. Disponível em: <http://dx.doi.org/10.1093/nar/gkx1098>. Citado na página 93.
- ZHAO, Y. et al. Improved models for transcription factor binding site identification using nonindependent interactions. Genetics, Genetics Soc America, v. 191, n. 3, p. 781–790, 2012. Citado na página 34.
- ZHONG, S.; HE, X.; BAR-JOSEPH, Z. Predicting tissue specific transcription factor binding sites. BMC Genomics, v. 14, p. 796, 2013. Citado 2 vezes nas páginas 54 e 60.

Apêndices

Apêndice A – Escores de dependência por arquivo

Neste apêndice é apresentado na tabela 6 todos os arquivos obtidos no banco **Jaspar 2016** e seus respectivos escores de dependência, calculados conforme descrito nas seções 5.4 e 5.4. A tabela foi ordenada por ordem crescente do escore.

Além disso, como também foi discutido na seção 5.4, foram produzido 11 arquivos simulando diferentes níveis de dependências, misturando-se dados forçosamente gerados com dependência e dados aleatórios, compondo-se 11 diferentes proporções.

Esses arquivos, gerados artificialmente, também tiveram seus escores calculados pelo mesmo programa que calculou os escores dos arquivos do **Jaspar 2016** e estão também apresentados na tabela 6, juntamente com os demais arquivos.

A ideia é que esses arquivos, com seus nomes grifados na tabela em vermelho com fundo amarelo, sirvam como referência para demarcarem faixas dos níveis de dependência existentes nos arquivos do **Jaspar 2016**. Essas faixas são denominadas "intervalos de classe" ou simplesmente "classes". Assim, por exemplo, os arquivos que estão entre os arquivos **dep_030** e **dep_040** (297 a 334 na numeração da tabela), possuem um nível de dependência entre as bases superior a 30% e inferior a 40% pois eles possuem um escore superior ao do arquivo que foi produzido com 30% de dependências e escore inferior ao do arquivo produzido com 40% de dependência.

Tabela 6 – Escore de dependência de dinucleotídeos dos arquivos do Jaspar 2016 juntamente com arquivos "dep_xxx" criados artificialmente.

	Arquivo FT	Escore		Arquivo FT	Escore
1	MA0461.1.sites	0,00065	38	MA0520.1.sites	0,00369
2	MA0143.3.sites	0,00140	39	MA0482.1.sites	0,00376
3	MA0161.1.sites	0,00172	40	MA0499.1.sites	0,00378
4	MA0526.1.sites	0,00188	41	MA0462.1.sites	0,00380
5	MA0529.1.sites	0,00210	42	MA0500.1.sites	0,00380
6	MA0491.1.sites	0,00226	43	MA0480.1.sites	0,00384
7	MA0105.3.sites	0,00234	44	MA0052.2.sites	0,00385
8	MA0154.2.sites	0,00234	45	MA0483.1.sites	0,00387
9	MA0477.1.sites	0,00238	46	MA0521.1.sites	0,00391
10	MA0506.1.sites	0,00245	47	MA0503.1.sites	0,00391
11	MA0079.3.sites	0,00249	48	MA0474.1.sites	0,00392
12	MA0471.1.sites	0,00253	49	MA0488.1.sites	0,00400
13	MA0489.1.sites	0,00255	50	MA0104.3.sites	0,00401
14	MA0490.1.sites	0,00261	51	MA0510.1.sites	0,00403
15	dep_000	0,00268	52	MA0093.2.sites	0,00403
16	MA0516.1.sites	0,00268	53	MA0003.2.sites	0,00404
17	MA0533.1.sites	0,00271	54	MA0058.2.sites	0,00408
18	MA0452.2.sites	0,00280	55	MA0083.2.sites	0,00411
19	MA0469.1.sites	0,00284	56	MA0024.2.sites	0,00414
20	MA0035.3.sites	0,00288	57	MA0100.2.sites	0,00414
21	MA0076.2.sites	0,00305	58	MA0512.1.sites	0,00415
22	MA0473.1.sites	0,00307	59	MA0519.1.sites	0,00421
23	MA0478.1.sites	0,00317	60	MA0513.1.sites	0,00426
24	MA0508.1.sites	0,00319	61	MA0940.1.sites	0,00429
25	MA0103.2.sites	0,00319	62	MA0494.1.sites	0,00443
26	MA0523.1.sites	0,00331	63	MA0518.1.sites	0,00444
27	MA0470.1.sites	0,00331	64	MA0507.1.sites	0,00444
28	MA0147.2.sites	0,00332	65	MA0137.3.sites	0,00445
29	MA0476.1.sites	0,00336	66	MA0036.2.sites	0,00446
30	MA0497.1.sites	0,00337	67	MA0522.1.sites	0,00453
31	MA0095.2.sites	0,00340	68	MA0050.2.sites	0,00458
32	MA0531.1.sites	0,00345	69	MA0475.1.sites	0,00471
33	MA0536.1.sites	0,00346	70	MA0537.1.sites	0,00482
34	MA0479.1.sites	0,00348	71	MA0514.1.sites	0,00483
35	MA0465.1.sites	0,00351	72	MA0463.1.sites	0,00485
36	MA0102.3.sites	0,00353	73	MA0098.2.sites	0,00498
37	MA0216.2.sites	0,00368	74	MA0501.1.sites	0,00500

Tabela 6 – (continuação)

	Arquivo FT	Escore		Arquivo FT	Escore
75	MA0498.1.sites	0,00510	112	MA0532.1.sites	0,01361
76	MA0485.1.sites	0,00511	113	MA0014.2.sites	0,01433
77	MA0467.1.sites	0,00521	114	MA0481.1.sites	0,01488
78	MA0464.1.sites	0,00528	115	MA0060.2.sites	0,01496
79	MA0466.1.sites	0,00541	116	MA0149.1.sites	0,01591
80	MA0530.1.sites	0,00542	117	MA0237.2.sites	0,01626
81	MA0527.1.sites	0,00547	118	MA0535.1.sites	0,01663
82	MA0509.1.sites	0,00550	119	MA0189.1.sites	0,02058
83	MA0144.2.sites	0,00551	120	MA0060.1.sites	0,02079
84	MA0247.2.sites	0,00558	121	MA0036.1.sites	0,02210
85	MA0504.1.sites	0,00586	122	MA0167.1.sites	0,02266
86	MA0517.1.sites	0,00597	123	MA0037.1.sites	0,02313
87	MA0493.1.sites	0,00614	124	MA0110.1.sites	0,02329
88	MA0106.2.sites	0,00631	125	MA0089.1.sites	0,02364
89	MA0511.1.sites	0,00644	126	MA0093.1.sites	0,02376
90	MA0505.1.sites	0,00645	127	MA0015.1.sites	0,02434
91	MA0562.1.sites	0,00685	128	MA0098.1.sites	0,02464
92	MA0546.1.sites	0,00693	129	MA0538.1.sites	0,02518
93	MA0543.1.sites	0,00704	130	MA0072.1.sites	0,02520
94	MA0544.1.sites	0,00704	131	MA0052.1.sites	0,02567
95	MA0150.2.sites	0,00715	132	MA0555.1.sites	0,02634
96	MA0545.1.sites	0,00793	133	MA0176.1.sites	0,02656
97	MA0258.1.sites	0,00825	134	MA0215.1.sites	0,02699
98	MA0515.1.sites	0,00872	135	MA0152.1.sites	0,02792
99	MA0472.1.sites	0,00893	136	MA0257.1.sites	0,02811
100	MA0547.1.sites	0,00907	137	MA0240.1.sites	0,02850
101	MA0563.1.sites	0,00934	138	MA0005.1.sites	0,02874
102	MA1012.1.sites	0,00961	139	MA0081.1.sites	0,02888
103	MA0468.1.sites	0,01135	140	MA0210.1.sites	0,02894
104	dep_010	0,01165	141	MA0001.1.sites	0,02897
105	MA0259.1.sites	0,01170	142	MA0217.1.sites	0,02903
106	MA0486.1.sites	0,01180	143	MA0444.1.sites	0,02907
107	MA0140.2.sites	0,01201	144	MA0205.1.sites	0,02915
108	MA0541.1.sites	0,01220	145	MA0132.1.sites	0,02926
109	MA0502.1.sites	0,01259	146	MA0105.1.sites	0,02937
110	MA0548.1.sites	0,01348	147	MA0080.1.sites	0,02964
111	MA0534.1.sites	0,01359	148	MA0087.1.sites	0,02991

Tabela 6 – (continuação)

	Arquivo FT	Escore		Arquivo FT	Escore
149	MA0009.1.sites	0,03041	186	MA0188.1.sites	0,03502
150	MA0016.1.sites	0,03048	187	MA0091.1.sites	0,03523
151	MA0182.1.sites	0,03048	188	MA0021.1.sites	0,03535
152	MA0450.1.sites	0,03075	189	MA0020.1.sites	0,03539
153	MA0103.1.sites	0,03077	190	MA0094.2.sites	0,03573
154	MA0459.1.sites	0,03086	191	MA0191.1.sites	0,03578
155	MA0186.1.sites	0,03102	192	MA0076.1.sites	0,03592
156	MA0196.1.sites	0,03122	193	MA0206.1.sites	0,03602
157	MA0179.1.sites	0,03142	194	MA0130.1.sites	0,03611
158	MA0241.1.sites	0,03150	195	MA0154.1.sites	0,03646
159	MA0245.1.sites	0,03186	196	MA0097.1.sites	0,03649
160	MA0165.1.sites	0,03191	197	MA0242.1.sites	0,03697
161	MA0250.1.sites	0,03205	198	MA0226.1.sites	0,03697
162	MA0177.1.sites	0,03213	199	MA0197.1.sites	0,03722
163	MA0048.1.sites	0,03229	200	MA0025.1.sites	0,03795
164	dep_020	0,03240	201	MA0125.1.sites	0,03795
165	MA0448.1.sites	0,03252	202	MA0026.1.sites	0,03806
166	MA0458.1.sites	0,03253	203	MA0251.1.sites	0,03830
167	MA0197.2.sites	0,03254	204	MA0220.1.sites	0,03863
168	MA0236.1.sites	0,03257	205	MA0195.1.sites	0,03911
169	MA0029.1.sites	0,03284	206	MA0002.1.sites	0,03938
170	MA0169.1.sites	0,03293	207	MA0038.1.sites	0,03995
171	MA0238.1.sites	0,03305	208	MA0083.1.sites	0,04011
172	MA0071.1.sites	0,03311	209	MA0183.1.sites	0,04038
173	MA0221.1.sites	0,03341	210	MA0247.1.sites	0,04047
174	MA0175.1.sites	0,03343	211	MA0209.1.sites	0,04048
175	MA0452.1.sites	0,03372	212	MA0115.1.sites	0,04064
176	MA0187.1.sites	0,03378	213	MA0225.1.sites	0,04068
177	MA0198.1.sites	0,03379	214	MA0171.1.sites	0,04069
178	MA0164.1.sites	0,03382	215	MA0078.1.sites	0,04103
179	MA0136.1.sites	0,03397	216	MA0255.1.sites	0,04137
180	MA0118.1.sites	0,03424	217	MA0107.1.sites	0,04180
181	MA0203.1.sites	0,03453	218	MA0228.1.sites	0,04185
182	MA0192.1.sites	0,03476	219	MA0095.1.sites	0,04211
183	MA0224.1.sites	0,03480	220	MA0077.1.sites	0,04211
184	MA0219.1.sites	0,03487	221	MA0451.1.sites	0,04216
185	MA0460.1.sites	0,03497	222	MA0231.1.sites	0,04238

Tabela 6 – (continuação)

	Arquivo FT	Escore		Arquivo FT	Escore
223	MA0119.1.sites	0,04294	260	MA0070.1.sites	0,05129
224	MA0253.1.sites	0,04317	261	MA0123.1.sites	0,05182
225	MA0559.1.sites	0,04318	262	MA0447.1.sites	0,05196
226	MA0256.1.sites	0,04378	263	MA0222.1.sites	0,05209
227	MA0138.1.sites	0,04381	264	MA0092.1.sites	0,05232
228	MA0184.1.sites	0,04391	265	MA0135.1.sites	0,05321
229	MA0230.1.sites	0,04392	266	MA0178.1.sites	0,05368
230	MA0229.1.sites	0,04402	267	MA0105.2.sites	0,05373
231	MA0558.1.sites	0,04415	268	MA0202.1.sites	0,05373
232	MA0443.1.sites	0,04441	269	MA0040.1.sites	0,05401
233	MA0235.1.sites	0,04451	270	MA0456.1.sites	0,05473
234	MA0150.1.sites	0,04459	271	MA0058.1.sites	0,05487
235	MA0218.1.sites	0,04459	272	MA0445.1.sites	0,05552
236	MA0156.1.sites	0,04460	273	MA0457.1.sites	0,05661
237	MA0232.1.sites	0,04505	274	MA0446.1.sites	0,05680
238	MA0030.1.sites	0,04517	275	MA0157.1.sites	0,05688
239	MA0031.1.sites	0,04517	276	MA0556.1.sites	0,05703
240	MA0166.1.sites	0,04550	277	MA0041.1.sites	0,05704
241	MA0127.1.sites	0,04609	278	MA0207.1.sites	0,05717
242	MA0170.1.sites	0,04618	279	MA0550.1.sites	0,05800
243	MA0181.1.sites	0,04626	280	MA0137.1.sites	0,05826
244	MA0162.1.sites	0,04737	281	MA0213.1.sites	0,05984
245	MA0082.1.sites	0,04739	282	MA0049.1.sites	0,06050
246	MA0214.1.sites	0,04825	283	MA0549.1.sites	0,06060
247	MA0084.1.sites	0,04866	284	MA0069.1.sites	0,06077
248	MA0185.1.sites	0,04926	285	MA0018.1.sites	0,06143
249	MA0540.1.sites	0,04950	286	MA0024.1.sites	0,06167
250	MA0126.1.sites	0,04963	287	MA0059.1.sites	0,06177
251	MA0065.1.sites	0,04981	288	MA0552.1.sites	0,06219
252	MA0019.1.sites	0,05009	289	MA0067.1.sites	0,06221
253	MA0028.1.sites	0,05025	290	MA0122.1.sites	0,06226
254	MA0008.1.sites	0,05033	291	MA0254.1.sites	0,06255
255	MA0237.1.sites	0,05041	292	MA0239.1.sites	0,06302
256	MA0034.1.sites	0,05049	293	MA0047.1.sites	0,06344
257	MA0554.1.sites	0,05058	294	MA0159.1.sites	0,06393
258	MA0131.1.sites	0,05119	295	MA0054.1.sites	0,06406
259	MA0216.1.sites	0,05122	296	dep_030	0,06482

Tabela 6 – (continuação)

	Arquivo FT	Escore		Arquivo FT	Escore
297	MA0163.1.sites	0,06499	334	MA0012.1.sites	0,10267
298	MA0193.1.sites	0,06535	335	dep_040	0,10601
299	MA0129.1.sites	0,06541	336	MA0079.1.sites	0,10617
300	MA0043.1.sites	0,06621	337	MA0112.1.sites	0,10779
301	MA0074.1.sites	0,06622	338	MA0560.1.sites	0,10818
302	MA0454.1.sites	0,06626	339	MA0010.1.sites	0,10821
303	MA0044.1.sites	0,06652	340	MA0045.1.sites	0,10880
304	MA0449.1.sites	0,06696	341	MA0088.1.sites	0,10928
305	MA0063.1.sites	0,06724	342	MA0014.1.sites	0,14436
306	MA0153.1.sites	0,06814	343	dep_050	0,15744
307	MA0007.1.sites	0,07036	344	MA0116.1.sites	0,16857
308	MA0252.1.sites	0,07103	345	MA0080.2.sites	0,18864
309	MA0248.1.sites	0,07289	346	MA0553.1.sites	0,20490
310	MA0057.1.sites	0,07327	347	dep_060	0,21785
311	MA0023.1.sites	0,07386	348	MA0079.2.sites	0,22068
312	MA0051.1.sites	0,07397	349	dep_070	0,29139
313	MA0101.1.sites	0,07496	350	dep_080	0,37603
314	MA0073.1.sites	0,07546	351	dep_090	0,47899
315	MA0062.1.sites	0,07644	352	dep_100	0,59871
316	MA0174.1.sites	0,07688			
317	MA0011.1.sites	0,07745			
318	MA0017.1.sites	0,07859			
319	MA0066.1.sites	0,07975			
320	MA0106.1.sites	0,07977			
321	MA0244.1.sites	0,08119			
322	MA0168.1.sites	0,08188			
323	MA0173.1.sites	0,08261			
324	MA0085.1.sites	0,08537			
325	MA0561.1.sites	0,08980			
326	MA0155.1.sites	0,09013			
327	MA0249.1.sites	0,09177			
328	MA0111.1.sites	0,09213			
329	MA0117.1.sites	0,09235			
330	MA0102.1.sites	0,09354			
331	MA0120.1.sites	0,09538			
332	MA0121.1.sites	0,09728			
333	MA0018.2.sites	0,10264			

Apêndice B – Correlações entre escores de dependência e as medidas de desempenho

Aqui são apresentados os resultados do cálculo de todas as correlações entre o escore da dependência entre as bases e todas as medidas de desempenho.

Por exemplo, para a primeira medida de desempenho da tabela 7 (AUC GRE N1), foram calculados o coeficiente de correlação de Pearson (-0.284386371) e o p-valor ($9.14362E - 08$) para a medida de desempenho AUC, do modelo N1 e para a gramática (GRE). Esse resultado pode ser visualizado em gráficos, como exemplificado na Figura 29.

Figura 29 – Gráfico de dispersão dos pontos relacionando o escore de dependência e a medida AUC da gramática (GRE) para o modelo N1.

Fonte: Antonio Ferrão Neto, 2017

Tabela 7 – Resultados dos cálculos dos coeficientes de correlação de Pearson e o seu respectivo p-valor para cada medida de desempenho.

MEDIDA DESEMPENHO	COEF. CORREL.	P-VALOR
AUC GRE N1	-0,284386371	9,14362E-08
AUC GRE N2	-0,293094965	3,507E-08
AUC GRE N3	-0,378516745	4,65325E-13
AUC PWM N1	-0,55222279	1,29798E-28
AUC PWM N2	-0,557796238	2,82375E-29
AUC PWM N3	-0,561260362	1,07836E-29
ERR: 0.1 GRE N1	0,391502193	6,15796E-14
ERR: 0.1 GRE N2	0,399420395	1,71614E-14
ERR: 0.1 GRE N3	0,407755089	4,30798E-15
ERR: 0.1 PWM N1	0,501213638	4,32816E-23
ERR: 0.1 PWM N2	0,502131969	3,50661E-23
ERR: 0.1 PWM N3	0,545350425	8,18793E-28
ERR: Youden GRE N1	0,412872564	1,80845E-15
ERR: Youden GRE N2	0,405065622	6,75791E-15
ERR: Youden GRE N3	0,40718382	4,74193E-15
ERR: Youden PWM N1	0,489885575	5,51517E-22
ERR: Youden PWM N2	0,487720418	8,875E-22
ERR: Youden PWM N3	0,525298353	1,39569E-25
F: 0.1 GRE N1	-0,5783441	7,89648E-32
F: 0.1 GRE N2	-0,586782827	6,24593E-33
F: 0.1 GRE N3	-0,535533368	1,05769E-26
F: 0.1 PWM N1	-0,652327924	1,0374E-42
F: 0.1 PWM N2	-0,653452812	6,71038E-43
F: 0.1 PWM N3	-0,631512258	2,38236E-39
F: Youden GRE N1	-0,577650353	9,69613E-32
F: Youden GRE N2	-0,586794172	6,22435E-33
F: Youden GRE N3	-0,535497032	1,06759E-26
F: Youden PWM N1	-0,649749088	2,79685E-42
F: Youden PWM N2	-0,649836741	2,70457E-42
F: Youden PWM N3	-0,629731251	4,49521E-39
FPR: 0.1 GRE N1	0,390946848	6,72665E-14
FPR: 0.1 GRE N2	0,398592909	1,96443E-14
FPR: 0.1 GRE N3	0,407018553	4,87527E-15
FPR: 0.1 PWM N1	0,500389245	5,22557E-23
FPR: 0.1 PWM N2	0,50155435	4,00331E-23
FPR: 0.1 PWM N3	0,544627303	9,91457E-28
FPR: Youden GRE N1	0,411931473	2,1238E-15
FPR: Youden GRE N2	0,403744171	8,41869E-15
FPR: Youden GRE N3	0,406012836	5,7696E-15
FPR: Youden PWM N1	0,488697849	7,16274E-22
FPR: Youden PWM N2	0,486474073	1,1653E-21
FPR: Youden PWM N3	0,523938632	1,9534E-25
PREC: 0.1 GRE N1	-0,551628161	1,5248E-28
PREC: 0.1 GRE N2	-0,557121993	3,40117E-29
PREC: 0.1 GRE N3	-0,499780745	6,00332E-23
PREC: 0.1 PWM N1	-0,627007504	1,1776E-38
PREC: 0.1 PWM N2	-0,62651475	1,4003E-38
PREC: 0.1 PWM N3	-0,600495274	8,60728E-35
PREC: Youden GRE N1	-0,550300204	2,18228E-28
PREC: Youden GRE N2	-0,556589957	3,93785E-29
PREC: Youden GRE N3	-0,499751136	6,04395E-23
PREC: Youden PWM N1	-0,624412867	2,92136E-38
PREC: Youden PWM N2	-0,62307384	4,65343E-38
PREC: Youden PWM N3	-0,598211199	1,78251E-34
TPR: 0.1 GRE N1	-0,157260488	0,003597267
TPR: 0.1 GRE N2	-0,181276459	0,000770729
TPR: 0.1 GRE N3	-0,217947241	4,93314E-05
TPR: 0.1 PWM N1	-0,38972013	8,17113E-14
TPR: 0.1 PWM N2	-0,379598481	3,94517E-13
TPR: 0.1 PWM N3	-0,365469992	3,24888E-12
TPR: Youden GRE N1	-0,101302746	0,061676254
TPR: Youden GRE N2	-0,125829554	0,02010914
TPR: Youden GRE N3	-0,169654153	0,001665504
TPR: Youden PWM N1	-0,365804319	3,09441E-12
TPR: Youden PWM N2	-0,367224901	2,51426E-12
TPR: Youden PWM N3	-0,359143288	8,08121E-12

Fonte: Antonio Ferrão Neto, 2017.

Apêndice C – Códigos

Este apêndice apresenta alguns trechos de código referenciados no texto principal.

Código C.1 – Cálculo do escore de alinhamento entre duas sequências (seq1 e seq2).

```

1 def calc_score(seq1, seq2):
2     match = 2
3     mismatch = -1
4     scoring = swalign.NucleotideScoringMatrix(match, mismatch)
5     sw = swalign.LocalAlignment(scoring)
6     alignment = sw.align(seq1, seq2)
7     return round((1.0 - alignment.identity), 3)

```

Código C.2 – Exemplo de clusterização hierárquica no R.

```

1 mat<-data.matrix(read.table("MA0107.1.sites.cut.fasta.Out.txt",
2                             fill=TRUE, col.names=paste("S", 1:14)))
3 clusters <- hclust(as.dist(mat), method = "ward.D")
4 clusterCut <- cutree(clusters, 5)
5 print("MA0107.1.sites.cut.fasta")
6 print(clusterCut)

```

Critério ROC01

Segundo o critério ROC01, o limiar ideal é dado pelo ponto da curva ROC com menor distância ao ponto (0,1), que representa o classificador ideal.

O trecho de código Python C.3 a seguir mostra como o ponto ótimo foi determinado, segundo o critério ROC01:

Código C.3 – ROC01: Cálculo da mínima distância entre a curva ROC e o ponto (0,1).

```

1     # dict_perf e' um dicionario que contem os resultados da curva ROC
2     # obtidos pela funcao "performance" do pacote ROCR.
3     # As abscissas dos pontos da curva ROC estao armazenadas
4     # na lista dict_perf['fpr'] e as ordenadas estao armazenadas na
5     # lista dict_perf['tpr'].
6     # Essas duas listas estao dentro do dicionario dict_perf, possuindo
7     # o mesmo tamanho e seus dados estao emparelhados, ou seja,
8     # cada elemento com indice "idx" na lista dict_perf['tpr'][idx],
9     # possui um correspondente elemento, com o mesmo indice "idx"
10    # na lista dict_perf['fpr'][idx], formando os pares de valores (x,y),
11    # que sao os pontos da curva ROC.

```

```

12 # Esses pontos podem ser percorridos por meio de um "loop" sobre uma
13 # dessas duas listas. Neste caso, foi utilizada a dict_perf['tpr']:
14 for idx, item in enumerate(dict_perf['tpr']):
15     tpr = dict_perf['tpr'][idx]
16     fpr = dict_perf['fpr'][idx]
17     # Calculo do ponto cuja distancia ao ponto (0,1) e minimo:
18     # Distancia entre os pontos A(xA,yA) e B(xB,yB):
19     # dAB = SQRT( (xB-xA)^2 + (yB-yA)^2 )
20     # Neste caso: xA = 0, yA = 1; xB = fpr, yB = tpr
21     dAB = math.sqrt(pow(fpr,2) + pow(tpr-1,2))
22     if dAB <= d_0_1:
23         d_0_1 = dAB
24         tpr_0_1 = tpr
25         fpr_0_1 = fpr

```

A variável `d_0_1` guarda o valor da mínima distância e as variáveis `tpr_0_1` e `fpr_0_1` guardam os valores da abscissa e ordenada, respectivamente, do ponto ótimo na curva ROC para o critério ROC01.

Critério Youden

Similarmente, a determinação do ponto ótimo, segundo o critério Youden foi efetuado conforme o trecho de código C.4.

Código C.4 – Youden: Cálculo da máxima diferença entre TPR e FPR.

```

1 # Youden: maxima diferenca entre tpr do ponto e o y(fpr) = fpr
2 # ou seja, para o ponto (tpr, fpr) o indice Youden J sera
3 # ( tpr - fpr ):
4 youden = tpr - fpr
5 if youden > max_youden:
6     max_youden = youden
7     tpr_youden = tpr
8     fpr_youden = fpr

```

A variável `max_youden` guarda o valor da máxima distância e as variáveis `tpr_youden` e `fpr_youden` guardam os valores da abscissa e da ordenada, respectivamente, do ponto ótimo na curva ROC para o critério Youden.

Por questões de simplicidade e performance, ambos os pontos ótimos, para critérios ROC01 e Youden foram calculados utilizando-se o mesmo "loop" descrito no trecho de código ROC01 (`for idx, item in enumerate(dict_perf['tpr']):`).

Apêndice D – Exemplo de formato de saída da validação cruzada

A Figura 30 mostra um exemplo de formato de saída da validação cruzada da PWM para uma das amostras, neste caso, o arquivo `MA0001.1.sites` do banco `Jaspar 2016`. O arquivo é dividido em cinco blocos de *folds*, iniciados com o símbolo “#” na primeira posição da linha. A seguir, iniciam-se as sequências positivas, identificadas com o símbolo “+” na primeira posição da linha (na figura, essas linhas estão em verde). A primeira coluna mostra a sequência que foi testada, a segunda, a terceira e a quarta colunas apresentam o resultado do score para o primeiro, segundo e terceiro modelos nulos, respectivamente. Logo após as sequências positivas, iniciam-se as sequências negativas, identificadas com o símbolo “-” na primeira posição da linha (na figura, essas linhas estão em vermelho). Essas sequências negativas utilizaram as mesmas sequências de treinamento que as sequências positivas para o *fold* em questão. As colunas aparecem ligeiramente desalinhadas porque foram separadas por [TABS].

A Figura 31 mostra um exemplo de formato de saída da validação cruzada da GRE para a mesma amostra do exemplo da PWM (arquivo `MA0001.1.sites` do banco `Jaspar 2016`). O formato do arquivo de saída é semelhante, exceto que, para cada *fold* foram produzidos 100 diferentes combinações de parâmetros `Lapfta`. Esses blocos, dentro de cada *fold*, foram identificados com o símbolo “*” na primeira posição da linha, seguido pelos valores dos parâmetros `p1` (sempre igual a 5), `p2` (variando de 0.1 a 1.0) e `p3` (variando de 0.01 a 0.10).

Figura 30 – Exemplo mostrando o formato do arquivo de saída da validação cruzada para a PWM.

# Fold 1				
+CCATTAATGG	12.4340955579	12.1853263242	12.14825197	
+CCGTTTTAG	8.02324501181	7.74212154832	5.66869329686	
+CCATTCTGG	8.35958556765	8.85726535842	6.87841172556	
+CCTAAAGTGT	3.58990046021	3.35341988757	3.31332402755	
+CCCTTTTTCG	2.97885089246	3.40854721409	-3.07205772	
.	.	.	.	
.	.	.	.	
+CCATAAATAG	15.0107727538	13.9751708136	12.657072546	
+CAATAATTAG	9.24767524672	7.46892851371	6.29419656816	
+CCAAATAAGG	10.4642721186	10.2248298795	8.1477981066	
-CTCAGTTAGA	-11.9604020486	-12.3402521993	-12.3225746922	
-TATAGGTCGT	-23.0569427088	-24.0931275876	-23.9287793097	
-TCAGATAGCT	-8.62821104145	-8.97305376988	-8.98436888793	
-AATGTACTAA	-16.1384193509	-17.3150118462	-17.2960875816	
.	.	.	.	
.	.	.	.	
-AGGTACTCTA	-27.8181026836	-27.916678749	-27.9620777294	
-AAACAGCAAT	-18.5611497966	-19.7116160594	-22.7311505685	
-TAGTTACAAC	-12.4727082842	-13.5419748242	-14.0007981069	
-ACTGATTGAC	-11.141349456	-11.232968871	-11.301553749	
# Fold 2				
+CCATAAATGG	13.4002427962	13.1479044156	12.6272333126	
+CTAAAAATAG	12.5681643678	10.7472380476	8.33042059559	
+CCAAACTAG	11.2428456837	10.9730106691	8.1504445242	
+CCATTTTATAG	12.3732660786	11.3412199963	10.0363768738	
.	.	.	.	
.	.	.	.	
-TAGTTACAAC	-10.17202551	-11.210581598	-11.6826978664	
-ACTGATTGAC	-9.75709624027	-9.73865377738	-9.83595029642	
# Fold 3				
+CCTAAAATGG	10.6346854073	10.3957251303	9.8575131211	
+CCAAAATTAG	12.9978306093	11.9642167931	10.6369822015	
+CTGTAAATAG	8.09826694489	7.02304477005	6.5523415342	
.	.	.	.	
.	.	.	.	
-AAACAGCAAT	-19.1832455219	-20.3259625107	-23.3182840871	
-TAGTTACAAC	-12.3653788368	-13.4107956203	-13.8805047182	
-ACTGATTGAC	-10.5760031095	-10.5611313775	-10.6574763879	
# Fold 4				
+CTATATATAG	13.1129142652	11.2881082468	10.3667595602	
+CCTAATATAG	11.3749730812	10.3469414486	9.86420760603	
+CTCAAAATAG	7.68483377302	6.60842865169	5.29169446181	
.	.	.	.	
.	.	.	.	
-AAACAGCAAT	-19.3502152442	-20.5074980852	-23.5548026025	
-TAGTTACAAC	-11.289316514	-12.3375449663	-12.8277070583	
-ACTGATTGAC	-11.2775514285	-11.2731114752	-11.3667884153	
# Fold 5				
+CCATTATATAG	13.6743063486	12.63699156	12.15873383	
+CCATACATGG	8.74056566806	9.19936982039	8.5056909388	
+CTAAAAGAAG	6.23482639767	5.18560372879	2.00687863957	
.	.	.	.	
.	.	.	.	
-CCGTTTTATA	-9.1526823414	-9.79299528859	-11.6323855023	
-ACATAACGAG	-1.73184746447	-1.69121500444	-3.14432078596	
-AGGTACTCTA	-31.740194784	-31.8300006361	-31.8776936587	
-AAACAGCAAT	-20.1393644627	-21.2759854851	-24.2832782797	
-TAGTTACAAC	-13.240934449	-14.2349552053	-14.7009524836	
-ACTGATTGAC	-10.2479819309	-10.2265866553	-10.3246609423	

Figura 31 – Exemplo mostrando o formato do arquivo de saída da validação cruzada para a GRE.

```

# Fold 1
*p1=5 p2=0.1 p3=0.01
+CCATTAATGG 14.89059 14.6355993126 14.5425080213
+CCGTTTTTAG 11.36754 11.1125493126 8.91963063318
. . .
. . .
+CAATAATTAG 11.38167 9.54140092894 8.17889688486
+CCAAATAAGG 8.9934 8.73840931265 6.54549063318
-CTCAGTTAGA -0.3101 -0.565090687352 -0.658181978653
-TATAGGTCGT -8.2275 -8.48249068735 -9.82066475459
. . .
. . .
-TAGTTACAAC -0.5984 -1.64602987921 -2.19156475459
-ACTGATTGAC -6.0165 -6.27149068735 -6.36458197865
*p1=5 p2=0.1 p3=0.02
+CCATTAATGG 14.51138 14.2563893126 14.1632980213
+CCGTTTTTAG 11.28262 11.0276293126 8.83471063318
. . .
. . .
-TAGTTACAAC 4.305 3.25737012079 2.71183524541
-ACTGATTGAC 1.3467 1.09170931265 0.998618021347
*p1=5 p2=1.0 p3=0.1
+CCATTAATGG 10.36672 10.1117293126 10.0186380213
+CCGTTTTTAG 9.0503 8.79530931265 6.60239063318
. . .
. . .
-TAGTTACAAC 4.3579 3.31027012079 2.76473524541
-ACTGATTGAC 1.6995 1.44450931265 1.35141802135
# Fold 2
*p1=5 p2=0.1 p3=0.01
+CCATAAATGG 12.22471 11.9697193126 11.3864089937
+CTAAAAATAG 14.19278 12.3525109289 9.84476941449
. . .
. . .
-TAGTTACAAC 3.8323 2.78467012079 2.23913524541
-ACTGATTGAC 3.0801 2.82510931265 2.73201802135
# Fold 5
*p1=5 p2=0.1 p3=0.01
+CCATTTATAG 14.49615 13.4485201208 12.9029852454
+CCATACATGG 11.3557 11.8933485045 11.0076180213
. . .
. . .
-TAGTTACAAC 2.2986 1.25097012079 0.705435245412
-ACTGATTGAC 3.6865 3.43150931265 3.33841802135
*p1=5 p2=1.0 p3=0.1
+CCATTTATAG 10.74245 9.69482012079 9.14928524541
+CCATACATGG 9.6948 10.2324485045 9.34671802135
. . .
. . .
-AAACAGCAAT 2.5702 1.52257012079 -1.77781058551
-TAGTTACAAC 2.5061 1.45847012079 0.912935245412
-ACTGATTGAC 3.7605 3.50550931265 3.41241802135

```

Apêndice E – Gráficos das medidas de desempenho

E.1 Gráficos das medidas de desempenho por tamanho do SLFT

Figura 32 – Gráfico da Sensibilidade (ou Taxa de Verdadeiros Positivos-TPR) (critério ROC01) por tamanho do SLFT, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 33 – Gráfico da Sensibilidade (ou Taxa de Verdadeiros Positivos-TPR) (critério Youden) por tamanho do SLFT, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 34 – Gráfico da Especificidade (1-FPR) (critério ROC01) por tamanho do SLFT, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 35 – Gráfico da Especificidade (1-FPR) (critério Youden) por tamanho do SLFT, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 36 – Gráfico da Precisão (critério ROC01) por tamanho do SLFT, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 37 – Gráfico da Precisão (critério Youden) por tamanho do SLFT, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 38 – Gráfico da Medida-F (critério ROC01) por tamanho do SLFT, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 39 – Gráfico da Medida-F (critério Youden) por tamanho do SLFT, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 40 – Gráfico do Erro Global (critério ROC01) por tamanho do SLFT, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 41 – Gráfico do Erro Global (critério Youden) por tamanho do SLFT, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

E.2 Gráficos das medidas de desempenho por tamanho da amostra

As faixas de tamanho de amostra, de 1 a 9, estão descritas na tabela da Figura 25.

Figura 42 – Gráfico da Sensibilidade (ou Taxa de Verdadeiros Positivos - TPR) (critério ROC01) por tamanho da amostra, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 43 – Gráfico da Sensibilidade (ou Taxa de Verdadeiros Positivos - TPR) (critério Youden) por tamanho da amostra, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 44 – Gráfico da Especificidade (1-FPR) (critério ROC01) por tamanho da amostra, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 45 – Gráfico da Especificidade (1-FPR) (critério Youden) por tamanho da amostra, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 46 – Gráfico da Precisão (critério ROC01) por tamanho da amostra, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 47 – Gráfico da Precisão (critério Youden) por tamanho da amostra, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 48 – Gráfico da Medida-F (critério ROC01) por tamanho da amostra, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 49 – Gráfico da Medida-F (critério Youden) por tamanho da amostra, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 50 – Gráfico do Erro Global (critério ROC01) por tamanho da amostra, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 51 – Gráfico do Erro Global (critério Youden) por tamanho da amostra, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

E.3 Gráficos das medidas de desempenho por faixas de Escore de Dependência

As faixas dos escores de dependência, de 1 a 7, estão descritas na tabela da Figura 22.

Figura 52 – Gráfico da Sensibilidade (ou Taxa de Verdadeiros Positivos - TPR) (critério ROC01) por faixas de Escore de Dependência, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 53 – Gráfico da Sensibilidade (ou Taxa de Verdadeiros Positivos - TPR) (critério Youden) por faixas de Escore de Dependência, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 54 – Gráfico da Especificidade (1-FPR) (critério ROC01) por tamanho da amostra, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 55 – Gráfico da Especificidade (1-FPR) (critério Youden) por faixas de Escore de Dependência, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 56 – Gráfico da Precisão (critério ROC01) por faixas de Escore de Dependência, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 57 – Gráfico da Precisão (critério Youden) por faixas de Escore de Dependência, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 58 – Gráfico da Medida-F (critério ROC01) por tamanho da amostra, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 59 – Gráfico da Medida-F (critério Youden) por faixas de Escore de Dependência, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 60 – Gráfico do Erro Global (critério ROC01) por faixas de Escore de Dependência, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Figura 61 – Gráfico do Erro Global (critério Youden) por faixas de Escore de Dependência, para os três modelos nulos, PWM e GRE.

Fonte: Antonio Ferrão Neto, 2017.

Apêndice F – Ferramenta de predição computacional baseada em gramáticas regulares estocásticas

Uma vez que as GREs já tenham sido inferidas, a criação de uma ferramenta de software baseada em GREs para identificação de SLFTs, é muito simples, conforme mostrado neste apêndice.

F.1 A ideia central da ferramenta de predição

Conforme representado na Figura 62, a ideia básica do algoritmo de predição é percorrer toda a sequência de bases do DNA (de tamanho L , na figura), fornecida em um arquivo FASTA (ou multi-FASTA), utilizando-se uma “janela deslizante” do tamanho do SLFT pretendido (de tamanho K na figura) e para cada posição da janela, calcular o escore da sequência, conforme descrito na seção 2.6.2. Se o escore calculado estiver acima de um limiar, também fornecido, o sítio definido pela janela deslizante será considerado um SLFT. Caso contrário, o sítio não será considerado um SLFT¹.

Figura 62 – Exemplo de utilização de uma janela deslizante de tamanho K para percorrer uma sequência de tamanho L . A cada passo, a janela se deslocará de uma posição e o sítio correspondente no DNA será testado, calculando-se o escore da sequência. Somente se esse escore estiver acima de um limiar, o sítio será considerado um SLFT.

Fonte: Antonio Ferrão Neto, 2017

Em termos de programação, a janela deslizante pode ser implementada com a utilização de um *loop*² sobre toda a extensão L da sequência de DNA fornecida, com passo unitário, ou seja, base a base, desde a primeira base da sequência até a base $L-K$. E, a

¹ Uma outra maneira de se descrever essa “janela deslizante”, em termos de subsequências, seria: a sequência de tamanho L é subdividida em $L-K+1$ subsequências de tamanho K , com início nas posições 1, 2, ..., $L-K$ da sequência de entrada. Cada uma dessas subsequências será analisada pela gramática.

² Por exemplo, com a utilização de comandos como `for` ou `while` das linguagens de programação.

partir de cada base da sequência percorrida por esse *loop*, são lidas³ K bases, compondo a janela deslizante, a ser testada em todas as suas posições.

F.2 GREs, PWMs e arquivos auxiliares

O desenvolvimento de um software de inferência gramatical não foi um dos objetivos deste trabalho, nem tampouco a comparação entre os diferentes softwares de inferência gramatical disponíveis. Portanto, um pré-requisito deste trabalho foi a utilização de um inferidor gramatical. Conforme discutido no item 5.3.2, todas as GREs deste trabalho foram inferidas utilizando-se o *framework* GrammarLab (MACHADO-LIMA, 2002; VIEIRA, 2004), que implementa o algoritmo Lapfa.

A saída do GrammarLab, para cada fator de transcrição, é um arquivo da GRE inferida, no formato “BFN”(“Backus Normal Form”), conforme exemplificado na Figura 66. Este arquivo contém as produções da GRE, como exemplificado na Figura 7.

E, juntamente com esses arquivos “BFN”, foram produzidos também arquivos “PWM”, contendo todas as informações relacionada à PWM de cada fator de transcrição. A Figura 67 mostra um exemplo deste arquivo.

Por uma decisão de projeto, ficou estabelecido que os nomes dos arquivos “BNF” não seriam compostos pelos nomes dos arquivos *Jaspar* 2016 pois, eventualmente, outros bancos de dados poderão vir a ser utilizados no futuro, com padrões de nomes distintos, o que poderia invalidar a nomenclatura adotada. Foi criada então uma numeração sequencial interna para os nomes dos arquivos e o relacionamento entre essa numeração sequencial interna e o nome externo do arquivo, no *Jaspar* 2016, é definida em uma tabela fornecida pelo arquivo auxiliar `tf_parameters.txt`, que é uma das entradas deste programa de predição. A Figura 64 mostra alguns trechos do arquivo `tf_parameters.txt`.

Além do arquivo `tf_parameters.txt`, também foi produzido um outro arquivo auxiliar, chamado `gre_tool_parameters.txt`, durante as validações cruzadas para testar o desempenho das GREs. Este arquivo contém todos os escores de corte (coluna com título “Cutoff: 0_1 mean” no arquivo) e todos os parâmetros Lapfa que produziram o maior AUC durante as validações cruzadas, além das medidas de desempenho correspon-

³ Várias linguagens permitem extrair uma *substring* de uma *string* com a utilização de uma função (`strncpy` da linguagem C ou C++, `substr` da linguagem C++, `substr` da linguagem Perl, etc.). Na linguagem Python, existe o conceito de *slicing* (“fatiamento”), como na linha 311 da listagem F.1.

dentess a esses parâmetros Lapfa “ótimos”. A Figura 65 mostra alguns trechos do arquivo `gre_tool_parameters.txt`.

Outro arquivo utilizado pelo programa de predição é o `lista_gen_freq.txt`, que contém as frequências genômicas para todas as espécies contidas no banco de dados empregado (Jaspar 2016). Este arquivo reproduz a Tabela 3 e um exemplo está apresentado na Figura 63.

Figura 63 – Arquivo de entrada `lista_gen_freq.txt`, que contém as frequências genômicas das espécies utilizadas pelo banco de dados Jaspqr 2016.

```
"#", "GCAT"
"Bos taurus", 0.2094, 0.2094, 0.2906, 0.2906
"Arabidopsis thaliana", 0.1830, 0.1830, 0.3170, 0.3170
"Caenorhabditis elegans", 0.1773, 0.1773, 0.3227, 0.3227
"Drosophila melanogaster", 0.2107, 0.2107, 0.2893, 0.2893
"Gallus gallus", 0.2146, 0.2146, 0.2854, 0.2854
"Homo sapiens", 0.2053, 0.2053, 0.2947, 0.2947
"Hordeum vulgare", 0.2215, 0.2215, 0.2785, 0.2785
"Mus musculus", 0.2131, 0.2131, 0.2869, 0.2869
"Nicotiana sp.", 0.1955, 0.1955, 0.3045, 0.3045
"Oryctolagus cuniculus", 0.2203, 0.2203, 0.2797, 0.2797
"Rattus norvegicus", 0.2118, 0.2118, 0.2882, 0.2882
"Rattus rattus", 0.2118, 0.2118, 0.2882, 0.2882
"Triticum aestivum", 0.2175, 0.2175, 0.2825, 0.2825
"Xenopus (Silurana) tropicalis", 0.2027, 0.2027, 0.2973, 0.2973
"Xenopus laevis", 0.2012, 0.2012, 0.2988, 0.2988
"Zea mays", 0.2325, 0.2325, 0.2675, 0.2675
"Oncorhynchus mykiss", 0.2500, 0.2500, 0.2500, 0.2500
"Antirrhinum majus", 0.2500, 0.2500, 0.2500, 0.2500
"Halocynthia roretzi", 0.2500, 0.2500, 0.2500, 0.2500
"Petunia x hybrida", 0.2500, 0.2500, 0.2500, 0.2500
"Pisum sativum", 0.2500, 0.2500, 0.2500, 0.2500
```

Fonte: Antonio Ferrão Neto, 2017

Todos esses arquivos auxiliares, assim como os arquivos “BNF” e “PWM”, são pré-requisitos para a execução do programa de predição. Eles fazem parte de um conjunto de arquivos que somente sofre alterações quando se deseja que os fatores de transcrição de um novo banco de dados sejam utilizados na predição. Neste caso, as novas GREs e PWMs precisam ser inferidas, seus arquivos “BNF” e “PWM” gerados e os arquivos `tf_parameters.txt` e `gre_tool_parameters.txt` devem ser editados para receberem as novas informações.

Figura 64 – Trechos do arquivo de entrada `tf_parameters.txt`, que permite relacionar, para os 341 fatores de transcrição utilizados neste trabalho, os nomes sequenciais internos do programa de predição (`FTxxxxxxx`) com os nomes externos, provenientes do Jaspar 2016 (`MAyyyy.z.sites`). Por exemplo, na linha destacada em vermelho, o nome interno `FT0000162` relaciona-se com o nome externo do FT no Jaspar 2016: `MA0217.1.sites`.

```
* FT0000001 AGL3 Arabidopsis thaliana SELEX MA0001.1.sites 10 97
* FT0000002 RUNX1 Homo sapiens SELEX MA0002.1.sites 11 26
* FT0000003 AG Arabidopsis thaliana SELEX MA0005.1.sites 11 90
* FT0000004 Ar Rattus rattus SELEX MA0007.1.sites 22 24
* FT0000005 HAT5 Arabidopsis thaliana SELEX MA0008.1.sites 8 25
* FT0000006 T Mus musculus SELEX MA0009.1.sites 11 40

...

* FT0000162 caup Drosophila melanogaster bacterial 1-hybrid MA0217.1.sites 5 19

...

* FT0000335 PIF3 Arabidopsis thaliana ChIP-seq MA0560.1.sites 10 527
* FT0000336 PIF4 Arabidopsis thaliana ChIP-seq MA0561.1.sites 8 335
* FT0000337 PIF5 Arabidopsis thaliana ChIP-seq MA0562.1.sites 8 286
* FT0000338 SEP3 Arabidopsis thaliana ChIP-seq MA0563.1.sites 11 150
* FT0000339 AGL27 Arabidopsis thaliana ChIP-seq MA1012.1.sites 14 142
* FT0000340 AP1 Arabidopsis thaliana ChIP-seq MA0940.1.sites 13 1622
* FT0000341 CEBPB Homo sapiens ChIP-seq MA0466.1.sites 11 99494
```

Fonte: Antonio Ferrão Neto, 2017

Figura 66 – Exemplo de GRE inferida pelo *framework* GrammarLab. O nome deste arquivo, FT0000162_N1_LAPFA_5_0.1_0.09.bnf, foi construído à partir do nome interno (Figura 64), seguido do modelo nulo (N1, N2 ou N3) e os valores dos parâmetros Lapfa que resultaram na maior AUC da curva ROC, no treinamento desta GRE. A palavra “LAPFA” no nome deste arquivo, indica que essa gramática foi inferida com o algoritmo Lapfa. Relacionando-se o nome interno “FT0000162” com o arquivo `tf_parameters.txt`, pode-se determinar que esta GRE é referente ao fator de transcrição *caup* de *Drosophila melanogaster*, treinada a partir dos dados do arquivo Jaspar 2016: MA0217.1.sites.

```

4
<1> ::= A<2> [0.205789473684211] #0#
<1> ::= T<5> [0.437368421052632] #0#
<1> ::= C<2> [0.118947368421053] #0#
<1> ::= G<2> [0.147894736842105] #0#
<1> ::= # [0.09] #0#
<2> ::= A<6> [0.561428571428571] #0#
<2> ::= T<6> [0.168571428571429] #0#
<2> ::= C<6> [0.09] #0#
<2> ::= G<6> [0.09] #0#
<2> ::= # [0.09] #0#
<5> ::= A<10> [0.456666666666667] #0#
<5> ::= G<6> [0.181666666666667] #0#
<5> ::= T<6> [0.181666666666667] #0#
<5> ::= C<6> [0.09] #0#
<5> ::= # [0.09] #0#
<6> ::= A<13> [0.64] #0#
<6> ::= C<13> [0.09] #0#
<6> ::= G<13> [0.09] #0#
<6> ::= T<13> [0.09] #0#
<6> ::= # [0.09] #0#
<10> ::= A<17> [0.64] #0#
<10> ::= C<13> [0.09] #0#
<10> ::= G<13> [0.09] #0#
<10> ::= T<13> [0.09] #0#
<10> ::= # [0.09] #0#
<13> ::= C<20> [0.64] #0#
<13> ::= A<20> [0.09] #0#
<13> ::= G<20> [0.09] #0#
<13> ::= T<20> [0.09] #0#
<13> ::= # [0.09] #0#
<17> ::= C<24> [0.64] #0#
<17> ::= A<20> [0.09] #0#
<17> ::= G<20> [0.09] #0#
<17> ::= T<20> [0.09] #0#
<17> ::= # [0.09] #0#
<20> ::= A<27> [0.64] #0#
<20> ::= C<27> [0.09] #0#
<20> ::= G<27> [0.09] #0#
<20> ::= T<27> [0.09] #0#
<20> ::= # [0.09] #0#
<24> ::= A<31> [0.64] #0#
<24> ::= C<27> [0.09] #0#
<24> ::= G<27> [0.09] #0#
<24> ::= T<27> [0.09] #0#
<24> ::= # [0.09] #0#
<27> ::= # [1] #0#
<31> ::= # [1] #0#

```

Fonte: Antonio Ferrão Neto, 2017

Figura 67 – Exemplo de PWM inferida durante a validação cruzada. O nome deste arquivo, FT0000162_N1.pwm, foi construído à partir do nome interno (Figura 64), seguido do modelo nulo (N1, N2 ou N3). Relacionando-se o nome interno “FT0000162” com o arquivo `tf_parameters.txt`, pode-se determinar que esta PWM é referente ao fator de transcrição *caup* de *Drosophila melanogaster*, treinada a partir dos dados do arquivo Jasper 2016: MA0217.1.sites.

```
4
# AUC mean= 0.991304347826
# AUC stdev= 0.00606275381546
# Cutoff: 0_1 mean= 3.3276241204
# Cutoff: 0_1 stdev= 0.821686738228
# Cutoff: Youden mean= 3.3276241204
# Cutoff: Youden stdev= 0.821686738228
# A C G T
* -0.234465253637 -2.0 -1.15200309345 1.29278174923
* 1.51096191928 -4.32192809489 -1.15200309345 -0.621488376746
* 1.94485844581 -4.32192809489 -4.32192809489 -4.32192809489
* -4.32192809489 1.94485844581 -4.32192809489 -4.32192809489
* 1.94485844581 -4.32192809489 -4.32192809489 -4.32192809489
```

Fonte: Antonio Ferrão Neto, 2017

Figura 68 – Exemplo de arquivo de parâmetros ("`exe_parameters.txt`").

```
#####
#Parâmetros de execução:
#####
CROMOSSOMES=X,Y,2L,2R,3L,3R,4
FT_LIST=7,8,9,11,12,18,21,37,66,95,121,122,123,124,125,126,127,128,129,130,131,132,133,134
MODELS=N1
CRITERIA=0_1
DEBUG=NO
PSEUDOCOUNTER=0.01
#####
#Diretórios e arquivos:
#####
#Arquivo FASTA contendo as sequências a serem percorridas. Por exemplo, um genoma:
INPUT_DATA=/home/aferrao/genomas/drosophila_melanogaster/Drosophila_melanogaster.BDGP6.dna_sm.toplevel.fa
#Arquivo texto contendo os sítios encontrados, candidatos a serem SLFT:
OUTPUT_DATA=/home/aferrao/tfbs_search_tool/output/found_tfbs.txt
#Arquivo contendo dados dos Fatores de Transcrição:
TF_PARAM=/home/aferrao/roc_process/new_output_after_roc/tf_parameters.txt
#Arquivo contendo os parâmetros e os cutoffs das gramáticas:
GRE_TOOL_PARAM=/home/aferrao/roc_process/new_output_after_roc/gre_tool_parameters.txt
#Arquivo contendo as frequências genômicas das espécies:
GENOME_FREQ=/home/aferrao/swalign/sites.cut.fasta/lista_gen_freq.txt
#####
#Diretórios:
#####
#Diretório contendo todas as gramáticas e as PWMs previamente treinadas:
PWM_GRE_DIRECTORY_IN=/home/aferrao/roc_process/new_output_after_roc/
#Diretório de compilação do autômato (em produção será utilizado apenas com DEBUG=YES):
AUTOMATON_DIR=/home/aferrao/lgc/src/util/automaton/
#####
```

Fonte: Antonio Ferrão Neto, 2017

F.3 Parâmetros do programa de predição de SLFTs

Devido à grande quantidade de valores que seriam necessários passar por linha de comando no sistema operacional, optou-se pela utilização de um arquivo de parâmetros, chamado `exe_parameters.txt`.

Os parâmetros são identificados pelo seu nome, seguido de um símbolo de igual (“=”) e o valor do parâmetro. Havendo mais de um valor, eles devem ser separados por vírgulas:

`<NOME_PARÂMETRO>=<VALOR1>[, <VALOR2>, <VALOR3>, ...]`

Os parâmetros necessários para a execução do programa são os seguintes:

1. **"#"**: este caractere, iniciando a linha, indica que a linha toda é um comentário e deve ser ignorada.
2. **"FT_LIST"**: lista de fatores de transcrição a serem preditos no arquivo FASTA de entrada, indicado pelo parâmetro "INPUT_DATA". A numeração dos FTs está definida no arquivo `tf_parameters.txt`. Por exemplo, FT0000001 corresponde ao FT 1, que corresponde ao FT **AGL3** da espécie *Arabidopsis thaliana*, cujos dados foram obtidos do arquivo `MA0001.1.sites` do Jaspar 2016.
3. **"CROMOSSOMES"**: lista dos cromossomos a serem processados. A indicação do cromossomo deve ser a mesma que aparece nos cabeçalhos de linha do arquivo FASTA de entrada (as linhas que começam com o caractere “>”).
4. **"MODELS"**: os modelos nulos a serem utilizados (N1, N2 ou N3).
5. **"CRITERIA"**: os critérios para a escolha do limiar de corte da curva ROC (0_1 ou Youden).
6. **"DEBUG"**: *flag* indicando se o programa será executado em modo de *debug* ou não. Se o valor for diferente de “NO”, ambos os autômatos serão executados: o que foi implementado pelo GrammarLab e o que foi implementado no próprio programa de predição. Neste caso, o programa ficará muito mais lento.
7. **"PSEUDOCOUNTER"**: valor do pseudocontador utilizado no cálculo da frequência de fundo no modelo N3, que utiliza a frequência de bases do próprio sítio. Quando uma base não estiver presente no sítio em questão, o valor de probabilidade deste pseudocontador é assumido.

8. **"INPUT_DATA"**: diretório e nome do arquivo FASTA de entrada, contendo as sequências onde os sítios serão pesquisados/preditos.
9. **"OUTPUT_DATA"**: diretório e nome do arquivo de saída analítico. Cada sítio que for predito gera uma linha neste arquivo, tanto para a PWM, quanto para a GRE.
10. **"TF_PARAM"**: diretório e nome do arquivo de entrada utilizado, principalmente, para relacionar os nomes internos dos arquivos com os nomes externos, por exemplo, seguindo o padrão de nomes do banco *Jaspar* 2016.
11. **"GRE_TOOL_PARAM"**: diretório e nome do arquivo de entrada contendo dados das GREs inferidas pelo *GrammarLab* durante as validações cruzadas.
12. **"GENOME_FREQ"**: diretório e nome do arquivo que contém as frequências genômicas para todas as espécies em estudo.
13. **"PWM_GRE_DIRECTORY_IN"**: diretório onde residem os arquivos das GREs (arquivos “BNF”) e das PWMs (arquivos “PWM”).
14. **"AUTOMATON_DIR"**: diretório o autômato do *GrammarLab* será compilado, quando o *flag* “DEBUG” for diferente de “NO”.

A Figura 68 mostra um exemplo do arquivo `exe_parameters.txt`.

F.4 Detalhes sobre o programa de predição de SLFTs

O programa de predição, da listagem F.1⁴, pode ser utilizado exatamente como ele está, juntamente com os arquivos de entrada `tf_parameters.txt`, `gre_tool_parameters.txt`, as GREs e PWMs inferidas e editando-se o arquivo `exe_parameters.txt`. Mas também é possível customizar o programa a fim de adequá-lo a alguma finalidade específica. E para facilitar o entendimento do código, visando-se uma customização ou adaptação, esta seção detalha o programa de predição apresentado na listagem F.1.

A biblioteca *BioPython*⁵ é um pré-requisito para este programa. Ela é utilizada apenas para fazer a leitura (*parse*) do arquivo FASTA de entrada, contendo as sequências onde os sítios serão procurados⁶. Na linha 2, a interface SeqIO do BioPython é importada.

⁴ O programa de predição `tfbs_search.py`, assim como os demais arquivos de entrada, estão publicamente disponíveis e podem ser obtidos em https://github.com/tonyfer2017/public_masters_source - acesso em 17/12/2017. Versões mais atualizadas poderão eventualmente ser obtidas no mesmo link.

⁵ Disponível em <http://biopython.org/wiki/SeqIO> - acessado em 19/12/2017.

⁶ Essa leitura de arquivos FASTA poderia ser feita simplesmente com a função `open()` do Python. Entretanto, por simplicidade e por ser o *BioPython* muito utilizado na Bioinformática, este pacote foi adotado para o *parse* de arquivos FASTA.

O programa de predição inicia a sua execução na linha 346, recebendo como argumento o nome do arquivo de parâmetros `exe_parameters.txt`, chamando a função `proc_fasta()` na linha 275. Na linha 278, esse arquivo de parâmetros é carregado em um dicionário por meio da função `parse_exe_par(exe_par)` definido entre as linhas de 183 a 200.

As variáveis globais `global_compila_main`, `global_gre_pwm_source` e `global_pseudocontador_seq` (linhas 279, 280 e 281) são carregadas a partir de dados do arquivo de parâmetros `exe_parameters.txt`. São variáveis globais para que esses valores possam ser utilizados em quaisquer trechos do programa. A variável global `global_gre_error` (linhas 276 e 277), acumula em um "set"⁷, na linha 226, os identificadores dos FTs cujas respectivas gramáticas, durante o cálculo do escore, produziram algum erro. No final da execução do programa, na linha 343, todo o conteúdo do "set" `global_gre_error` é enviado para a tela.

A seguir, nas linhas 282 e 283, todas as PWMs e todas as GREs dos fatores de transcrição, que foram solicitados como parâmetro no arquivo `exe_parameters.txt` (parâmetro "FT_LIST"), são lidas dos arquivos "PWM" e "BNF" e carregadas, respectivamente, para os dicionários Python `dict_pwm_parsed` e `dict_gre_parsed` (funções `parse_pwm`, linhas 58 a 86 e `parse_gre`, linhas 89 a 126). A ideia é manter todas as PWMs e GREs solicitadas em memória⁸.

Depois esses dicionários são utilizados em cada posição da "janela deslizante" (linhas 307 a 311)⁹, para se calcular o escore do sítio definido pela janela sobre o DNA.

Os comandos "for" das linhas 314 e 315 forçam o programa a repetir os cálculos do escore para cada modelo nulo (N1, N2 e N3) e critério de corte da curva ROC (ROC01 ou Youden), que foram requisitados no arquivo `exe_parameters.txt`.

⁷ "set" é uma estrutura de dados que não permite repetições.

⁸ Essa técnica diminui bastante o tempo de processamento do programa, já que as PWMs e as GREs serão utilizadas para testar cada posição da "janela deslizante". E o número desses testes poderá ser de milhares, ou milhões de vezes, para uma região do DNA (*locus* gênico), ou mesmo todo o genoma de uma espécie, para cada fator de transcrição.

⁹ O "for" que se inicia na linha 307, percorre todas as posições da sequência FASTA presente na variável "sequence", que teve seu valor atribuído na linha 301. Essas posições serão percorridas desde a primeira (posição zero) até o fim da sequência, descontando-se o tamanho da janela, ou do sítio (na linha 312, o "if" impede que a janela deslizante ultrapasse o final da sequência). Já o "for" da linha 309, irá percorrer todos os fatores de transcrição que foram solicitados pelo usuário (por meio do arquivo `exe_parameters.txt`), calculando a janela deslizante para cada fator de transcrição na linha 311, com o tamanho da janela definido na linha 310.

Nas linhas 316 e 317, os valores dos limiares¹⁰, tanto para a GRE, quanto para a PWM, para esse fator, modelo e critério, são armazenados nas variáveis `gre_cutoff` e `pwm_cutoff`, respectivamente.

Nas linhas 318 e 319, são calculados os escores para a PWM e a GRE. O cálculo do escore da PWM é efetuado pela função `calc_escore_pwm`, definido entre as linhas 203 e 211, que recebe como parâmetro a sequência, que foi definida pela janela deslizante e a PWM do FT atual e que já está disponível em memória no dicionário Python `dict_pwm_parsed`, carregado de arquivo na linha 282. O cálculo do escore da GRE é efetuado pela função `calc_escore_gre`, definido entre as linhas 214 e 228, que recebe como parâmetro o número do FT, a sequência que foi definida pela janela deslizante e a gramática, que já está disponível em memória no dicionário Python `dict_gre_parsed`, carregado de arquivo na linha 283. Esta função cumpre exatamente o mesmo papel do autômato do **GrammarLab**, calculando o escore da sequência para a GRE do FT em questão. O número do FT somente é recebido como parâmetro nesta função, para acumular alguma condição de erro na variável "set" `global_gre_error`, na linha 226.

A fim de se verificar se o cálculo do escore da gramática, calculado pela função `calc_escore_gre`, produz o mesmo resultado que o cálculo realizado pelo autômato do **GrammarLab**, foi introduzido o bloco "if", entre as linha 320 e 325. Esse bloco "if" só funciona quanto a variável `console_print` for verdadeira. E ela só é verdadeira se o conteúdo da variável "DEBUG" for diferente de "NO", conforme definido nas linhas 287 a 290. E, sendo ela verdadeira, o programa calcula também um outro escore (`score_lapfa`), chamando a função `calc_score_lapfa`, definida entre as linhas 162 e 180, que por sua vez faz a chamada ao autômato do **GrammarLab** via sistema operacional. Na linha 323, ambos os escores são impressos na tela, permitindo a conferência dos valores calculados. Essa opção de debug só deve ser ativada para testes, uma vez que a diferença na velocidade de processamento é muito grande.

Entre as linhas 326 e 332, os escores das GREs são corrigidos para cada um dos modelos nulos. As PWMs já consideram em seus cálculos as correções do modelo nulo.

O programa emite um relatório que reporta todas as ocorrências de sítios preditos, tanto para PWMs, quanto para GREs. Sempre que um sítio for predito pela PWM, na

¹⁰ O limiar (ou *cutoff*) irá definir o limite para que um sítio, definido pela janela deslizante, possa ser considerado um SLFT ou não.

linha 333, ou pela GRE, na linha 336, a função `out_report1` será chamada (linhas 334 e 337), para reportar o sítio predito.

Figura 69 – Exemplo de trecho de arquivo de saída ("found_tfbs.txt").

METHOD	TF_CODE	CHROMOSSOME	GLOBAL_POSITION	LOCAL_POSITION	SLIDING_WINDOW	MODEL	SCORE_PWM/GRE	CUTOFF_PWM/GRE	CRITERION	TF_ACRONYM	SPECIE_NAME
PWM	FT0000155	2L	0	cgacaa N1	0.0	-0.46747501617	0_1	achi	Drosophila melanogaster		
PWM	FT0000180	2L	0	cgacaatgc	N1	0.0	-0.687101903346	0_1	prd	Drosophila melanogaster	
PWM	FT0000191	2L	0	cgacaa N1	0.0	-1.10337855693	0_1	vis	Drosophila melanogaster		
PWM	FT0000206	2L	0	cgacaatgca	N1	0.0	-1.68319287661	0_1	h	Drosophila melanogaster	
PWM	FT0000155	2L	1	gacaat N1	0.0	-0.46747501617	0_1	achi	Drosophila melanogaster		
PWM	FT0000180	2L	1	gacaatgca	N1	0.0	-0.687101903346	0_1	prd	Drosophila melanogaster	
PWM	FT0000191	2L	1	gacaat N1	0.0	-1.10337855693	0_1	vis	Drosophila melanogaster		
PWM	FT0000206	2L	1	gacaatgca	N1	0.0	-1.68319287661	0_1	h	Drosophila melanogaster	
PWM	FT0000155	2L	2	acaatg N1	0.0	-0.46747501617	0_1	achi	Drosophila melanogaster		
...											
GRE	FT0000155	2L	5203	tGATAA N1	3.73056822835	3.725594	0_1	achi	Drosophila melanogaster		
GRE	FT0000128	2L	5205	ATAAGAT N1	5.91685937354	5.621508	0_1	CG11617	Drosophila melanogaster		
GRE	FT0000146	2L	5205	ATAAGAT N1	4.99932835761	4.821058	0_1	schlank	Drosophila melanogaster		
GRE	FT0000124	2L	5206	TAAGATG N1	5.75587534524	4.805768	0_1	B-H1	Drosophila melanogaster		
GRE	FT0000155	2L	5206	TAAGAT N1	5.49303162423	3.725594	0_1	achi	Drosophila melanogaster		
GRE	FT0000157	2L	5206	TAAGA N1	3.52555312303	3.13181	0_1	ara	Drosophila melanogaster		
GRE	FT0000191	2L	5206	TAAGAT N1	3.46719289636	3.257594	0_1	vis	Drosophila melanogaster		
GRE	FT0000151	2L	5210	ATGATTC N1	5.42128071057	5.375936	0_1	Rx	Drosophila melanogaster		
...											
PWM	FT0000155	Y	137547943	3667335 tctctt N1	0.0	-0.46747501617	0_1	achi	Drosophila melanogaster		
PWM	FT0000180	Y	137547943	3667335 tctctctc	N1	0.0	-0.687101903346	0_1	prd	Drosophila melanogaster	
PWM	FT0000191	Y	137547943	3667335 tctctt N1	0.0	-1.10337855693	0_1	vis	Drosophila melanogaster		
PWM	FT0000206	Y	137547943	3667335 tctctctct	N1	0.0	-1.68319287661	0_1	h	Drosophila melanogaster	
PWM	FT0000155	Y	137547944	3667336 ctcttc N1	0.0	-0.46747501617	0_1	achi	Drosophila melanogaster		
PWM	FT0000180	Y	137547944	3667336 ctctctct	N1	0.0	-0.687101903346	0_1	prd	Drosophila melanogaster	
PWM	FT0000191	Y	137547944	3667336 ctcttc N1	0.0	-1.10337855693	0_1	vis	Drosophila melanogaster		
PWM	FT0000206	Y	137547944	3667336 ctctctctt	N1	0.0	-1.68319287661	0_1	h	Drosophila melanogaster	

Fonte: Antonio Ferrão Neto, 2017

Código F.1 – Programa de predição utilizando gramáticas regulares estocásticas previamente inferidas.

```

1  # coding=utf-8
2  from Bio import SeqIO
3  from math import log, exp
4  import math
5  import os
6  import subprocess
7  global_compila_main = ""
8  global_gre_pwm_source = ""
9  global_pseudocontador_seq = 0.01
10 global_gre_error = set()
11
12
13 def parse_freq_gen(genfreqfile):
14     genfreq = open(genfreqfile, 'r')
15     dict_freq_gen = {}
16     for freq in genfreq:
17         if freq[1:2] == "#":
18             continue
19         pos_virg1 = freq.find(",")
20         pos_virg2 = freq.find(",", pos_virg1 + 1)
21         pos_virg3 = freq.find(",", pos_virg2 + 1)
22         pos_virg4 = freq.find(",", pos_virg3 + 1)
23         freq_g = float(freq[pos_virg1 + 1:pos_virg2])
24         freq_c = float(freq[pos_virg2 + 1:pos_virg3])
25         freq_a = float(freq[pos_virg3 + 1:pos_virg4])
26         freq_t = float(freq[pos_virg4 + 1:-1])
27         dict_freq_bas = {}
28         dict_freq_bas['A'] = freq_a
29         dict_freq_bas['C'] = freq_c

```

```

30 dict_freq_bas['G'] = freq_g
31 dict_freq_bas['T'] = freq_t
32 dict_freq_gen[freq[1:pos_virg1 - 1]] = dict_freq_bas
33 genfreq.close()
34 return dict_freq_gen
35
36
37 def parse_tf_par(list_tf_par, tf_parameters):
38     dict_tf_par = {}
39     list_tf_par_f = ["FT" + str(ft).zfill(7) for ft in list_tf_par]
40     with open(tf_parameters, 'r') as h_tf_par:
41         for tf_par_line in h_tf_par:
42             data = tf_par_line.split("\t")
43             if (data[1].rstrip() == ""):
44                 continue
45             if (data[1] not in list_tf_par_f):
46                 continue
47             dict_tf_par[data[1]] = {}
48             dict_tf_par[data[1]]["TF_CODE"] = data[2].rstrip()
49             dict_tf_par[data[1]]["TF_NAME"] = data[3].rstrip()
50             dict_tf_par[data[1]]["TF_TECH"] = data[4].rstrip()
51             dict_tf_par[data[1]]["TF_ORIG_NAME"] = data[5].rstrip()
52             dict_tf_par[data[1]]["TF_TFBS_LEN"] = int(data[6].rstrip())
53             dict_tf_par[data[1]]["TF_TFBS_LEN"] = int(data[6].rstrip())
54             dict_tf_par[data[1]]["TF_SAMPLE_SIZE"] = int(data[7].rstrip())
55             return dict_tf_par
56
57
58 def parse_pwm(list_exe_par, list_models):
59     os.chdir(global_gre_pwm_source)

```

```

60 list_models = [x for x in list_models if x != "N3"]
61 dict_pwm= {}
62 for tf_name in list_exe_par:
63     dict_pwm[tf_name] = {}
64     for model in list_models:
65         dict_pwm[tf_name][model] = {}
66         dict_pwm[tf_name][model]["PWM"] = {}
67         pwm_filename = "FT" + str(tf_name).zfill(7) + "_" + model + ".pwm"
68         with open(pwm_filename, 'r') as pwm_file:
69             position = 0
70             for pwm_line in pwm_file:
71                 data = pwm_line.split("\t")
72                 if (data[0] == "#"):
73                     if (data[1] != "A"):
74                         dict_pwm[tf_name][model][data[1][:-1]] = float(data[2].rstrip())
75                     else:
76                         dict_pwm[tf_name][model]["PWM"]["A"] = {}
77                         dict_pwm[tf_name][model]["PWM"]["C"] = {}
78                         dict_pwm[tf_name][model]["PWM"]["G"] = {}
79                         dict_pwm[tf_name][model]["PWM"]["T"] = {}
80                     elif (data[0] == "*"):
81                         dict_pwm[tf_name][model]["PWM"]["A"][position] = float(data[1].rstrip())
82                         dict_pwm[tf_name][model]["PWM"]["C"][position] = float(data[2].rstrip())
83                         dict_pwm[tf_name][model]["PWM"]["G"][position] = float(data[3].rstrip())
84                         dict_pwm[tf_name][model]["PWM"]["T"][position] = float(data[4].rstrip())
85                         position += 1
86         return dict_pwm
87
88
89 def parse_gre(list_exe_par, gre_parameters):

```



```

90 dict_gre= {}
91 for ft in list_exe_par:
92     tf_name = "FT" + str(ft).zfill(7)
93     ft_found = False
94     with open(gre_parameters, 'r') as gre_par:
95         for gre_line in gre_par:
96             if (gre_line.find(tf_name) < 0) and not ft_found:
97                 continue
98             ft_found = True
99             data = gre_line.split("\t")
100             if (data[0] == "@"):
101                 dict_gre[ft] = {}
102                 dict_gre[ft]["TF_NAME"] = data[1].rstrip()
103                 dict_gre[ft]["SOURCE_NAME"] = data[2].rstrip()
104                 dict_gre[ft]["TFBS_LENGTH"] = int(data[3].rstrip())
105                 dict_gre[ft]["SAMPLE_SIZE"] = int(data[4].rstrip())
106                 dict_gre[ft]["FT_CODE"] = data[5].rstrip()
107                 dict_gre[ft]["SPECIE_NAME"] = data[6].rstrip()
108                 continue
109             elif (data[0] == "#"):
110                 continue
111             elif (data[0] == "*"):
112                 dict_gre[ft][data[1]] = {}
113                 dict_gre[ft][data[1]]["P1"] = float(data[2].rstrip())
114                 dict_gre[ft][data[1]]["P2"] = float(data[3].rstrip())
115                 dict_gre[ft][data[1]]["P3"] = float(data[4].rstrip())
116                 dict_gre[ft][data[1]]["AUC_mean"] = data[5].rstrip()
117                 dict_gre[ft][data[1]]["Cutoff_0_1"] = float(data[6].rstrip())
118                 dict_gre[ft][data[1]]["Cutoff_Youden"] = float(data[7].rstrip())
119                 gre_file_name = tf_name + "_" + data[1].rstrip() + "_LAPPA_" + str(int(float(data[2].rstrip()))) + \

```

```

120         "_" + str(data[3].rstrip()) + "_" + str(data[4].rstrip()) + ".bnf"
121
122     dict_gre[ft][data[1]][ "GRE_FILENAME" ] = gre_file_name
123     dict_gre[ft][data[1]][ "GRE" ] = {}
124     dict_gre[ft][data[1]][ "GRE" ] = parse_bnf(gre_file_name)
125     if (data[1] == "N3"):
126         break
127
128     return dict_gre
129
130 def calc_logP_N1(seq):
131     return len(seq) * math.log(0.25, 2)
132
133 def calc_logP_N2(seq, freq):
134     result = 0.0
135     for base in seq:
136         if base.upper() == "A":
137             result += math.log(freq['A'], 2)
138         elif base.upper() == "C":
139             result += math.log(freq["C"], 2)
140         elif base.upper() == "G":
141             result += math.log(freq["G"], 2)
142         elif base.upper() == "T":
143             result += math.log(freq["T"], 2)
144     return result
145
146
147 def calc_logP_N3(seq):
148     psc = global_pseudocontador_seq # 0.01
149     result = 0.0

```

```

150 for base in seq:
151     if base == "A":
152         result += math.log((seq.count('A') + psc) / len(seq), 2)
153     elif base == "C":
154         result += math.log((seq.count('C') + psc) / len(seq), 2)
155     elif base == "G":
156         result += math.log((seq.count('G') + psc) / len(seq), 2)
157     elif base == "T":
158         result += math.log((seq.count('T') + psc) / len(seq), 2)
159     return result
160
161
162 def calc_score_lapfa(sliding_window, gre_file):
163     os.chdir(global_compila_main)
164     p = subprocess.Popen(global_compila_main + "/" + main + gre_file,
165                          shell=True, stdout=subprocess.PIPE, stderr=subprocess.STDOUT)
166     for line in p.stdout.readlines():
167         print line,
168     retval = p.wait()
169     p = subprocess.Popen("g++Main2.cpp-oMain2", shell=True, stdout=subprocess.PIPE, stderr=subprocess.STDOUT)
170     for line in p.stdout.readlines():
171         print line,
172     retval = p.wait()
173     result = 0.0
174     p = subprocess.Popen(global_compila_main + "/" + main2 + sliding_window, shell=True,
175                          stdout=subprocess.PIPE, stderr=subprocess.STDOUT)
176     for line in p.stdout.readlines():
177         print line
178         result = float(line)
179     retval = p.wait()

```

```

180     return result
181
182
183     def parse_exe_par(par):
184         dict_exe = {}
185         with open(par, 'r') as exe_par_hdl:
186             for line in exe_par_hdl:
187                 line = line.rstrip()
188                 if (line[0:1] == "#"):
189                     continue
190                 sep_pos = line.find("#")
191                 if (sep_pos == -1):
192                     continue
193                 left_line = line[0:sep_pos]
194                 right_line = line[sep_pos + 1:]
195                 words = right_line.split(",")
196                 try:
197                     dict_exe[left_line] = [int(i) for i in words]
198                 except ValueError:
199                     dict_exe[left_line] = words
200             return dict_exe
201
202
203     def calc_score_pwm(sequence, pwm):
204         score = 0.0
205         i = 0
206         for base in sequence:
207             if base not in ('A', 'C', 'G', 'T'):
208                 continue
209             score = score + pwm[base][i]

```

```
210         i += 1
211     return score
212
213
214 def calc_score_gre(tf, sequence, gre):
215     knot = 1
216     score = 0.0
217     for i, base in enumerate(sequence):
218         if base not in ["A", "C", "G", "T"]:
219             real_base = "#"
220         else:
221             real_base = base
222     try:
223         score += gre[knot][real_base][1]
224         knot = gre[knot][real_base][0]
225     except:
226         global_gre_error.add(tf)
227     score += gre[knot][ "#" ][1]
228     return score
229
230
231 def get_number(str_number):
232     first_pos = str_number.find("<")
233     last_pos = str_number.find(">")
234     num_str = str_number[first_pos+1:last_pos]
235     ret_number = int(num_str)
236     return ret_number
237
238
239 def get_probability(str_number):
```

```

240 first_pos = str_number.find("[")
241 last_pos = str_number.find("]")
242 num_str = str_number[first_pos+1:last_pos]
243 return float(num_str)
244
245
246 def parse_bnf(gre):
247     os.chdir(global_gre_pwm_source)
248     neper_number = exp(1)
249     dict_gre = {}
250     knot_son = 0
251     with open(gre, 'r') as gre_f:
252         for line_gre in gre_f:
253             sep_pos = line_gre.find("::=")
254             if (sep_pos > 0):
255                 left_line = line_gre[0:sep_pos]
256                 right_line = line_gre[sep_pos+3:]
257                 base_letter = right_line[1:2]
258                 knot_father = get_number(left_line)
259                 if base_letter != "#":
260                     knot_son = get_number(right_line)
261                     if knot_father not in dict_gre:
262                         dict_gre[knot_father] = {}
263                     dict_gre[knot_father][base_letter] = [knot_son, log(get_probability(right_line), neper_number)]
264                 return dict_gre
265
266
267 def out_report1(f_hdl, pwm_gre, tfid, crm, pos_glb, pos_sld_wdw, sld_wdw, mdl, score, cutoff, crit, dict_gre):
268     f_hdl.write(pwm_gre + "\t" + "FT" + str(tfid).zfill(7) + "\t" + crm + "\t"
269                 + str(pos_glb) + "\t" + str(pos_sld_wdw) + "\t" + sld_wdw + "\t")

```

```

270 + mdl + "\t" + str(score) + "\t" + str(cutoff) + "\t" + crit + "\t"
271 + dict_gre[tfid][ "FT_CODE" ] + "\t" + dict_gre[tfid][ "SPECIE_NAME" ]
272 + "\n" )
273
274
275 def proc_fasta(exe_par):
276     global global_compila_main, global_gre_pwm_source, global_pseudocontador_seq, global_gre_error
277     global_gre_error = set()
278     dict_exe_param = parse_exe_par(exe_par)
279     global_compila_main = dict_exe_param[ "AUTOMATON_DIR" ][0]
280     global_gre_pwm_source = dict_exe_param[ "PWM_GRE_DIRECTORY_IN" ][0]
281     global_pseudocontador_seq = float(dict_exe_param[ "PSEUDOCOUNTER" ][0])
282     dict_pwm_parsed = parse_pwm(dict_exe_param[ "FT_LIST" ], dict_exe_param[ "MODELS" ])
283     dict_gre_parsed = parse_gre(dict_exe_param[ "FT_LIST" ], dict_exe_param[ "GRE_TOOL_PARAM" ][0])
284     dict_tf_param = parse_tf_par(dict_exe_param[ "FT_LIST" ], dict_exe_param[ "TF_PARAM" ][0])
285     dict_freq_bas_especie = parse_freq_gen(dict_exe_param[ "GENOME_FREQ" ][0])
286     count_pos_global = -1
287     if (dict_exe_param[ "DEBUG" ][0] == "NO"):
288         console_print = False
289     else:
290         console_print = True
291     fasta_sequences = SeqIO.parse(open(dict_exe_param[ "INPUT_DATA" ][0]), 'fasta')
292     with open(dict_exe_param[ "OUTPUT_DATA" ][0], 'w') as out_file:
293         last_sliding_window = ""
294         last_name = ""
295         out_file.write( "METHOD" + "\t" + "TF_CODE" + "\t" + "CROMOSSOME" + "\t"
296             + "GLOBAL_POSITION" + "\t" + "LOCAL_POSITION" + "\t" + "SLIDING_WINDOW" + "\t"
297             + "MODEL" + "\t" + "SCORE_PWM/GRE" + "\t" + "CUTOFF_PWM/GRE" + "\t" + "CRITERION" + "\t"
298             + "TF_ACRONYM" + "\t" + "SPECIE_NAME" + "\n" )
299     for fasta in fasta_sequences:

```

```

300 sliding_window = ""
301 cromossome, sequence = fasta.id, str(fasta.seq)
302 if (cromossome not in dict_exe_param["CROMOSSOMES"]):
303     continue
304 if (last_name == cromossome):
305     sequence = last_sliding_window + sequence
306 len_seq = len(sequence)
307 for i, base in enumerate(sequence):
308     count_pos_global += 1
309     for tf_id in dict_exe_param["FT_LIST"]:
310         tfbs_len = dict_tf_param["FT" + str(tf_id)].zfill(7)["TF_TFBS_LEN"]
311         sliding_window = sequence[i:i + tfbs_len]
312         if (i > len_seq - tfbs_len):
313             break
314         for model in dict_exe_param["MODELS"]:
315             for criterion in dict_exe_param["CRITERIA"]:
316                 gre_cutoff = dict_gre_parsed[tf_id][model][ "Cutoff_" + criterion]
317                 pwm_cutoff = dict_pwm_parsed[tf_id][model][ "Cutoff:" + criterion + "_mean"]
318                 score_pwm = calc_score_pwm(sliding_window, dict_pwm_parsed[tf_id][model][ "PWM" ])
319                 score_gre = calc_score_gre(tf_id, sliding_window, dict_gre_parsed[tf_id][model][ "GRE" ])
320                 if (console_print):
321                     score_lapfa = calc_score_lapfa(sliding_window, global_gre_pwm_source +
322                                                     dict_gre_parsed[tf_id][model][ "GRE_FILENAME" ])
323                     print("Escore_automato_GrammarLab_" + str(score_lapfa) +
324                           "___Escore_calculado_" + str(score_gre) +
325                           "___(antes_de_descontar_a_freq_de_fundo).")
326                 if (model == "N1"):
327                     score_gre = score_gre - calc_logP_N1(sliding_window)
328                 elif (model == "N2"):
329                     score_gre = score_gre - calc_logP_N2(sliding_window,

```

```

330         dict_freq_bas_especie[dict_gre_parsed[tf_id][SPECIE_NAME]]])
331     elif (model == "N3"):
332         score_gre = score_gre - calc_logP_N3(sliding_window)
333     if score_pwm > pwm_cutoff:
334         out_report1(out_file, "PWM", tf_id, cromossome, count_pos_global, i, sliding_window,
335                     model, score_pwm, pwm_cutoff, criterion, dict_gre_parsed)
336     if score_gre > gre_cutoff:
337         out_report1(out_file, "GRE", tf_id, cromossome, count_pos_global, i, sliding_window,
338                     model, score_gre, gre_cutoff, criterion, dict_gre_parsed)
339     out_file.flush()
340     last_sliding_window = sliding_window
341     last_name = cromossome
342     print ("GBEs_found_with_errors:")
343     print "".join(str(x) for x in global_gre_error)
344
345
346 proc_fasta("exe_parameters.txt")

```

F.5 Como executar o programa de predição de SLFTs

O programa de predição deve ser executado sob o sistema operacional **Linux**. Dependendo do tamanho do trecho de DNA onde os sítios serão preditos, o arquivo de saída `"found.tfbs.txt"` pode ser muito grande e o espaço para esse arquivo deve ser previsto.

É preciso instalar, no **Linux**, um ambiente **Python 2.7** e o pacote **BioPython**¹¹.

É preciso editar o arquivo de parâmetros `exe_parameters.txt`, conforme os parâmetros descritos na seção F.3, definindo-se assim, todos os parâmetros do programa;

O nome e localização do arquivo **FASTA** de entrada deve ser definido pelo parâmetro `"INPUT_DATA"` do arquivo `exe_parameters.txt`. Esse arquivo deve existir nessa localidade.

O nome e localização do arquivo de saída deve ser definido pelo parâmetro `"OUTPUT_DATA"` do arquivo `exe_parameters.txt`. Esse arquivo não precisa existir, mas é preciso que a sua localidade (diretório) exista.

Todos os arquivos de gramáticas “BNF” e “PWM” precisam estar no diretório indicado pelo parâmetro `"PWM_GRE_DIRECTORY_IN"`, definida no arquivo `exe_parameters.txt`.

A localização e o nome dos arquivos `lista_gen_freq.txt`, `tf_parameters.txt` e `gre_tool_parameters.txt` devem ser definidos pelos parâmetros: `"GENOME_FREQ"`, `"TF_PARAM"` e `"GRE_TOOL_PARAM"`, respectivamente, do arquivo `exe_parameters.txt`. Esses arquivos precisam existir nas suas respectivas localizações.

O diretório de compilação do autômato do **GrammarLab** só deve ser utilizado se a opção de debug for utilizada. Neste caso, a instalação do **GrammarLab** se faz necessária, sendo este o diretório para a sua compilação.

No início da execução do programa da listagem F.1, na linha 346:

```
346: proc_fasta("exe_parameters.txt")
```

Se o arquivo `exe_parameters.txt` não estiver no mesmo diretório do programa `tfbs_search.py`, deve-se passar o caminho completo. Por exemplo:

```
346: proc_fasta("/home/aferrao/tfbs_search_tool/exe_parameters.txt")
```

¹¹ Eventualmente, pode ser necessário utilizar um ambiente virtual específico para o **Python 2.7**, sem afetar a instalação original, que possua, por exemplo, o **Python 3.X**. Neste caso, pode-se utilizar um ambiente virtual para o **Python**, como o **Anaconda**, no caso deste exemplo, chamado `"2.7Env"`:

```
>source activate 2.7Env
```

A execução do programa pode ser realizada simplesmente com o comando:

```
>nohup nice python tfbs_search.py &
```