UNIVERSIDADE FEDERAL DE SÃO CARLOS
Programa de Pós-Graduação em Biotecnologia

SimAffling – um ambiente computacional para suporte e simulação do processo de DNA *shuffling*

LUCIANA MONTERA CHEUNG

São Carlos – SP
2008

UNIVERSIDADE FEDERAL DE SÃO CARLOS
Programa de Pós-Graduação em Biotecnologia

SimAffling – um ambiente computacional para suporte e simulação do processo de DNA *shuffling*

Luciana Montera Cheung

Tese apresentada ao Programa de Pós-Graduação em Biotecnologia da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Doutor em Biotecnologia.

Área de Concentração: Biologia Computacional
Orientadores: Prof. Dra. Maria do Carmo Nicoletti
Prof. Dr. Flávio Henrique da Silva

São Carlos
2008

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária/UFSCar

C526sa

Cheung, Luciana Montera.
SimAffling – um ambiente computacional para suporte e simulação do processo de DNA shuffling / Luciana Montera Cheung. -- São Carlos : UFSCar, 2009.
252 f.

Tese (Doutorado) -- Universidade Federal de São Carlos, 2008.

1. Biotecnologia. 2. Simulação (Computadores). 3. Evolução in vitro. 4. DNA shuffling. 5. Modelos de predição. 6. Bioinformática. I. Título.

CDD: 660.6 (20ª)

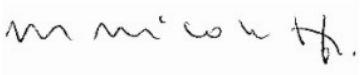
LUCIANA MONTERA CHEUNG

SimAffling – um ambiente computacional para suporte e simulação do processo de DNA *shuffling*

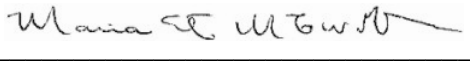
Tese ou Disssertação apresentada à Universidade Federal de São Carlos, como parte dos requisitos para obtenção do título de Doutor em Biotecnologia.

Aprovado em 06 de Novembro de 2008

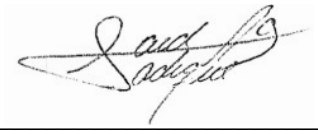
BANCA EXAMINADORA

Presidente  _____

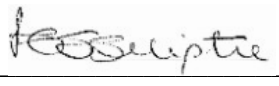
(Orientadora – Profa. Dra. Maria do Carmo Nicoletti)

1º Examinador  _____

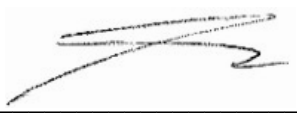
(Universidade de Brasília – Profa. Dra. Maria Emilia Machado Telles Walter)

2º Examinador  _____

(Universidade Federal de Mato Grosso do Sul – Prof. Dr. Said Sadique Adi)

3º Examinador  _____

(Universidade Federal de São Carlos – Profa. Dra. Heloisa Sobreiro Selistre de Araújo)

4º Examinador  _____

(Universidade de São Paulo– Prof. Dr. André Carlos Ponce de Leon Ferreira de Carvalho)

DEDICATÓRIA

À memória da minha querida e admirada mãe Maria Aparecida de Lucca Montera, ao meu pai Antônio Primo Montera, e ao meu amado marido Andrés Batista Cheung, pela importância que representam em minha vida.

AGRADECIMENTOS

Muitas são as pessoas que merecem meu reconhecimento e agradecimento. Começo a agradecer meus orientadores Maria do Carmo Nicoletti e Flávio Henrique da Silva pela disposição e empenho gastos na minha orientação. Aos professores Cláudio Alberto Torres Suazo e Fernando Manuel Araújo Moreira que passaram pela coordenação do Programa de Pós-Graduação em Biotecnologia durante meu período de doutoramento, pelo trabalho em direção da melhoria do nosso programa, bem como pela atenção com que os assuntos referentes a mim sempre foram tratados. A todos os professores do programa que, seja por meio de suas disciplinas ou por conversas informais, colaboraram para o meu desenvolvimento acadêmico e profissional.

Ao professor Pablo Moscato da The University of Newcastle por ter me recebido na cidade de Newcastle – Austrália, durante o período de estágio de doutorado no exterior. As pesquisas realizadas junto ao seu grupo serviram não apenas para o meu trabalho de doutorado, mas também como uma porta de entrada para trabalhos futuros.

Ao colega de estudo Moacir Ponti por ter se mostrado um companheiro para todas as horas. Muitos outros colegas encontrei durante o curso, cada um com seu jeito e carisma únicos sempre serão lembrados, entre eles Marcia Dellamano, Sidnei Pereira, Adilson Silva, Antônio Carlos Horta, Alexandre Jahnecke e Márcia Cominetti.

Um agradecimento especial à professora e orientadora Maria do Carmo Nicoletti que, de forma especial, muito me ensinou e contribuiu para minha formação durante este trabalho de pesquisa.

À CAPES pela bolsa de estudos concedida durante os quatro anos de duração deste trabalho, incluindo o período de estágio no exterior.

Ao meu marido e amigo Andrés Batista Cheung que, com seu entusiasmo pela pesquisa, sempre me apoiou e me incentivou nos momentos difíceis que tive durante o desenvolvimento deste trabalho. A toda a minha família, especialmente minha mãe, pai, irmão, irmã, cunhados e sobrinhos que pelo "simples" fato de existirem, tornam minha vida muito mais feliz.

À minha mãe, que viu o começo, o meio, mas não o fim deste trabalho. Nenhuma palavra que eu escrevesse aqui seria o bastante para expressar a gratidão e o amor que tenho por ela. Mulher de força, coragem e sabedoria, onde sempre encontrei tudo o que precisei.

RESUMO

LUCIANA, M.C. **SimAffling – um ambiente computacional para suporte e simulação do processo de DNA *shuffling*.** 2008. 252 folhas. Tese (Doutorado) – Programa de Pós-Graduação em Biotecnologia, Universidade Federal de São Carlos, São Carlos - SP, 2008.

A Evolução Molecular dos organismos vivos é um processo lento que ocorre ao longo dos anos e diz respeito às mutações e recombinações sofridas por um determinado organismo em seu material genético, ou seja, em seu DNA. As mutações ocorrem na forma de remoções, inserções e/ou substituições de nucleotídeos ao logo da cadeia de DNA. A Evolução Molecular Direta é um processo laboratorial, ou seja, *in vitro*, que visa melhorar funções biológicas específicas de moléculas por meio de mutações/recombinações em seu material genético, imitando o processo natural de evolução. Diversas técnicas que simulam a evolução molecular em laboratório, entre elas a técnica de DNA *shuffling*, têm sido amplamente utilizadas na tentativa de melhorar determinadas propriedades de uma variedade de produtos comercialmente importantes como vacinas, enzimas industriais e substâncias de interesse famacológico. A metodologia original de DNA *shuffling* pode ser sumarizada pelas seguintes etapas: 1) seleção dos genes de interesse, dito parentais; 2) fragmentação enzimática dos genes; 3) ciclos de PCR (*Polymerase Chain Reaction*), para que ocorra a remontagem dos fragmentos; 4) amplificação das seqüências remontadas cujo tamanho é igual a dos parentais. O sucesso ou não da técnica de DNA *shuffling* pode ser medido pelo número de moléculas recombinantes encontradas na biblioteca de DNA *shuffling* obtida, uma vez que estas podem apresentar melhorias funcionais em relação aos parentais pelo fato de, possivelmente, acumularem em sua seqüência mutações benéficas presentes em parentais distintos. Atualmente podem ser encontradas na literatura algumas poucas modelagens computacionais capazes de sugerir otimizações para o processo, com vistas em aumentar a diversidade genética da biblioteca resultante. O presente trabalho apresenta um estudo comparativo de quatros modelos para predição/estimativa de resultados de experimentos de DNA *shuffling* encontrados na literatura bem como a proposta e implementação de uma ferramenta computacional de simulação para o processo de DNA *shuffling*. A ferramenta de simulação foi implementada em um ambiente que disponibiliza outras funcionalidades referentes à análise das seqüências a serem submetidas ao *shuffling* bem como ferramentas para análise das seqüências resultantes do processo.

Palavras-chave: Simulação computacional. Evolução *in vitro*. DNA *shuffling*. Modelos de predição. Biologia Computacional. Bioinformática.

ABSTRACT

LUCIANA, M.C. **SimAffling – um ambiente computacional para suporte e simulação do processo de DNA *shuffling*.** 2008. 252 folhas. Thesis (Doctoral) – Programa de Pós-Graduação em Biotecnologia, Universidade Federal de São Carlos, São Carlos - SP, 2008.

The Molecular Evolution of the living organisms is a slow process that occurs over the years producing mutations and recombinations at the genetic material, i.e. at the DNA. The mutations can occur as nucleotide remotion, insertion and/or substitution at the DNA chain. The Directed Molecular Evolution is an *in vitro* process that tries to improve biological functions of specific molecules producing mutations at the molecule's genetic material, mimicking the natural process of evolution. Many technics that simulate *in vitro* molecular evolution, among them the DNA shuffling, have been used aiming to improve specific properties of a variety of commercially important products as pharmaceutical proteins, vaccines and enzymes used in industries. The original DNA shuffling methodology can be sumarized by the following steps: 1) selection of the parental sequences; 2) random fragmentation of the parental sequences by an enzyme; 3) repeated cycles of PCR (Polymerase Chain Reaction), in order to reassemble the DNA fragments produced in the previous step; 4) PCR amplification of the reassembled sequences obtained in step 3). The DNA shuffling technic success can be measured by the number of recombinat molecules found at the DNA shuffling library obtained, since these recombinant molecules potentially have improved functionalities in relation to their parent since their sequence may accumulate beneficial mutations originated from distinct parent sequences. Nowadays some few models can be found in the literature whose purpose is to suggest optimization to this process aiming the increase of the genetic diversity of the DNA shuffling library obtained. This research work presents a comparative study of four models used to predict/estimate the DNA shuffling results. In addition a computational tool for simulating the DNA shuffling proccess is proposed and implemented in an environment where other functionalities related to the analyses of the parental sequences and the resulting sequences from the DNA shuffling library is also implemented.

Keywords: Computational simulation. *In vitro* evolution. DNA shuffling. Predicting models. Computational Biology. Bioinformatics.

LISTA DE TABELAS

Tabela 1.1. Código, nome e códons dos vinte aminoácidos..... 30

Tabela 1.2. Código Genético. 31

Tabela 1.3. Seqüências de reconhecimento de algumas endonucleases de restrição, com indicação dos respectivos sítios de clivagem (→) e pontos de simetria (•)..... 39

Tabela 3.1. Representação de todas as possíveis variantes do..... 88

Tabela 3.2. Estatísticas do alinhamento entre os pares de seqüências..... 94

Tabela 3.3. Resultados estimados pelo DRIVeR considerando o *shuffling* 95

Tabela 3.4. Porcentagem de cobertura do espaço amostral para..... 98

Tabela 3.5. Relação entre a variação da energia..... 107

Tabela 3.6. Valores de variação de entropia (ΔH) e entalpia (ΔS) para os pares de bases vizinhos..... 109

Tabela 3.7. Probabilidade de que uma região alvo de..... 121

Tabela 4.1. Mutações encontradas entre os parentais Oryza e SD. Bases vizinhas foram 141

Tabela 5.1. Estimativas do SimAffling considerando fragmentos de tamanho 10 pb. 180

Tabela 5.2. Estimativas do SimAffling considerando fragmentos de tamanho 20 pb. 180

Tabela 5.3. Estimativas do SimAffling considerando fragmentos de tamanho 30 pb. 180

Tabela 5.4. Estimativas do SimAffling considerando fragmentos de tamanho 40 pb. 181

Tabela 5.5. Seqüências parentais utilizadas para validação de modelos para o DNA *shuffling*..... 189

Tabela 5.6. Comparação entre estimativas e valores experimentais para casos de estudo de DNA *shuffling*. 192

LISTA DE FIGURAS

Figura 1.1. Estrutura das quatro bases nitrogenadas Adenina, Timina, Citosina e Guanina presentes na molécula de DNA e da base Uracila que substitui a Timina na molécula de RNA. 24

Figura 1.2. Pentose presente na molécula de DNA. (a) Visualização linear. (b) Visualização estrutural. 24

Figura 1.3. Um nucleotídeo, o qual é composto por um grupo fosfato, um açúcar e uma base nitrogenada. (a) Representação simplificada. (b) Representação não simplificada. 25

Figura 1.4. Ligação fosfodiéster que une dois nucleotídeos. 25

Figura 1.5. Pontes de hidrogênio entre os pares de bases (a) A–T e (b) C–G. 26

Figura 1.6. Representação de um fragmento de molécula de DNA..... 27

Figura 1.7. Representação simplificada do mecanismo de duplicação do DNA..... 28

Figura 1.8. Dogma Central da Biologia Celular. 28

Figura 1.9. Transcrição de DNA em RNA. 29

Figura 1.10. Síntese de uma proteína a partir de uma região do DNA correspondente a um gene..... 31

Figura 1.11. Síntese de proteína pelo ribossomo. Os anticódons dos tRNAs ligam-se aos códons do mRNA dentro do ribossomo e a cadeia polipeptídica vai sendo formada pela união dos aminoácidos. 32

Figura 1.12. Estrutura geral de um aminoácido. 33

Figura 1.13. Atuação da enzima DNA polimerase na replicação do DNA. (a) A polimerase é capaz de estender o *primer1* criando uma fita complementar ao molde. (b) A polimerase não é capaz de estender o *primer2* – uma extensão não ocorre a partir da extremidade 5'..... 35

Figura 1.14. Ciclos de PCR com *primer*s. A cada ciclo, uma molécula de DNA dá origem a duas novas moléculas idênticas à molécula original. 36

Figura 1.15. Esquema da construção *in vitro* de moléculas de DNA recombinante. (a) As seqüências são fragmentadas por uma mesma enzima de restrição. (b) Os fragmentos resultantes são incubados sob condições de pareamento entre fragmentos complementares. (c) A enzima DNA ligase é adicionada para garantir a formação de ligações fosfodiésteres entre os fragmentos remontados..... 40

Figura 1.16. Mapa de restrição do vetor pUC8..... 41

Figura 1.17. Esquema do procedimento geral para a clonagem de um fragmento de DNA utilizando um plasmídeo como vetor. (a) Fragmentação do plasmídeo e da molécula de DNA por uma mesma enzima de

restrição (Enz1) e inserção do fragmento a ser clonado no plasmídeo, resultando em uma molécula de DNA recombinante. (b) Inserção da molécula de DNA recombinate na célula hospedeira. (c) Crescimento (multiplicação) da célula hospedeira, bem como da molécula de DNA recombinante. 42

Figura 2.1. Representação esquemática geral do processo de evolução 46

Figura 2.2. Apresentação cronológica de diversas técnicas de evolução *in vitro*. 47

Figura 2.3. Representação esquemática da SDM no início da seqüência. A mutação ocorre pela substituição da Adenina, localizada na sexta posição da seqüência original, pela Guanina nas seqüências resultantes (bases sombreadas de preto). Note que apenas uma das fitas da molécula original é utilizada durante a PCR e que, a cada ciclo, novas seqüências mutantes (fita simples) são produzidas. 49

Figura 2.4. Representação esquemática da SDM no meio da seqüência. A mutação irá ocorrer devido ao pareamento incorreto (*mismatch*) entre uma base da molécula de fita simples (representada pelo retângulo branco) e uma base do *primer* (representado pelo retângulo preto). (a) Reação PCR 1 para amplificação da primeira metade da seqüência original resultando em moléculas de fita simples com a mutação desejada. (b) Reação PCR 2 para amplificação da segunda metade da seqüência original resultando em moléculas de fita simples com a mutação desejada. (c) e (d) Produtos resultantes das reações PCR 1 e PCR 2, respectivamente. Os produtos resultantes de ambas as reações são submetidos a uma mesma reação de PCR, sem a adição de *primers*. (e) Recombinação entre os produtos das reações de PCR 1 e PCR 2, por meio do pareamento entre regiões complementares. (f) Extensão dos fragmentos recombinados, resultando nas seqüências com a mutação desejada. 50

Figura 2.5. Representação esquemática do método de mutação sítio-dirigida. (a) Na primeira reação de PCR a mutação desejada é introduzida (representada pelo pequeno trecho branco no *primer* 1), porém apenas uma porção do gene é amplificada, como mostrado em (b). (c) Na segunda reação de PCR, o produto da primeira reação é utilizado como *primer* juntamente com um novo *primer* para amplificar o gene por inteiro já com a mutação desejada. (d) Como resultado da segunda reação têm-se o gene, bem como uma fração deste amplificados. Após o tratamento do produto desta última reação pela enzima de restrição apropriada, os fragmentos que correspondem ao gene original, agora com a mutação desejada (e), podem ser clonados em vetores apropriados..... 51

Figura 2.6. Representação esquemática da produção de seqüências recombinantes seguindo a metodologia StEP. Apenas uma das fitas de cada seqüência, e seus respectivos *primers*, são mostrados para simplificar a representação. (a) Pareamento entre os *primers* e as seqüências molde. (b) Extensão dos *primers*. (c) A temperatura da reação é elevada para que os *primers* se soltem dos moldes (desnaturação) e a extensão seja interrompida. (d) Variações de temperatura favorecem o *switch* entre *primers* e seqüência molde. (e) Após a execução cíclica das etapas (b), (c) e (d), tem-se as seqüências resultantes. 54

Figura 2.7. Representação esquemática do processo de DNA *shuffling* entre dois parentais. (a) Fragmentação enzimática das seqüências parentais. (b) Desnaturação dos fragmentos resultando em

fragmentos de fita simples. (c) Pareamento entre fragmentos de fita simples que compartilham regiões de bases complementares. (d) Extensão por polimerase dos fragmentos pareados. (e) Exemplo de uma seqüência resultante do processo de *shuffling*, a qual é composta pela união de fragmentos originários de ambos os parentais. 56

Figura 2.8. Representação dos possíveis pareamentos entre dois fragmentos de tamanhos m e n. 59

Figura 2.9. Representação de possíveis configurações de pareamento entre dois fragmentos para o qual é possível apenas uma extensão parcial pela polimerase. Para este tipo de configuração, a extensão nunca irá resultar em uma molécula de fita dupla completa. 59

Figura 2.10. Representação esquemática detalhada das etapas do processo de DNA *shuffling* sem a etapa final de amplificação: (a) Seleção dos Parentais. (b) Fragmentação. (c) Desnaturação. (d) Pareamento. (e) Extensão. Os pareamentos indicados por A_1, A_3, A_4 e A_5 correspondem a heteroduplexes, enquanto que em A_2 é formado um homoduplex. 60

Figura 2.11. Esquema geral da metodologia *Random-priming recombination*. (a) Síntese de pequenos fragmentos de fita simples complementares aos parentais a partir de *primers* randômicos. As posições marcadas com X's representam novas mutações introduzidas pela reação de PCR. (b) Remoção das seqüências parentais. (c) Remontagem e amplificação dos fragmentos resultantes da reação (b)..... 64

Figura 2.12. Representação da atividade da enzima Exo III, que remove nucleotídeos a partir da extremidade 3' OH. 65

Figura 2.13. Atuação da enzima Nuclease S1, que remove nucleotídeos a partir da extremidade 5', resultando em fragmento *blunt end.* 66

Figura 2.14. Representação esquemática da fragmentação de dois parentais pela enzima Exo III, seguida por tratamento com Nuclease S1 e ligação dos fragmentos por Ligase. (a) Os parentais são tratados com Exo III. (b) Pela ação da Nuclease S1 as extremidades não pareadas são eliminadas. (c) Os fragmentos resultantes da ação da Exo III e da Nuclease S1 são ligados pela enzima Ligase, resultando em um fragmento híbrido. 67

Figura 2.15. Representação esquemática da metodologia ITCHY. (a.1) e (a.2) As seqüências parentais são inseridas em vetores apropriados. (b.1) Os vetores do tipo 1 são fragmentados pela enzima de restrição que reconhece o sítio E_1. (b.2) Os vetores do tipo 2 são fragmentados pela enzima de restrição que reconhece o sítio E_2. Nas reações (b.1) e (b.2) as enzimas Exo III e Nuclease S1 também são adicionadas, e alíquotas de ambas as reações são retiradas em intervalos de tempo determinado para que sejam obtidos fragmentos de diferentes tamanhos. (c.1) e (c.2) Os fragmentos resultantes de ambas as reações são purificados. (d) União dos fragmentos resultantes. 68

Figura 2.16. Representação esquemática da técnica de *Family shuffling* utilizando apenas ssDNA complementares dos parentais. (a) Moléculas de fita dupla de DNA dos parentais 1 e 2. (b) Fitas simples

de DNA (ssDNA) de orientações opostas correspondentes aos parentais. (c) Fragmentação enzimática das ssDNA. (d) Pareamento entre fragmentos originários de parentais distintos. (e) Extensão por polimerase resultando em fragmentos heteroduplexes. 70

Figura 2.17. Mecanismo de reparo independente. (a) As seqüências parentais 1 e 2 diferem em dois pares de bases M_1 e M_2 os quais estão representados, respectivamente, por círculos e quadrados pretos no Parental 1 e por círculos e quadrados brancos no Parental 2. (b) Amplificação dos parentais. (c) e (d) Seqüências resultantes após a desnaturação e o pareamento. (c) Seqüências pareadas com *mismatches*. (d) Seqüências perfeitamente pareadas. (e) Reparo independente das seqüências com *mismatches* ((c)). (f) Seqüências resultantes do reparo que se tornaram idênticas aos parentais. (g) Seqüências resultantes do reparo, as quais podem ser vistas como o resultado de um cruzamento entre os parentais. 72

Figura 2.18. *Heteroduplex recombination*, variante A. (a.1) e (a.2) As seqüências parentais são inseridas em vetores de clonagem iguais em duas reações distintas. (b.1) e (b.2) Cada uma das reações é tratada com uma enzima de restrição específica; Enz 1 para reação (a.1) e Enz 2 para a reação (a.2), resultando na linearização das moléculas inicialmente circulares. Em seguida, o produto de ambas as reações é misturado, desnaturado e submetido a temperaturas ideais para que ocorra o pareamento entre os fragmentos. Como resultando deste pareamento, quatro moléculas distintas podem ocorrer, sendo estas: (c) Heteroduplexes e circulares ou (d) Homoduplexes e lineares. Apenas moléculas circulares podem ser eficientemente utilizadas para transformar bactérias. 73

Figura 2.19. *Heteroduplex recombination*, variante B. (a.1) e (a.2) Os fragmentos parentais são inseridos em vetores de expressão iguais em duas reações distintas. (b.1) e (b.2) Reações de amplificação por PCR são executadas para que apenas as seqüências parentais aumentem em número. Em seguida, apenas os fragmentos amplificados são selecionados. (c) Os fragmentos purificados são desnaturados e remontados, resultando, possivelmente, em fragmentos heteroduplexes. (d) Os fragmentos remontados, assim como os vetores que irão receber estes fragmentos, são digeridos com enzimas de restrição e remontados em uma mesma molécula. Os heteroduplexes circulares estão prontos para serem utilizados na transformação de bactérias. 74

Figura 2.20. Representação esquemática da metodologia SCRATCHY. (a) Seqüência Parental 1 e Parental 2. (b) Fragmentos truncados em ambas as direções dos parentais (5'→3' e 3'→5') resultantes do ITCHY. (c) Ligação randômica entre os fragmentos resultantes. (d) Fragmentação e desnaturação dos fragmentos. (e) Remontagem dos fragmentos segundo o protocolo de DNA *shuffling*. 76

Figura 3.1. Representação do evento de cruzamento entre dois parentais. (a) Parentais A e B que diferem entre si em dois pares de bases identificados no Parental A com fundo preto e no Parental B com fundo branco. Em (a) as setas indicam as posições onde a molécula de DNA será cortada pela enzima de fragmentação. (b) Fragmentos resultantes da ação da enzima. (c) Desnaturação dos fragmentos. (d)

Pareamento entre os fragmentos F_1 e F_2, os quais são originários de parentais distintos. (e) Extensão dos fragmentos pareados resultando em um fragmento heteroduplex. 83

Figura 3.2. Seqüências parentais que diferem em dois pares de bases. (a) Representação por fita dupla. (b) Representação simplificada contendo apenas uma fita e as bases que diferenciam uma seqüência da outra, representadas pelos símbolos ■ e o. 83

Figura 3.3. Possíveis pontos de corte da enzima DNase I. (a) Caso a enzima produza cortes entre as duas mutações consecutivas, há chance do evento de cruzamento ocorrer entre os fragmentos resultantes. (b) Caso os cortes produzidos pela enzima não estejam localizados entre as mutações, a possível recombinação dos fragmentos não irá gerar um heteroduplex, uma vez que as bases distintas entre as moléculas parentais permaneceram em um mesmo fragmento. 84

Figura 3.4. Parentais A e B, nos quais cada uma das M = 5 bases distintas entre eles estão assinaladas com um quadrado preto no Parental A e com um círculo branco no Parental B e a representação da forma geral de uma variante k do total de $2^M - 2$ variantes possíveis, resultante da ocorrência de todos os possíveis cruzamentos entre os parentais A e B. 86

Figura 3.5. Duas seqüências parentais A e B e uma variante k resultante de cruzamentos entre os parentais. B_k é a representação binária da seqüência variante k e é função do número de cruzamentos entre as bases consecutivas que diferem entre si nos parentais. 86

Figura 3.6. Valores de n_i para os parentais A e B. 89

Figura 3.7. Exemplos de pontos de cruzamentos que podem gerar cruzamentos silenciosos. 91

Figura 3.8. Exemplo do pareamento entre dois fragmentos de DNA no qual um *gap* ocorreu. 100

Figura 3.9. Gráficos comparativos entre os resultados *in silico* (modelo) e *in vitro* (experimento) do DNA *shuffling* do gene gfp. 103

Figura 3.10. Duas regiões de sobreposição entre dois pares de fragmentos de DNA distintos. Os pares de bases vizinhos e consecutivos estão indicados pelos símbolos ⊓ e ⊔. 108

Figura 3.11. Diferentes pareamentos entre o *template* A e cada um dos fragmentos F_1, F_2, F_3 e F_4, com diferentes tamanhos de sobreposição, contendo ou não *mismatches* (destacados com por um retângulo). 111

Figura 3.12. Representação genérica do pareamento entre dois fragmentos de DNA com uma região de sobreposição de tamanho v. 111

Figura 3.13. Esquema representativo do processo de remontagem dos fragmentos. No ciclo de remontagem dos fragmentos, o fragmento resultante em (c) passa a ser o *template* em (a) ao qual outro fragmento irá se parear, e assim sucessivamente. 114

Figura 3.14. Possíveis sobreposições entre dois fragmentos originários dos parentais m e k, ambos de tamanho L, os quais podem resultar em cruzamentos, caso m ≠ k. 115

Figura 3.15. Ilhas e Oceanos. Ilhas formadas por: (a) três, (b) dois e (c) um fragmento. As regiões de espaços (*gaps*) entre as ilhas são os oceanos. 118

Figura 3.16. Parentais A e B, os quais diferem entre si em apenas duas bases, ditas mutações M_1 e M_2 que estão separadas por t bases. 120

Figura 3.17. Exemplo de pareamento com *mismatch* (*). 121

Figura 3.18. Esquema de remontagem dos fragmentos. (a) Sobreposição entre fragmentos com extremidades complementares. (b) Após a extensão das extremidades 3' dos fragmentos complementares, tem-se uma ilha. 122

Figura 3.19. Distintos tamanhos de L. (a) Dois parentais A e B que diferem em dois pares de bases distantes entre si por t pares de bases. (b) Fragmentos resultantes da fragmentação enzimática cujo tamanho é L > t. (c) Fragmentos resultantes da ação enzimática cujo tamanho é L ≤ t. 124

Figura 4.1. Arquitetura do sistema ISAS e principais funcionalidades. 128

Figura 4.2. Subsistema *Sequence Basics* e suas funcionalidades. A cada uma das quatro bases que compõem a seqüência de DNA é atribuída uma cor característica para auxiliar na visualização da composição da seqüência. 130

Figura 4.3. Subsistema *Sequence Basics*. Seqüência de DNA com os códons em destaque. 131

Figura 4.4. Subsistema *Sequence Basics*. Estatíticas dos códons e aminoácidos que compõem a seqüência. 132

Figura 4.5. Busca por *primers*. Tela de execução da função que, dada uma seqüência de DNA, encontra um par de *primers* para a amplificação da mesma. O gráfico *Function value through the iteration* representa o valor associado à adequabilidade dos pares de *primers* encontrados ao logo da busca. 133

Figura 4.6. Seqüência de nucleotídeos de duas cistatinas, Oryza e SD, armazenadas no formato FASTA. 135

Figura 4.7. *Pairwise Parental Analyses*. Alinhamento entre as seqüências Oryza e SD. 136

Figura 4.8. *Pairwise Parental Analyses*. Mutações encontradas entre os parentais Oryza e SD (destacadas em cinza) 138

Figura 4.9. *Pairwise Parental Analyses*. Regiões de mutação entre os parentais representadas na forma condensada no alinhamento. 139

Figura 4.10. *Multiple Parental Analyses*. Análise de 37 seqüências candidatas a parentais para as quais foram calculadas a matriz de *score* (derivada dos alinhamentos), a matriz de distância média entre as mutações consecutivas e a matriz de distância baseada em mutações, para todos os pares de seqüências. 144

Figura 4.11. Representação gráfica da adequabilidade dos pares de seqüências avaliados segundo a medida baseada em mutações. 145

Figura 4.12. *Shuffling Simulations*. Simulações do processo de DNA *shuffling* entre os parentais Oryza e SD utilizando o DRIVeR. 148

Figura 4.13. *Shuffling Simulations*. Simulação do processo de DNA *shuffling* entre os parentais Oryza e SD utilizando o eShuffle. 150

Figura 4.14. *Post Shuffling*. Busca por contaminantes nas seqüências *shuffled* durante o processo de *Library Housekeeping*. 153

Figura 4.15. *Post Shuffling*. Busca pelo complemento reverso dos *primers* utilizados na amplificação das seqüências *shuflled*. 154

Figura 4.16. Sequencia *shuffled* A05 na qual uma aproximação do *primer forward* foi encontrada (seqüência sublinhada) 155

Figura 4.17. *Post Shuffling*. Busca nas seqüências *shuffled* por mutações originárias dos parentais. 157

Figura 4.18. *Post Shuffling*. Alinhamento entre a seqüência de aminoácidos correspondentes à seqüência *shuffled* H07 e os parentais Oryza e SD. 158

Figura 5.1. Esquema representativo da fragmentação dos parentais pelo eShuffle. 163

Figura 5.2. Pseudo-código do SimAffling (conclusão). 167

Figura 5.3. Esquema de fragmentação das seqüências parentais. 170

Figura 5.4. Todas as possíveis sobreposições entre os fragmentos f1 e f 2. 172

Figura 5.5. Possíveis casos de extensão de fragmentos pareados. 175

Figura 5.6. Esquema de um evento de cruzamento entre dois fragmentos remontados. 176

Figura 5.7. Duas seqüências de tamanho 90 pb que diferem entre si em 4 pb, destacadas em preto. 179

Figura 5.8. Tela de execução do Simulador de DNA shuffling. 181

LISTA DE GRÁFICOS

Gráfico 3.1. Influência dos parâmetros L e λ^{obs} no número C de seqüêmcias variantes esperadas em uma biblioteca resultante do *shuffling* de duas seqüências, ambas de tamanho N = 1.500, que diferem em M = 9 pares de bases. 92

Gráfico 3.2. Influência dos valores do parâmetro n_i no número total C de variantes esperadas em uma biblioteca de *shuffling*, para valores fixos de N = 1.500, M = 9 e λ^{true} = 8. 93

Gráfico 3.3. Número observado de cruzamentos *versus* número esperando de variantes distintas estimados pelo DRIVeR considerando bibliotecas de DNA *shuffling* de tamanho 1.000. 96

Gráfico 3.4. Número observado de cruzamentos *versus* número esperando de variantes distintas estimados pelo DRIVeR considerando bibliotecas de DNA *shuffling* de tamanho 5.000. 96

Gráfico 3.5. Número observado de cruzamentos *versus* número esperando de variantes distintas estimados pelo DRIVeR considerando bibliotecas de DNA *shuffling* de tamanho 10.000. 97

Gráfico 5.1. Número médio de cruzamentos e % de seqüências *full-length* obtidas na simulação do *shuffling* entre os parentais seq1 e seq2, para 500 simulações com diferentes tamanhos iniciais de fragmentos (L), usando o SimAffling. 182

Gráfico 5.2. Histograma do tamanho dos fragmentos remontados ao final de 1.000 simulações partindo-se de fragmentos de tamanho 10 pb. A barra do histograma destacada em azul representa a porcentagem de fragmentos remontados cujo tamanho está entre 87,5 e 92,5 pb. 183

Gráfico 5.3. Histograma do tamanho dos fragmentos remontados ao final de 1.000 simulações partindo-se de fragmentos de tamanho 20 pb. A barra do histograma destacada em azul representa a porcentagem de fragmentos remontados cujo tamanho está entre 87,5 e 92,5 pb. 183

Gráfico 5.4. Histograma do tamanho dos fragmentos remontados ao final de 1.000 simulações partindo-se de fragmentos de tamanho 30 pb. A barra do histograma destacada em azul representa a porcentagem de fragmentos remontados cujo tamanho está entre 87,5 e 92,5 pb. 184

Gráfico 5.5. Histograma do tamanho dos fragmentos remontados ao final de 1.000 simulações partindo-se de fragmentos de tamanho 40 pb. A barra do histograma destacada em azul representa a porcentagem de fragmentos remontados cujo tamanho está entre 90,0 e 94,0. 184

Gráfico 5.6. Convergência do simulador ao longo de 100 simulações. 185

Gráfico 5.7. Convergência do simulador ao longo de 500 simulações. 186

Gráfico 5.8. Convergência do simulador ao longo de 1.000 simulações. 186

Gráfico 5.9. Relação entre o número médio de cruzamentos, o número de fragmentos a serem remontados e o tempo gasto na simulação. 188

SUMÁRIO

INTRODUÇÃO **20**

1 BIOLOGIA MOLECULAR – CONCEITOS E PROCESSOS BÁSICOS **22**

1.1 INTRODUÇÃO 22
1.2 DNA E RNA 23
1.3 GENES, PROTEÍNAS E SÍNTESE DE PROTEÍNAS 28
1.4 SÍNTESE *IN VITRO* DE DNA E REAÇÃO DE PCR 33
1.5 CLONAGEM E TECNOLOGIA DO DNA RECOMBINANTE 37
1.6 CONSIDERAÇÕES FINAIS 42

2 TÉCNICAS DE EVOLUÇÃO MOLECULAR DIRETA **44**

2.1 INTRODUÇÃO 44
2.2 EVOLUÇÃO *IN VITRO* E O PROCESSO DE MUTAGÊNESES 45
2.3 *SITE-DIRECTED MUTAGENESIS* 47
2.4 *OLIGONUCLEOTIDE-DIRECTED RANDOM MUTAGENESIS* 51
2.5 *STAGGERED EXTENSION PROCESSES* (STEP) 53
2.6 DNA *SHUFFLING* 54
2.6.1 Seleção dos genes de interesse *55*
2.6.2 Fragmentação *56*
2.6.3 Ciclos de PCR *57*
2.6.4 Amplificação *61*
2.7 *FAMILY SHUFFLING* 61
2.8 *RANDOM-PRIMING RECOMBINATION* 62
2.9 *RESTRICTION FRAGMENT SHUFFLING* 63
2.10 ITCHY 65
2.11 *SINGLE-STRANDED* DNA 69
2.12 *HETERODUPLEX RECOMBINATION* 70
2.13 SCRATCHY 75
2.14 CONSIDERAÇÕES FINAIS 76

3 REVISÃO DE QUATRO MODELOS PARA O PROCESSO DE DNA *SHUFFLING* **78**

3.1 INTRODUÇÃO 78
3.2 O MODELO DRIVER 80
3.2.1 Introdução *80*
3.2.2 Contextualização *81*
3.2.3 O cruzamento no modelo DRIVeR *82*
3.2.4 Descrição do modelo DRIVeR *84*
3.2.5 Usando o DRIVeR *93*
3.3 O MODELO SHUFFIT 98
3.3.1 Introdução *98*
3.3.2 Descrição do modelo *99*
3.4 O MODELO ESHUFFLE 103
3.4.1 Introdução *103*
3.4.2 Conceitos de termodinâmica e o modelo Nearest–Neighbor *104*
3.4.3 O modelo de pareamento *110*
3.4.4 Remontagem dos fragmentos *113*
3.5 O MODELO SUN 117
3.5.1 Introdução *117*
3.5.2 O modelo de distribuição de clones randômicos *117*
3.5.3 O modelo de recombinação *122*
3.6 CONSIDERAÇÕES FINAIS 125

4 PROPOSTA E IMPLEMENTAÇÃO DO ISAS, UMA FERRAMENTA COMPUTACIONAL PARA APOIO AO PROCESSO *IN VITRO* DE DNA *SHUFFLING* **127**

4.1 INTRODUÇÃO 127

4.2 O SISTEMA ISAS – ARQUITETURA E PRINCIPAIS FUNCIONALIDADES 128
4.2.1 O subsistema Sequence Basics 129
4.2.2 O subsistema Pre Shuffling 133
4.2.2.1 *Parental Analyses* 134
4.2.2.2 *Shuffling Simulations* 145
4.2.3 O subsistema Post Shuffling 150
4.2.3.1 *Library Housekeeping* 151
4.2.3.2 *Search for Recombinants* 155
4.2.3.3 *Parental and Shuffled Sequence Alignment* 157
4.3 CONSIDERAÇÕES FINAIS 159

5 SIMAFFLING: SIMULADOR DE DNA *SHUFFLING* 161

5.1 INTRODUÇÃO – MOTIVAÇÃO PARA A PROPOSTA SIMAFFLING 161
5.2 SIMULADOR DE DNA *SHUFFLING* 164
5.3 CONSIDERAÇÕES SOBRE A CONVERGÊNCIA DO SIMAFFLING E TESTES PRELIMINARES 177
5.4 COMPARAÇÃO ENTRE RESULTADOS SIMULADOS E EXPERIMENTAIS 188
5.5 CONSIDERAÇÕES FINAIS 193

6 CONCLUSÕES 194

APÊNDICE A – DRIVER 210

APÊNDICE B – SHUFFIT 214

APÊNDICE C – ESHUFFLE 218

APÊNDICE D – DETERMINAÇÃO DE *PRIMERS* UTILIZANDO *SIMULATED ANNEALING* 232

APÊNDICE E – MEDIDA PARA AVALIAR A ADEQUABILIDADE DE PARENTAIS CANDIDATOS AO PROCESSO DE DNA *SHUFFLING* – UMA PROPOSTA BASEADA EM MUTAÇÕES 241

APÊNDICE F – CONCEITOS DE ESTATÍSTICA 249

Introdução

"Adversidades são grandes oportunidades."
Provérbio Árabe

O DNA *shuffling* é uma técnica de Evolução Molecular Direta, ou simplesmente, Evolução *in vitro*, que objetiva a produção de moléculas de DNA recombinates em laboratório por meio da recombinação entre fragmentos de DNA de diferentes origens. Espera-se que as recombinações entre fragmentos de diferentes origens, ditos parentais, dêem origem a moléculas recombinates, que podem representar novos genes, por exemplo, com funcionalidades melhoradas ou novas em relação aos parentais.

O protocolo de DNA *shuffling* envolve diversas etapas que requerem tempo, além de investimento em recursos humanos e materiais. Neste sentido, o presente trabalho de pesquisa apresenta um estudo de algums modelos que podem ser utilizados na estimativa de resultados de experimentos desse tipo, bem como ferramentas auxiliares para que otimizações sejam implementadas antes que o experimento de DNA *shuffling* seja realizado em laboratório. Além do estudo e da descrição dos quatro modelos encontrados na literatura, foi proposto e implementado como um software, outro modelo para que o processo de DNA *shuffling* possa ser simulado *in silico* e os resultados utilizados para sugerir otimazações para o processo *in vitro*. O simulador de DNA *shuffling* implementado foi disponibilizado em um ambiente computacional

que implementa outras funcionalidades destinadas ao apoio e suporte a algumas etapas envolvidas no processo de DNA *shuffling*.

Uma medida de similaridade entre seqüências de DNA foi proposta para permitir a análise/avaliação de um conjunto de seqüências candidatas, duas-a-duas, a parentais em experimentos de DNA *shuffling* com o objetivo de determinar qual par, ou quais pares de seqüências, se utilizadas como parentais, produziriam melhores resultados.

Adicionalmente, foi proposta e implementada uma modelagem baseada no algoritmo de *Simulated Annealing* para o problema de projeto de *primers*. A modelagem apresentada tem especial importância visto que *primers* são utilizados em duas etapas distintas do processo de DNA *shuffling* e desempenham papel fundamental nessas etapas.

1 Capítulo

Biologia Molecular – Conceitos e Processos Básicos

"Uma longa viagem começa com um único passo."
Lao Tse (604 a.C.)

1.1 Introdução

No que segue são apresentados os principais conceitos de Biologia Celular e Molecular necessários para a contextualização do trabalho realizado e compreensão dos modelos, processos e algoritmos apresentados e discutidos ao longo desta tese. O objetivo deste capítulo é também o de estabelecer e padronizar a nomenclatura, ser um referencial aos conceitos fundamentais utilizados e tornar o trabalho auto-contido. O conteúdo deste capítulo foi compilado de várias referências, entre elas: (ALBERTS et al., 2002), (SNUSTAD; SIMMONS, 2001), (LODISH et al., 2001), (BROWN, 1999) e (WATSON et al., 1992).

As células são as unidades formadoras dos seres vivos. São estruturas delimitadas por membranas cujo conteúdo interno é formado por uma variedade de substâncias químicas em solução aquosa que interagem para garantir a ocorrência de funções básicas necessárias à vida, como por exemplo, aquisição e uso de energia, realização de diversas reações químicas, manutenção do ambiente interno constante e capacidade de reprodução.

Existem dois tipos principais de células, as procariontes e as eucariontes, que diferem entre si pelas estruturas internas que as compõem. As células procariontes são consideradas mais simples e são constituídas basicamente por parede celular, membrana plasmática e citoplasma. No citoplasma estão o material genético (DNA ou RNA), as proteínas e outras pequenas moléculas. Já as células eucariontes são mais complexas em termos de estrutura e funções. A principal diferença entre esses dois tipos de células é a presença de um núcleo nas células eucariontes, o qual armazena o material genético.

1.2 DNA e RNA

As informações genéticas de todos os organismos vivos estão armazenadas em suas células em macromoléculas chamadas ácidos nucleicos. Os ácidos nucleicos estão presentes em todos os organismos vivos em duas formas distintas: ácido desoxirribonucleico (DNA) e ácido ribonucleico (RNA). As moléculas de DNA e RNA são compostas por unidades menores, chamadas nucleotídeos, ligados entre si. Os nucleotídeos são formados por três unidades básicas:

- uma molécula chamada base nitrogenada;
- um açúcar; e
- um grupo fosfato.

As bases nitrogenadas encontradas em moléculas de DNA são quatro: Adenina (A), Timina (T), Citosina (C) e Guanina (G), enquanto que em moléculas de RNA as bases nitrogenadas são: Adenina (A), Citosina (C), Guanina (G) e Uracila (U). Além de diferenciar em relação à presença das bases T e U, as moléculas de RNA e DNA diferem em basicamente dois aspectos:

- uma molécula de RNA é composta por uma única fita de nucleotídeos, enquanto que uma molécula de DNA é composta por 2 fitas; e
- o açúcar presente no RNA é a ribose, enquanto que o açúcar presente no DNA é a desoxirribose.

As bases nitrogenadas são formadas por anéis contendo nitrogênio (N) e carbono (C). As estruturas das cinco bases nitrogenadas são apresentadas na Figura 1.1.

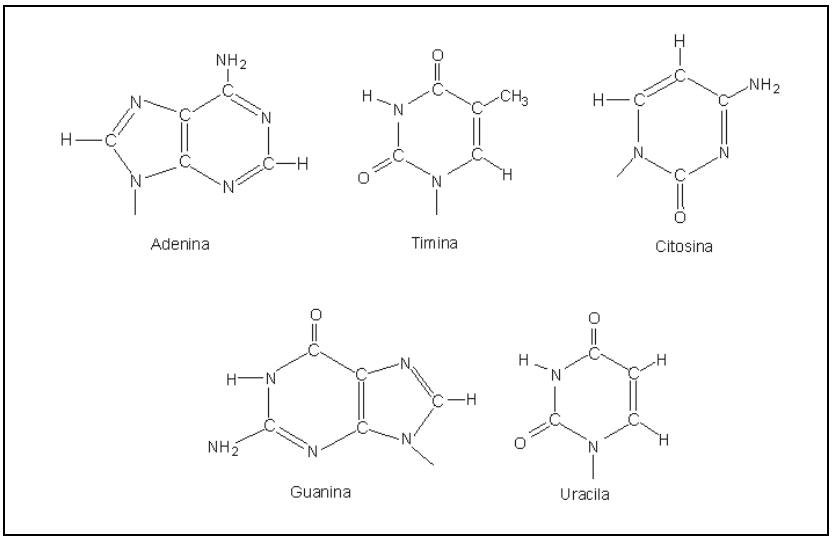


Figura 1.1. Estrutura das quatro bases nitrogenadas Adenina, Timina, Citosina e Guanina presentes na molécula de DNA e da base Uracila que substitui a Timina na molécula de RNA.

O açúcar presente na molécula de DNA (deoxirribose) é composto por uma cadeia de cinco átomos de carbono e, por isso, é chamado de pentose. Os cinco átomos são referenciados por C_1, C_2, C_3, C_4 e C_5 e se dispõem como mostrado na Figura 1.2.

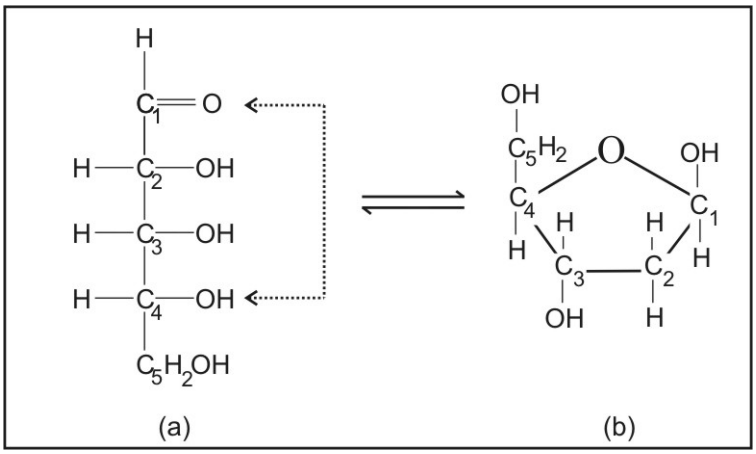


Figura 1.2. Pentose presente na molécula de DNA. (a) Visualização linear. (b) Visualização estrutural.

O grupo fosfato (P) presente no nucleotídeo encontra-se ligado ao carbono C_5. A representação de um nucleotídeo é mostrada na Figura 1.3.

Diversos nucleotídeos se unem para formar uma molécula de DNA (ou RNA) por meio de ligações fosfodiésteres. Essas ligações ocorrem entre o grupo OH do carbono C_3 da pentose de um nucleotídeo e o grupo fosfato, ligado ao carbono C_5 do nucleotídeo seguinte e é catalisada pela enzima DNA Polimerase, que participa da síntese da molécula de DNA. A Figura 1.4 apresenta dois nucleotídeos ligados por ligação fosfodiéster.

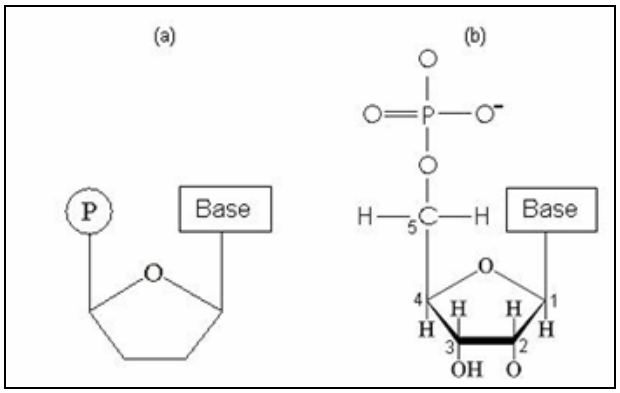


Figura 1.3. Um nucleotídeo, o qual é composto por um grupo fosfato, um açúcar e uma base nitrogenada. (a) Representação simplificada. (b) Representação não simplificada.

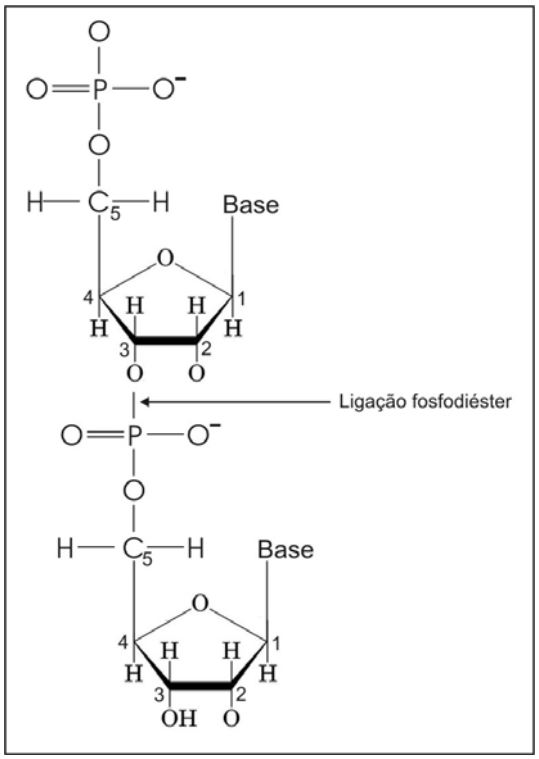


Figura 1.4. Ligação fosfodiéster que une dois nucleotídeos.

O modelo da estrutura do DNA proposto por Crick e Watson (1953), estabelece que a molécula de DNA consiste de duas cadeias (fitas) de nucleotídeos que interagem e formam uma estrutura descrita como dupla hélice. A estrutura de dupla hélice é mantida principalmente devido a dois tipos de ligações existentes entre os nucleotídeos: as ligações fosfodiésteres, que ocorrem entre nucleotídeos de uma mesma cadeia, e as pontes de hidrogênio. As pontes de hidrogênio são ligações formadas entre pares específicos de bases, sendo que cada uma das bases deste par pertence a uma das fitas da molécula de DNA. A base Timina liga-se com a base Adenina por meio da formação de duas pontes de hidrogênio, e a Citosina liga-se com a Guanina por três

pontes de hidrogênio. Desta forma, os pares de bases A–T e C–G são ditos complementares e, adicionalmente, podem ser representados por A=T e C≡G, respectivamente. A Figura 1.5 mostra as pontes de hidrogênio entre os pares de bases (pb).

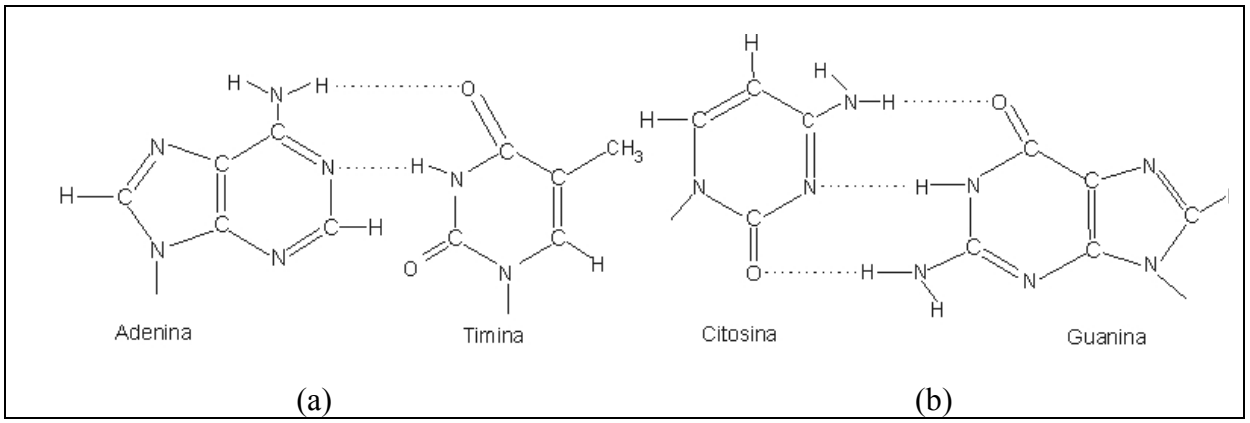


Figura 1.5. Pontes de hidrogênio entre os pares de bases (a) A–T e (b) C–G.

É importante notar que, devido à forma como as cadeias de nucleotídeos se combinam, cada uma delas terá em uma de suas extremidades um carbono C_5 livre[1] e, na outra, um carbono C_3 livre. As extremidades livres de uma cadeia determinam a orientação da cadeia. Por convenção, a cadeia de nucleotídeos é representada segundo a orientação dita 5'→ 3', ou seja, os nucleotídeos de uma cadeia são descritos iniciando pelo nucleotídeo que possui o carbono 5' livre e finalizando no nucleotídeo que possui o carbono 3' livre. A cadeia de nucleotídeos com orientação 5'→ 3' é também chamada de fita + da molécula de DNA; a fita complementar, de orientação 3'→ 5', é dita fita – . As fitas + e – são ditas antiparalelas. A Figura 1.6 representa um fragmento de molécula de DNA.

Como comentado anteriormente, o que diferencia um nucleotídeo de outro é o tipo de base nitrogenada presente na molécula. Desta forma, uma cadeia de nucleotídeos pode ser representada apenas pelas suas bases. Adicionalmente, uma molécula de DNA pode ser representada apenas pelas bases que compõem uma de suas fitas, uma vez que as fitas são complementares. Assim, a molécula de DNA apresentada na Figura 1.6 pode ser representada simplesmente por: 5' AGCTA 3'.

No ciclo de vida de uma célula, antes que a célula se divida (reproduza), ela deve duplicar o seu material genético. O mecanismo de duplicação do DNA, também chamado de replicação ou cópia, tem como princípio a complementaridade entre os pares de bases que compõem a dupla

[1] Neste contexto, diz-se que um nucleotídeo tem um carbono livre se ele pode vir a se ligar com outro nucleotídeo.

fita dessa molécula. Nesse processo, cada uma das fitas serve como molde para a síntese de uma nova cadeia de nucleotídeos, complementar à fita molde.

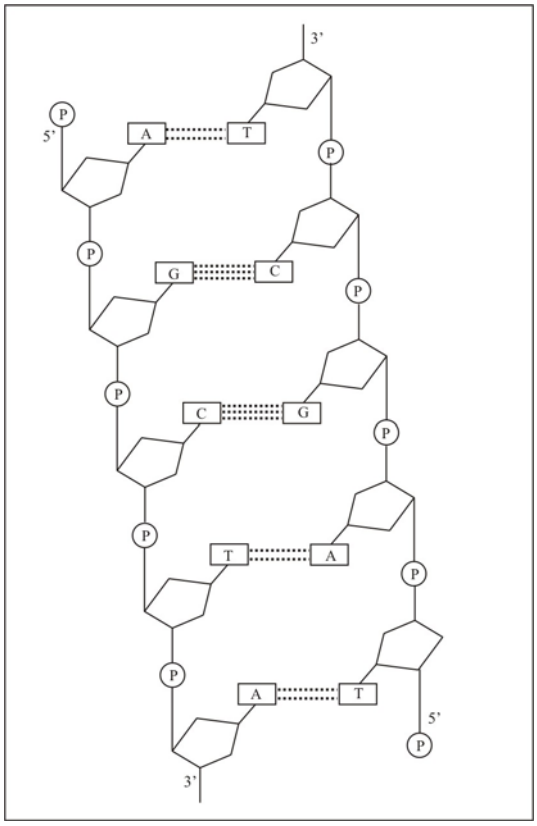


Figura 1.6. Representação de um fragmento de molécula de DNA.

Para que a duplicação seja possível, as fitas que formam a dupla hélice da molécula de DNA devem se separar, ao menos temporariamente, para que possam ser utilizadas como molde para a síntese da nova fita. A síntese de uma nova fita de DNA a partir de um molde é viabilizada por enzimas do tipo DNA Polimerase, as quais catalisam a adição de nucleotídeos à cadeia sendo sintetizada. A polimerase só é capaz de ligar um novo nucleotídeo ao carbono C_3 de outro nucleotídeo e, por isso, diz-se que a síntese só acontece no sentido 5'→ 3'. A Figura 1.7 mostra de maneira simplificada como ocorre a duplicação de uma molécula de DNA.

Como pode ser observado na Figura 1.7, cada uma das duas moléculas de DNA resultantes da duplicação é composta por uma fita "velha" (molde) e uma fita recém sintetizada. Por esse motivo, o processo de duplicação é chamado semiconservativo. A replicação ocorre apenas no sentido 5'→ 3' em cada uma das fitas da molécula; ele se inicia em pontos específicos de cada uma das fitas chamados de origens de replicação (ori). Quando o processo de duplicação termina,

duas moléculas de DNA, idênticas entre si (e idênticas à molécula de DNA original) foram produzidas.

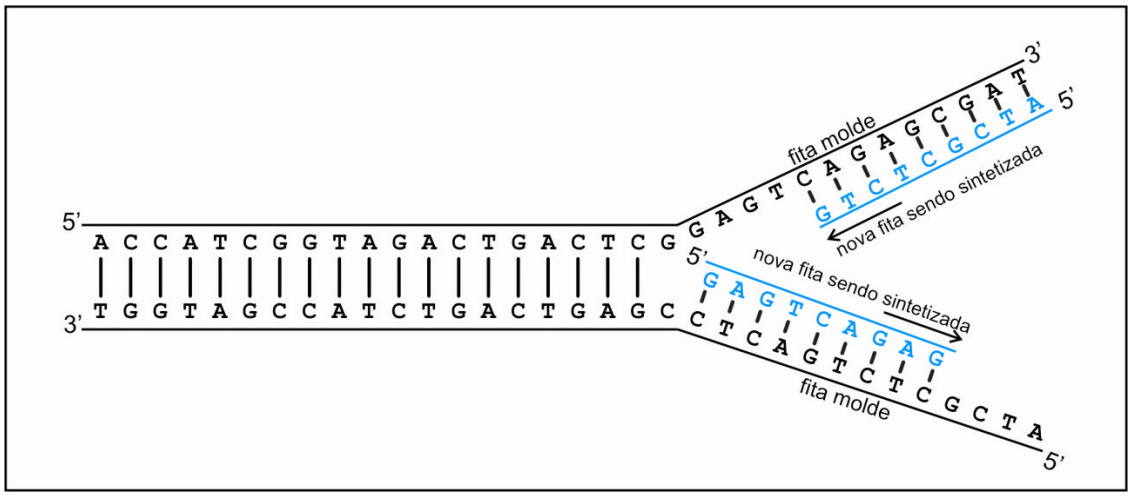


Figura 1.7. Representação simplificada do mecanismo de duplicação do DNA.

1.3 Genes, proteínas e síntese de proteínas

As informações genéticas de um indivíduo encontram-se organizadas ao longo da molécula de DNA em unidades chamadas genes. Os genes são as unidades responsáveis pela produção de um grupo especial de macromoléculas, as proteínas, que estão envolvidas em diversas funções vitais tais como metabolismo, função estrutural das células, catálise de reações químicas (as enzimas), mecanismo de contração muscular, sistema imunológico, entre outros.

A síntese de proteínas a partir dos genes envolve também a participação do ácido ribonucleico, ou RNA. O mecanismo que rege a síntese de proteínas é conhecido como Dogma Central da Biologia Celular e estabelece que o fluxo de informação genética é "DNA para RNA para proteína", ou, de forma simplista "DNA faz RNA, que faz proteína que, por sua vez, facilita os dois passos prévios bem como a replicação do DNA". Com poucas exceções, todas as células biológicas obedecem essa regra. A relação entre DNA, RNA e proteínas, estabelecida pelo Dogma Central da Biologia Celular é sumarizada no diagrama da Figura 1.8.

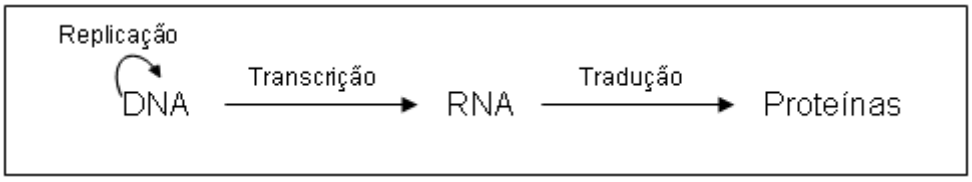


Figura 1.8. Dogma Central da Biologia Celular.

Como identificado na Figura 1.8, a síntese de proteínas envolve dois processos distintos: Transcrição e Tradução. As enzimas que atuam no processo de transcrição do DNA em RNA são chamadas de RNA polimerases. Essas enzimas são capazes de identificar a posição de um gene dentro da molécula de DNA e iniciar sua transcrição a partir do ponto identificado. A síntese da molécula de RNA também segue o princípio da complementaridade entre os pares de bases; porém é a base Uracila, e não a Timina, que forma par com a base Adenina neste tipo de molécula. A Figura 1.9 exemplifica como ocorre a transcrição de DNA em RNA por meio da ação do complexo de enzimas RNA polimerases.

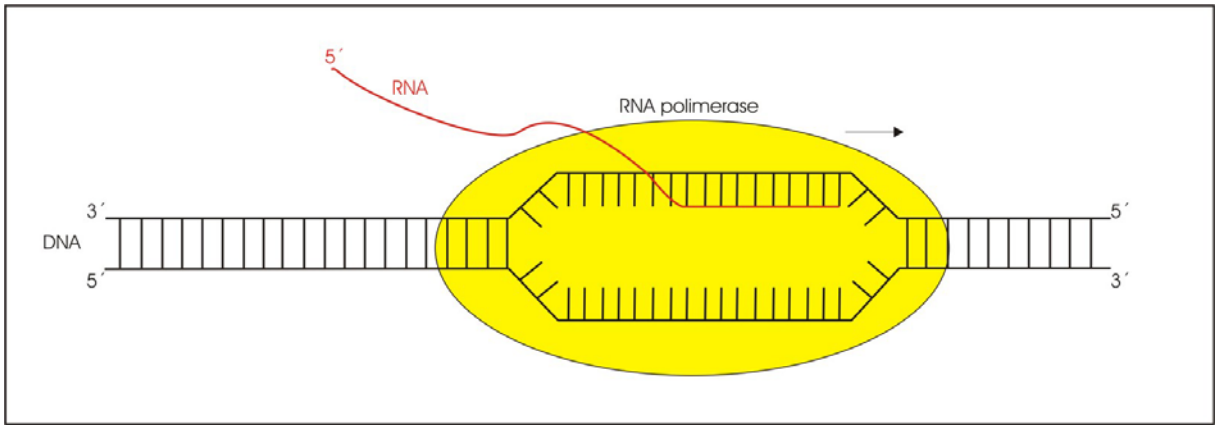


Figura 1.9. Transcrição de DNA em RNA.

Uma vez sintetizada a molécula de RNA, ela deve ser 'traduzida' para que uma proteína seja produzida. O processo de tradução segue o código genético o qual estabelece uma relação entre triplas de bases consecutivas, chamadas códons, e outro tipo de molécula chamada aminoácido (ver Tabela 1.1). Os aminoácidos são, desta forma, as unidades básicas constituintes de uma proteína. Considerando que para cada uma das três unidades de um códon, existem quatro possíveis bases distintas (A, U, C e G), espera-se um total de $4^3 = 64$ códons distintos. A tabela do código genético, porém, é composta por vinte aminoácidos distintos e não 64, uma vez que alguns códons distintos codificam para um mesmo aminoácido, como pode ser visto na Tabela 1.2. Existem ainda dois grupos especiais de códons, são eles o *start* códon (AUG) e os *stop* códons (UAA, UAG e UGA), os quais sinalizam o início e o final do gene a ser transcrito, respectivamente. Os nomes e as abreviaturas utilizadas para representar cada um dos vinte aminoácidos, bem como os códons correspondentes, são apresentados na Tabela 1.1. A Tabela 1.2 apresenta o código genético.

Tabela 1.1. Código, nome e códons dos vinte aminoácidos.

Código de uma letra	Código de três letras	Nome do aminoácido	Códons correspondentes
A	Ala	Alanina	GCU, GCC, GCA, GCG
C	Cys	Cisteína	UGU, UGC
D	Asp	Ácido Aspártico	GAU, GAC
E	Glu	Ácido Glutâmico	GAA, GAG
F	Phe	Fenilalanina	UUU, UUC
G	Gly	Glicina	GGU, GGC, GGA, GGG
H	His	Histidina	CAU, CAC
I	Ile	Isoleucina	AUU, AUC, AUA
K	Lys	Lisina	AAA, AAG
L	Leu	Leucina	UUA, UUG, CUU, CUC, CUA, CUG
M	Met	Metionina	AUG
N	Asn	Asparagina	AAU, AAC
P	Pro	Prolina	CCU, CCC, CCA, CCG
Q	Gln	Glutamina	CAA, CAG
R	Arg	Arginina	CGU, CGC, CGA, CGG, AGA, AGG
S	Ser	Serina	UCU, UCC, UCA, UCG, AGU, AGC
T	Thr	Treonina	ACU, ACC, ACA, ACG
V	Val	Valina	GUU, GUC, GUA, GUG
W	Trp	Triptofano	UGG
Y	Tyr	Tirosina	UAU, UAC

O processo de transcrição segue o princípio de complementaridade entre os pares de bases e o processo de tradução segue o código genético. A Figura 1.10 ilustra, de uma maneira simplificada, como ocorre a síntese de uma proteína a partir de uma região do DNA correspondente a um gene.

Tabela 1.2. Código Genético.

1º posição (5')	2º posição				3º posição (3')
	U	C	A	G	
U	UUU — Phe	UCU — Ser	UAU — Tyr	UGU — Cys	U
	UUC — Phe	UCC — Ser	UAC — Tyr	UGC — Cys	C
	UUA — Leu	UCA — Ser	UAA — *stop***	UGA — *stop***	A
	UUG — Leu	UCG — Ser	UAG — *stop***	UGG — Trp	G
C	CUU — Leu	CCU — Pro	CAU — His	CGU — Arg	U
	CUC — Leu	CCC — Pro	CAC — His	CGC — Arg	C
	CUA — Leu	CCA — Pro	CAA — Gln	CGA — Arg	A
	CUG — Leu	CCG — Pro	CAG — Gln	CGG — Arg	G
A	AUU — Ile	ACU — Thr	AAU — Asn	AGU — Ser	U
	AUC — Ile	ACC — Thr	AAC — Asn	AGC — Ser	C
	AUA — Ile	ACA — Thr	AAA — Lys	AGA — Arg	A
	AUG — Met*	ACG — Thr	AAG — Lys	AGG — Arg	G
G	GUU — Val	GCU — Ala	GAU — Asp	GGU — Gly	U
	GUC — Val	GCC — Ala	GAC — Asp	GGC — Gly	C
	GUA — Val	GCA — Ala	GAA — Glu	GGA — Gly	A
	GUG — Val	GCG — Ala	GAG — Glu	GGG — Gly	G

**start* códon
***stop* códon

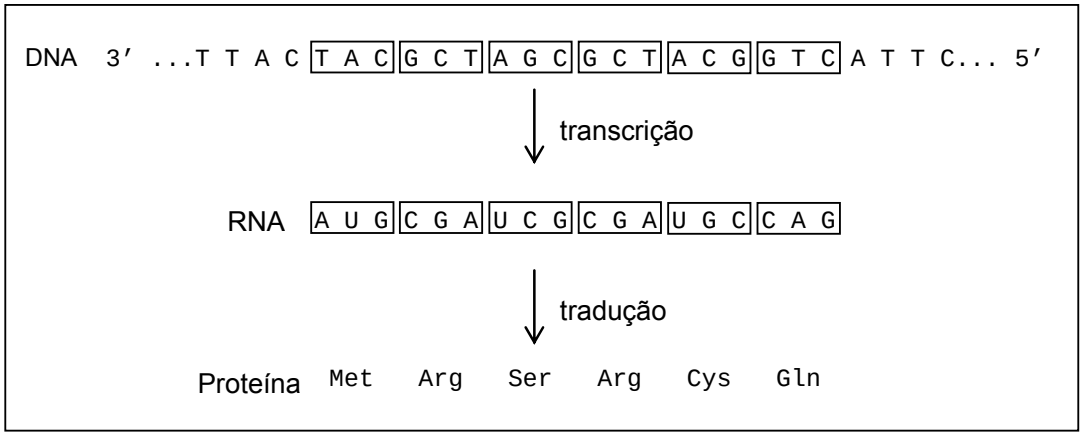


Figura 1.10. Síntese de uma proteína a partir de uma região do DNA correspondente a um gene.

O mecanismo de síntese de proteínas, como ocorre nas células, é um processo bem mais complexo do que o esquema apresentado na Figura 1.10 e envolve outras macromoléculas. De fato, existem três tipos de RNAs envolvidos nesse processo de síntese:

– *RNA mensageiro* (mRNA): é a forma como as informações contidas no gene saem do núcleo da célula (para o caso de células eucariontes) para que a síntese protéica ocorra no citoplasma. Esta molécula é sintetizada a partir do DNA por um mecanismo complexo do qual participam as enzimas do tipo RNA polimerases;

– *RNA transportador* (tRNA): esse tipo de RNA armazena a "chave" para que os códons contidos no mRNA sejam decifrados, ou seja, traduzidos em aminoácidos. Cada tRNA contém um aminoácido e um anticódon específico para este aminoácido. Por exemplo, o tRNA correspondente ao aminoácido serina contém, além desta molécula, o anticódon UCA, o qual é complementar ao códon AGU, códon correspondente ao aminoácido serina (ver Tabela 1.2). No momento da síntese de uma proteína, os anticódons dos tRNAs ligam-se aos códons do mRNA por complementaridade entre as bases e a cadeia polipeptídica (protéica) vai sendo formada. Esta associação entre mRNA e tRNA é estabilizada por moléculas chamadas ribossomos; e
– *RNA ribossômico* (rRNA): os rRNAs associados a um conjunto de proteínas formam os ribossomos. Os ribossomos movem-se ao logo do mRNA catalizando a união dos aminoácidos a fim de produzir a proteína.

A Figura 1.11 mostra como o ribossomo utiliza-se do mRNA e dos tRNAs para que uma proteína seja sintetizada.

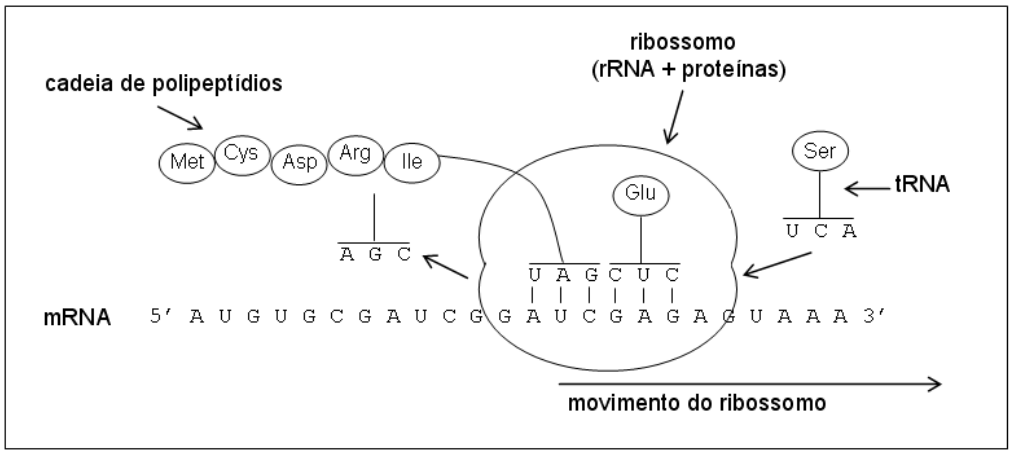


Figura 1.11. Síntese de proteína pelo ribossomo. Os anticódons dos tRNAs ligam-se aos códons do mRNA dentro do ribossomo e a cadeia polipeptídica vai sendo formada pela união dos aminoácidos.

Como visto, uma proteína é composta por uma cadeia de aminoácidos. Todos os vinte aminoácidos são compostos por um carbono alfa (C_α), ligado a quatro diferentes grupos químicos:

– um grupo amino (NH_2);
– um grupo carboxil (COOH);
– um átomo de hidrogênio (H); e

– um grupo variável, chamado de cadeia lateral ou grupo R, que é diferente em cada um dos aminoácidos.

A representação geral de um aminoácido é mostrada na Figura 1.12.

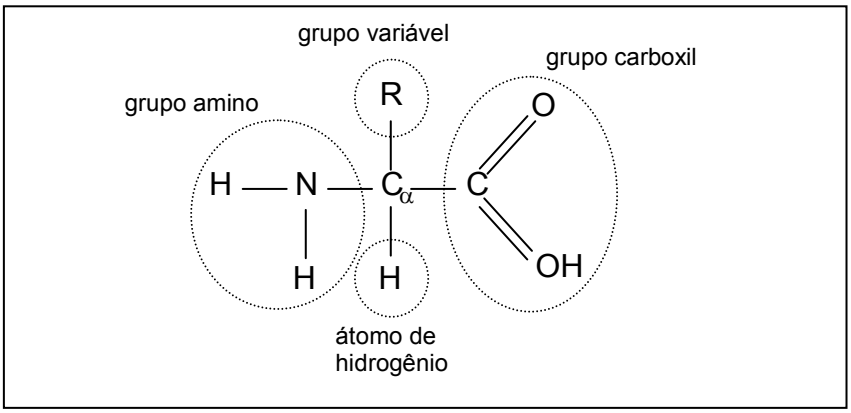


Figura 1.12. Estrutura geral de um aminoácido.

Estruturalmente, as proteínas podem se organizar em até quatro níveis, denominados estruturas primária, secundária, terciária e quaternária. A organização espacial de uma proteína está fortemente relacionada à função que ela desempenha no organismo. A estrutura primária de uma proteína corresponde à seqüência linear dos aminoácidos que constituem sua cadeia polipeptídica. A estrutura secundária refere-se à localização de partes da cadeia polipeptídica que assumem as conformações espaciais de alfa hélice (em forma de espiral) ou folhas beta (estrutura planar). O terceiro nível de organização estrutural se refere ao arranjo tridimensional de todos os aminoácidos da cadeia. A estrutura quaternária ocorre em um grupo especial de proteínas chamadas multiméricas, as quais são formadas pela união de duas ou mais cadeias de polipeptídios, também chamadas de subunidades, e descreve o número e as posições relativas dessas subunidades na proteína.

1.4 Síntese *in vitro* de DNA e reação de PCR

Muitas técnicas utilizadas na análise laboratorial de moléculas de DNA requerem que a seqüência esteja disponível em quantidades significativas para viabilizar a realização de diversos experimentos. Considere, por exemplo, uma seqüência de DNA que codifica para um gene e que essa seqüência deva ser submetida a uma gama de experimentos que permitam a investigação de

sua função e estrutura, entre outros. Para viabilizar tais experimentos, quantidade suficiente dessa seqüência deve estar disponível. A possibilidade de se criar inúmeras cópias de uma molécula de DNA em laboratório otimiza este trabalho, uma vez que a molécula em questão não precisa ser extraída diretamente do organismo ao qual pertence.

O processo conhecido como síntese *in vitro* de DNA se refere à produção, em laboratório, de diversas cópias de uma molécula de DNA original. Em 1957 foi realizada a primeira síntese *in vitro* de DNA, por Arthur Kornberg, Uriel Littauer e sua equipe (LITTAUER; KORNBERG, 1957). A descoberta da enzima DNA polimerase I, isolada da bactéria *Escherichia coli*, viabilizou o desenvolvimento da técnica, uma vez que essa enzima atua na replicação do DNA.

Para que as enzimas do tipo polimerase atuem na replicação ou cópia de moléculas de DNA, é necessário a presença de três componentes básicos: *primers*, moldes e os quatro nucleotídeos. *Primers* são seqüências iniciadoras do processo de síntese e correspondem a curtas cadeias de nucleotídeos complementares à região inicial (*primer forward*) e região final (*primer reverse*) da seqüência que se deseja copiar, dita seqüência alvo. Em uma mesma reação, *primers* e seqüência alvo (a qual deve estar na forma desnaturada, ou seja, fita simples) se unem por complementaridade entre suas bases. Uma vez formado o complexo – *primer* e seqüência alvo – as enzimas do tipo polimerase ligam-se ao grupo OH da extremidade do carbono C_3 do *primer* (extremidade 3' OH) para dar início ao processo de síntese. Para que a nova molécula seja sintetizada, a DNA polimerase utiliza a seqüência alvo como molde e, segundo o princípio da complementariedade entre as bases, adiciona novos nucleotídeos aos *primers*, sendo estes nucleotídeos complementares aos nucleotídeos da fita molde. Este processo recebe o nome de extensão. Ao final da extensão, uma nova molécula, complementar à seqüência molde, é produzida. Este tipo de reação é denominado Reação de Extensão por Polimerase ou simplesmente PCR (*Polymerase Chain Reaction*). Devido à necessidade do grupo 3' OH livre na seqüência do *primer* para que a enzima DNA polimerase se acople, a síntese de uma nova molécula só ocorre no sentido 5'→ 3'. Caso o *primer* pareie[2] com a seqüência de DNA molde de forma que o sentido de extensão seja 3'→ 5', a polimerase não atua e, portanto, a extensão não ocorre. O esquema de atuação de enzimas do tipo polimerase é representado na Figura 1.13.

[2] O termo *annealing* é também utilizado para denominar o pareamento ou a união de fragmentos de DNA por complementaridade entre suas bases.

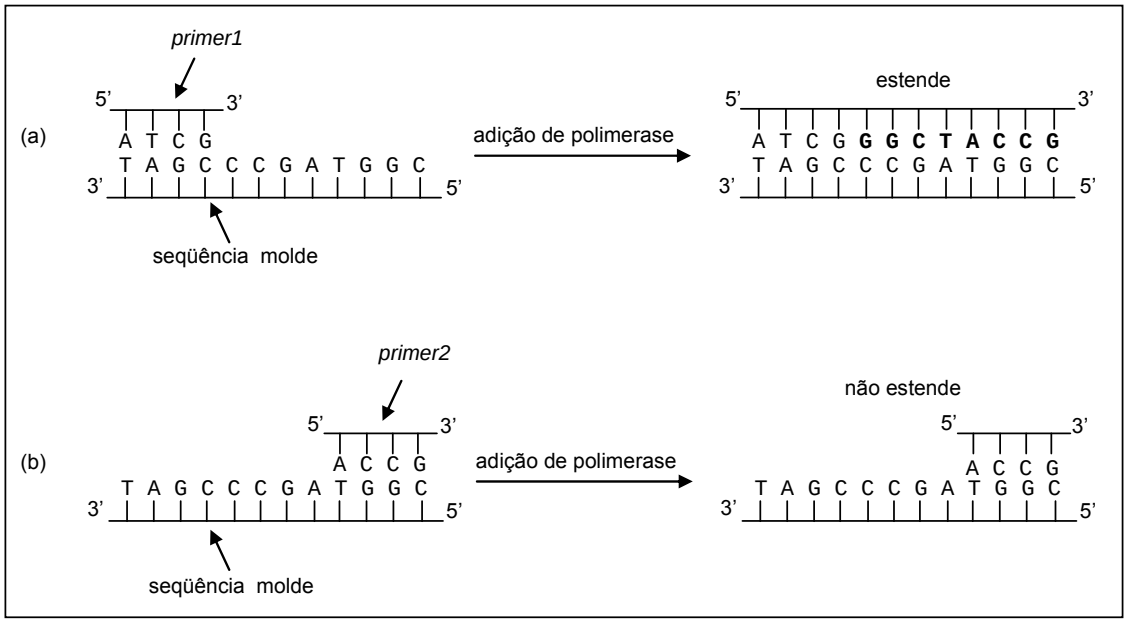


Figura 1.13. Atuação da enzima DNA polimerase na replicação do DNA. (a) A polimerase é capaz de estender o *primer1* criando uma fita complementar ao molde. (b) A polimerase não é capaz de estender o *primer2* – uma extensão não ocorre a partir da extremidade 5'.

O termo amplificação é utilizado para caracterizar experimentos com vistas ao aumento (em número) de uma molécula. A amplificação de uma molécula de DNA em laboratório ocorre por meio de diversos ciclos de extensão por polimerase. No primeiro ciclo, a molécula de DNA alvo é desnaturada[3] resultando em duas seqüências molde, as quais, após se parearem com os *primers*, são estendidas pela polimerase e resultam em duas novas moléculas de DNA idênticas à molécula original. No segundo ciclo, as duas moléculas resultantes do ciclo anterior são desnaturadas, dando origem a quatro moldes que, após o pareamento com os *primers* e a extensão, dão origem a quatro novas moléculas de DNA idênticas à molécula original. Assim, após um número n de ciclos de PCR (desnaturação, pareamento e extensão) tem-se, idealmente, 2^n moléculas de DNA idênticas à molécula original. A Figura 1.14 ilustra as três etapas do ciclo de PCR.

Diversas variáveis e parâmetros, tais como o número de ciclos, a temperatura e o tempo de duração de cada ciclo, bem como a qualidade e a quantidade de *primers* utilizados podem interferir na reação de PCR e, conseqüentemente, o número 2^n de seqüências de DNA resultantes pode não ser obtido após a execução de n ciclos de PCR. A reação de PCR volta a ser abordada em

[3] O processo de desnaturação de uma molécula de DNA faz com que a fita dupla desta molécula se separe em duas fitas simples. A desnaturação de uma molécula é obtida pelo aquecimento da molécula a temperaturas próximas de 94ºC.

maiores detalhes no Capítulo 2, como parte do processo de DNA *shuffling*, um dos focos desta pesquisa.

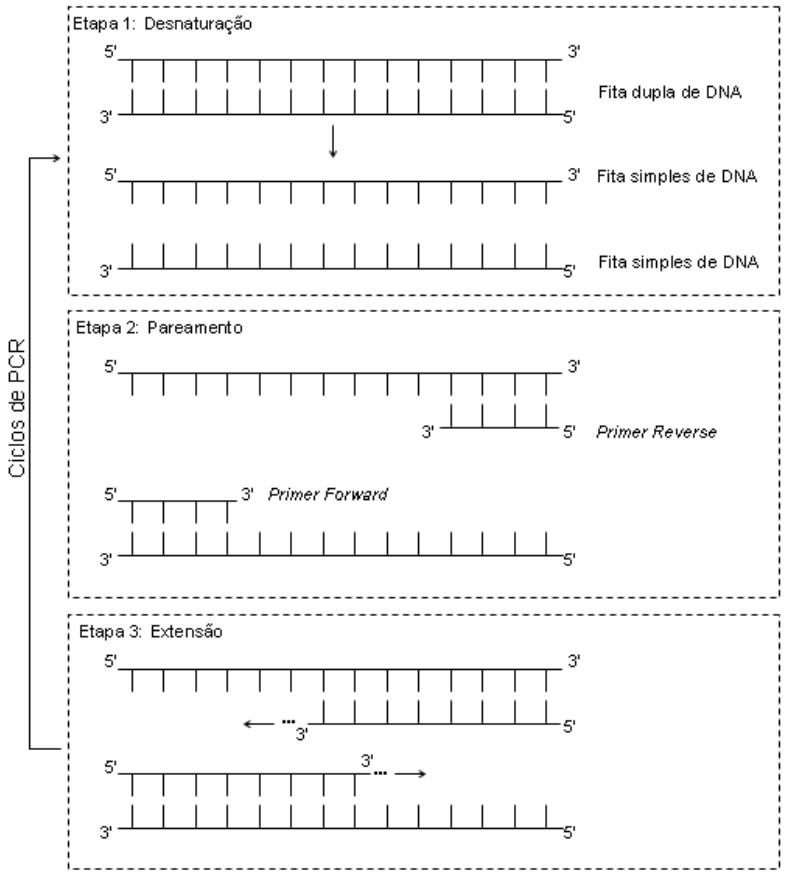


Figura 1.14. Ciclos de PCR com *primer*s. A cada ciclo, uma molécula de DNA dá origem a duas novas moléculas idênticas à molécula original.

Outro fator importante a ser considerado no processo de síntese *in vitro* de DNA são as mutações que uma reação de PCR pode introduzir nas seqüências produzidas. Nucleotídeos não complementares ao molde podem ser inseridos durante a síntese da nova fita de DNA, e estes erros serão propagados nos ciclos seguintes da reação, resultando em seqüências distintas das originais. Os erros ou mutações ocorridos durante a síntese de DNA são também chamados substituições. Estudos realizados *in vitro* com a enzima *Taq* polimerase revelam que esses erros são introduzidos a uma taxa aproximada de 1 em cada 2×10^4 nucleotídeos inseridos. Contudo, a fidelidade da síntese de DNA pela utilização da *Taq* polimerase pode variar significantemente com mudanças na concentração de Mg^{2+} e variações de pH da solução utilizada como *buffer* da reação, entre outros (METZKER; CASKEY, 2001).

Se por um lado a ocorrência de substituições de nucleotídeos durante a amplificação de uma seqüência alvo é indesejada, em experimentos de Evolução Molecular Direta tais substituições são, muitas vezes, o principal objetivo destes experimentos, como discutido no Capítulo 2, que apresenta um conjunto de técnicas de Evolução Molecular Direta, abordando detalhes dos processos envolvidos.

1.5 Clonagem e tecnologia do DNA recombinante

A clonagem de uma molécula de DNA consiste em seu isolamento e sua propagação (multiplicação) em moléculas idênticas em um organismo vivo. O procedimento de clonagem envolve basicamente três etapas essenciais:

- Isolamento do fragmento de DNA de interesse (chamado de inserto ou DNA alvo);
- Incorporação do fragmento isolado em um fragmento genético auto-replicante (por exemplo, um virus ou um plasmídeo), denominado vetor de clonagem. Como resultado, tem-se uma molécula de DNA recombinante[4].
- Incorporação da molécula recombinante (vetor + inserto) em uma célula hospedeira para que ocorra sua amplificação. Cada célula hospedeira, ao se reproduzir, resulta em um conjunto de células idênticas, denominado colônia. Caso a célula hospedeira, que deu origem a uma colônia, tenha recebido a molécula de DNA recombinante, cada uma das células desta colônia carrega várias cópias da molécula recombinante. A coleção de todas as colônias é chamada de Biblioteca. Durante este processo, a molécula recombinante também se duplica dentro da célula hospedeira, aumentando ainda mais o número de cópias do DNA alvo produzidas.

O isolamento e a incorporação do fragmento de interesse em um vetor de clonagem são viabilizados, entre outros, pela existência de uma classe específica de enzimas chamadas endonucleases. Essas enzimas fragmentam moléculas de DNA, realizando cortes internos ao longo da molécula. Esses cortes podem ocorrer em posições aleatórias ou específicas ao longo da seqüência de DNA, dependendo do tipo de endonuclease utilizada.

[4] O avanço das técnicas envolvidas na criação de uma molécula de DNA recombinante influenciou diretamente na popularização das técnicas de clonagem.

As endonucleases que cortam o DNA em posições específicas são chamadas de sítio-específicas, uma vez que localizam seqüências de nucleotídeos na molécula de DNA, os chamados sítios de restrição, e realizam o corte (ou clivagem) da molécula apenas nesses pontos. A maioria das endonucleases produz cortes desencontrados nas duas fitas complementares da molécula de DNA, ou seja, os cortes acontecem em posições distintas em cada uma das fitas. Na maioria dos casos, as seqüências de reconhecimento das enzimas de restrição são palíndromas, isto é, são seqüências de pares de nucleotídeos que têm a mesma leitura em ambos os sentidos, partindo-se de um ponto central da mesma, ou ponto de simetria. A Tabela 1.3 apresenta as seqüências de reconhecimento de algumas endonucleases de restrição, indicando os respectivos sítios de clivagem e pontos de simetria. Uma vasta coleção de informações sobre enzimas de restrição e proteínas relacionadas pode ser encontrada na base de dados REBASE (*The Restriction Enzyme Database*[5] (ROBERTS et al*.,* 2007)).

Além das enzimas de restrição, ou sítio-específicas, existe também outra classe de enzimas, como por exemplo, a enzima DNase I, que produz cortes em posições aleatórias em cada uma das fitas de uma molécula de DNA, ou seja, são enzimas não sítio-específicas. Este tipo de enzima, bem como as sítio-específicas, são muito utilizadas em experimentos de Evolução Molecular Direta, como será visto no Capítulo 2.

[5] http://rebase.neb.com

Tabela 1.3. Seqüências de reconhecimento de algumas endonucleases de restrição, com indicação dos respectivos sítios de clivagem (→) e pontos de simetria (•).

Enzima	Fonte	Seqüência de reconhecimento e sítio de clivagem	Fragmentos resultantes da clivagem
EcoRI	*Escherichia coli* linhagem RY13	5′...GAATTC...3′ 3′...CTTAAG...5′	5′...G AATTC...3′ 3′...CTTAA G...5′
HindIII	*Haemophilus influenzae* linhagem Rd	5′...AAGCTT...3′ 3′...TTCGAA...5′	5′...A AGCTT...3′ 3′...TTCGA A...5′
PstI	*Providencia stuartii*	5´...CTGCAG...3´ 3´...GACGTC...5´	5´...CTGCA G...3´ 3´...G ACGTC...5´
AluI	*Arthobacter lutes*	5´...AGCT...3´ 3´...TCGA...5´	5´...AG CT...3´ 3´...TC GA...5´
TaqI	*Thermus aquaticus*	5´...TCGA...3´ 3´...AGCT...5´	5´...T CGA...3´ 3´...AGC T...5´

O corte desencontrado produzido por algumas enzimas de restrição, somado ao fato de suas seqüências de reconhecimento serem palíndromas, são características muito importantes que permitem que dois ou mais fragmentos de DNA digeridos (fragmentados) com uma mesma enzima de restrição possam ser combinados (remontados) em uma mesma molécula. A Figura 1.15, adaptada de Snustad e Simmons (2001), mostra esquematicamente como dois fragmentos de DNA de origens distintas podem ser remontados em um único fragmento. Observe que após o pareamento entre as bases complementares (e, conseqüentemente, da formação de ligações de hidrogênio entre elas) resta ainda uma última ligação fosfodiéster a ser formada entre as moléculas. A formação de tal ligação é catalizada pela ação de enzimas do tipo DNA ligase.

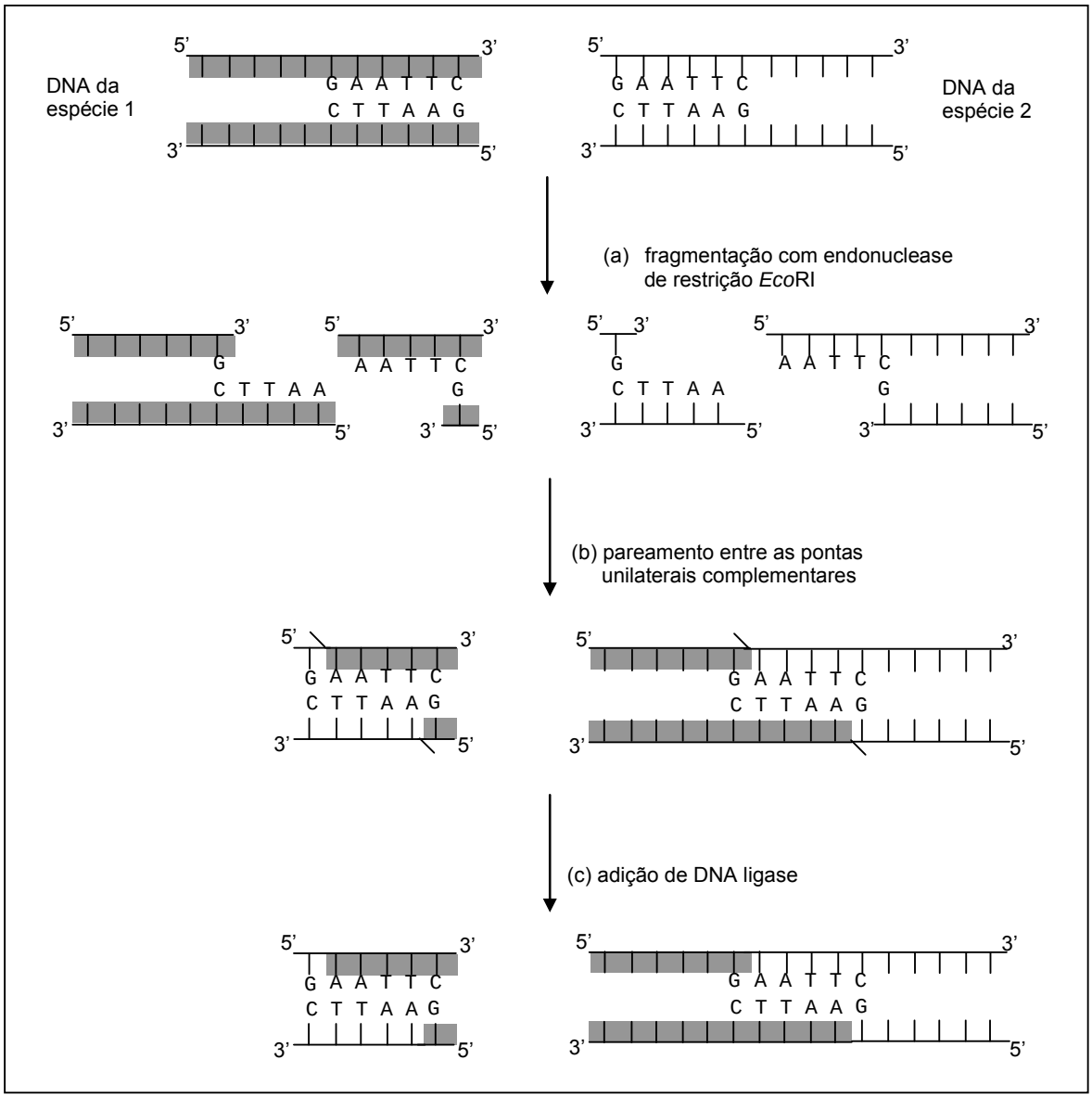


Figura 1.15. Esquema da construção *in vitro* de moléculas de DNA recombinante. (a) As seqüências são fragmentadas por uma mesma enzima de restrição. (b) Os fragmentos resultantes são incubados sob condições de pareamento entre fragmentos complementares. (c) A enzima DNA ligase é adicionada para garantir a formação de ligações fosfodiésteres entre os fragmentos remontados.

Existem diversos tipos de vetores que podem ser utilizados em experimentos de clonagem, cada um deles com características específicas e mais adequados a determinado tipo de experimento dependendo, entre outros, do tamanho do fragmento de DNA que se deseja ligar ao vetor e do tipo de célula que será utilizada como hospedeira para a multiplicação deste fragmento. Dentre os vetores comumente utilizados estão os plasmídeos, fagos, cosmídeos, bacmídeos e YAC's (*Yeast Artificial Chromosome*). Vetores de clonagem típicos possuem as seguintes características:

– capacidade de auto-replicação;

– gene(s) de resistência a determinada(s) droga(s) como, por exemplo, o gene que confere a resistência à ampicilina ou à tetraciclina. São esses genes que irão permitir a identificação das céculas que contém ou não o fragmento de DNA sendo clonado. Estes genes são chamados marcadores;
– sítios de reconhecimento únicos para enzimas de restrição, o que irá permitir que o DNA de interesse seja nele inserido.

As características específicas de um vetor de clonagem são representadas por esquemas denominados mapas de restrição. Um exemplo de vetor de clonagem comumente utilizado é o plasmídeo pUC8. Este vetor contém 2.700 pares de bases (2,7kb) e sítios de reconhecimento únicos para uma variedades de enzimas de restrição. Um esquema de seu mapa de restrição é mostrado na Figura 1.16 (adaptada de (BROWN, 1999)). O plasmídeo pUC8 possui dois marcadores, o gene que confere resistência à ampicilina, o qual é utilizado para a determinação das células que contêm o vetor e o gene lacZ´ que codifica para parte da enzima β-galactosidade. Note que, nos vetores que recebem o inserto, o gene lacZ´ se torna inativo, uma vez que o inserto será inserido na região correspondente a este gene.

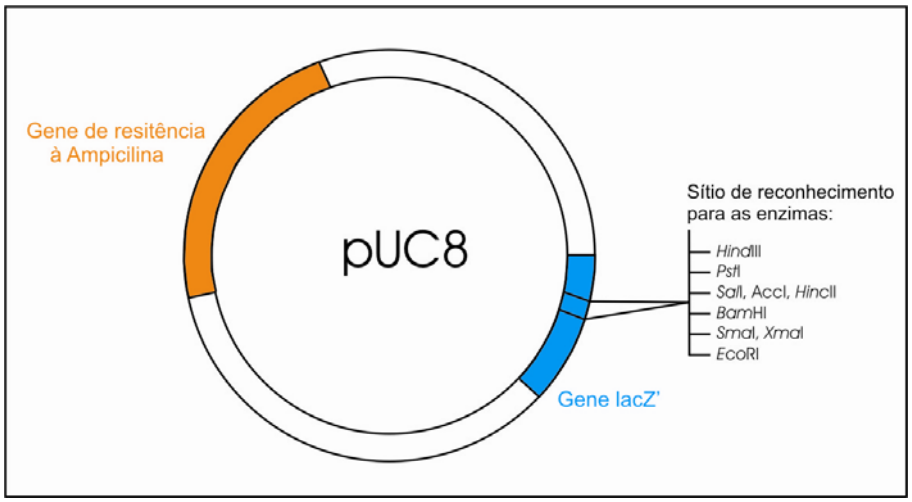


Figura 1.16. Mapa de restrição do vetor pUC8.

Uma vez que o fragmento de DNA a ser clonado foi inserido no vetor de clonagem, este fragmento recombinante deve então ser inserido em uma célula hospedeira para que as replicações, tanto do vetor quanto da célula hospedeira ocorram e, assim, a molécula recombinante seja clonada. A Figura 1.17, adaptada de Lodish et al. (2001), mostra um esquema geral do procedimento de clonagem de um fragmento de DNA.

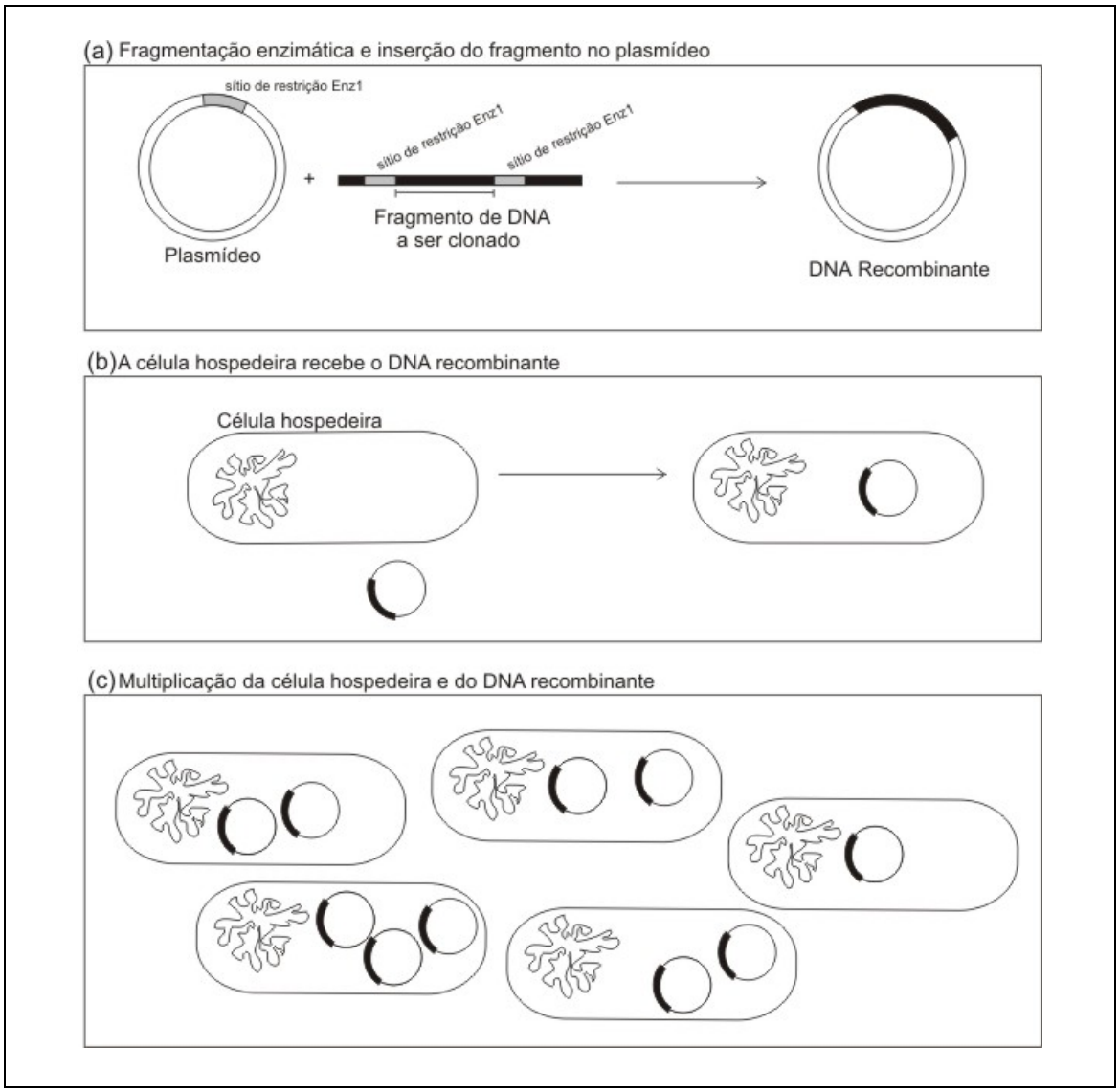


Figura 1.17. Esquema do procedimento geral para a clonagem de um fragmento de DNA utilizando um plasmídeo como vetor. (a) Fragmentação do plasmídeo e da molécula de DNA por uma mesma enzima de restrição (Enz1) e inserção do fragmento a ser clonado no plasmídeo, resultando em uma molécula de DNA recombinante. (b) Inserção da molécula de DNA recombinate na célula hospedeira. (c) Crescimento (multiplicação) da célula hospedeira, bem como da molécula de DNA recombinante.

1.6 Considerações finais

O entendimento e a familiarização com os conceitos básicos de Biologia Molecular apresentados neste capítulo são de fundamental importância para a leitura e compreensão dos demais capítulos desta Tese. A justificativa para que alguns tópicos terem sido abordados de uma maneira simples e superficial, como o caso do tópico sobre genes, proteínas e síntese de proteínas, apresentados na Seção 1.4, é que a descrição foi feita focando os aspectos mais relevantes necessários à

compreensão do trabalho de pesquisa realizado e deve servir como uma leitura inicial sobre o assunto.

2 Capítulo

Técnicas de Evolução Molecular Direta

"Aquilo que sabemos, saber que sabemos; aquilo que não sabemos, saber que não sabemos; isto que é, verdadeiramente, saber." Confúcio (551-479 a.C)

2.1 Introdução

Por mutagênese entende-se a produção de uma mudança na seqüência de DNA. Este capítulo apresenta os principais conceitos e elementos envolvidos na mutagênese, bem como uma revisão bibliográfica das principais técnicas de mutagênese e recombinação existentes, eventos que desempenham um papel fundamental em experimentos de Evolução Molecular Direta. O objetivo deste capítulo é fornecer um panorama de uma subárea específica da Biologia Molecular, i.e., Evolução Molecular Direta, ou Evolução *in vitro*, dentro do qual este trabalho se insere, com foco em uma técnica particular conhecida como DNA *shuffling*.

A evolução molecular dos organismos vivos é um processo lento que ocorre ao longo do tempo e diz respeito às mutações ou alterações sofridas por um determinado organismo em seu material genético, ou seja, em seu DNA. Essas mutações ocorrem na forma de remoções, inserções, substituições ou recombinações de nucleotídeos ao logo da cadeia de DNA. A Evolução Molecular Direta, por sua vez, é um processo laboratorial utilizado para melhorar funcionalidades de seqüências biológicas específicas por meio de diversificação gênica e seleção, num processo que imita a evolução natural das espécies (MAHESHRI; SCHAFFER, 2003).

2.2 Evolução *in vitro* e o processo de mutagêneses

Diversas técnicas de evolução *in vitro* têm sido propostas e utilizadas nos últimos anos na tentativa de melhorar determinadas propriedades, tais como atividade, estabilidade e especificidade de uma variedade de produtos comercialmente importantes tais como proteínas de interesse farmacêutico, vacinas e enzimas ((PATNAIK et al., 2002), (CHANG et al., 1999), (NESS et al., 1999), (STEMMER 1994a), (STEMMER 1994b)). De uma maneira geral, estas técnicas utilizam seqüências gênicas com propriedades de interesse, que são modificadas em laboratório com o objetivo de gerar novas seqüências quiméricas que, possivelmente, codificam para proteínas híbridas com novas funcionalidades ou funcionalidades melhoradas (STEVENSON; BENKOVIC, 2002).

O princípio que direciona a maioria dos protocolos de evolução *in vitro* é a produção de diversidade molecular por meio de técnicas de mutagênese e recombinação. Muitas destas técnicas têm sido desenvolvidas para introduzir diversidade em seqüências gênicas codificadoras de proteínas, sendo muitas delas baseadas na reação polimerase em cadeia – PCR (*Polymerase Chain Reaction*). Uma coleção de técnicas que implementam o processo de mutagênese e recombinação é apresentada nas próximas seções, com o objetivo de evidenciar tanto o processo e metodologia adotados quanto evidenciar o modelo conceitual subjacente.

O princípio básico dos processos de evolução *in vitro*, tal como descrito em Sun (1999), está representado na Figura 2.1 e pode ser sumarizado como segue. Inicialmente, uma biblioteca de moléculas – DNA, RNA ou proteínas – é construída. Essa biblioteca de moléculas pode ser construída utilizando-se moléculas randômicas de peptídeos ou oligonucleotídeos, ou variantes de uma ou mais moléculas iniciais (parentais) obtidas por técnicas de mutagênese (Pool 1 da Figura 2.1).

A utilização de moléculas randômicas para a construção de uma biblioteca inicial de moléculas pode, entretanto, não ser uma boa escolha uma vez que o potencial de diversidade de uma biblioteca completamente randômica pode ser tão grande que as chances de que essa biblioteca contenha moléculas de DNA com propriedades de interesse podem ser mínimas. Por exemplo, a construção de moléculas de DNA randômicas de tamanho N pode resultar em até 4^N moléculas distintas. Já a construção de bibliotecas utilizando-se processos de mutagênese a partir de alguma(s) molécula(s) inicial, a qual já possui propriedades de interesse, pode ser mais útil.

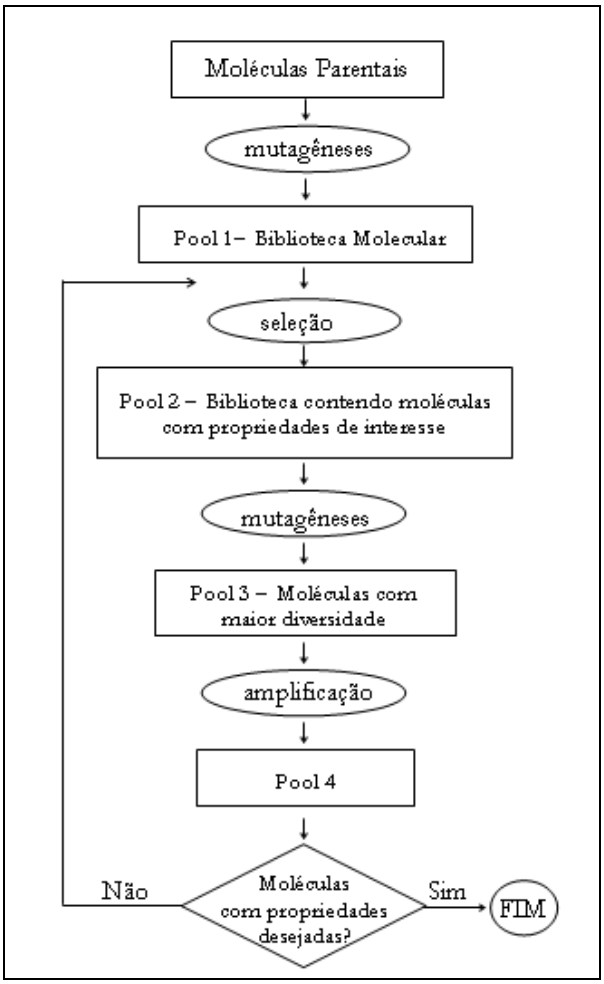


Figura 2.1. Representação esquemática geral do processo de evolução *in vitro*.

Uma vez contruída a biblioteca inicial, é possível selecionar moléculas com características desejadas para dar continuidade ao processo, as quais são agrupadas em uma nova biblioteca (Pool 2 da Figura 2.1). Após a seleção, as moléculas passam novamente pelo processo de mutação para que o número e a diversidade das moléculas aumente ainda mais, sendo criado assim o agrupamento Pool 3 da Figura 2.1. As moléculas são então submetidas a um processo de amplificação para que aumentem em número. Os três processos descritos anteriormente (seleção, mutagênese e amplificação) correspondem a um ciclo do experimento de evolução *in vitro*. Esses ciclos são repetidos até que moléculas com propriedades desejadas sejam obtidas.

O esquema representado na Figura 2.1 é teórico e corresponde a uma generalização da técnica de Evolução Molecular Direta, incluindo fases e processos que, eventualmente, não são contemplados por algumas das técnicas existentes, como discutido na revisão que segue.

A mutagênese é um passo fundamental nos experimentos de evolução *in vitro*. Como pode ser inferido do diagrama apresentado na Figura 2.1, a mutagênese pode ser utilizada tanto para

gerar moléculas iniciais para compor uma biblioteca molecular (formação do Pool 1) quanto para aumentar a diversidade molecular depois do processo de seleção (formação do Pool 3). É de primordial importância que técnicas de evolução *in vitro* implementem a geração de diversidade molecular por meio de mutagênese. Experimentos de evolução *in vitro* que não implementam mutagênese estão restritos apenas à seleção de moléculas que se encontram na biblioteca construída inicialmente. A Figura 2.2 apresenta uma linha cronológica com trabalhos relevantes que propõem/descrevem técnicas de Evolução *in vitro*. As seções de 2.3 a 2.13 descrevem aqueles que, de uma maneira ou outra, forneceram algum subsídio para a pesquisa realizada e discutida nesta tese, e são apresentadas em uma ordem que buscou priorizar o didatismo. Os nomes originais das técnicas/métodos foram mantidos com vistas a torná-los facilmente identificados na literatura.

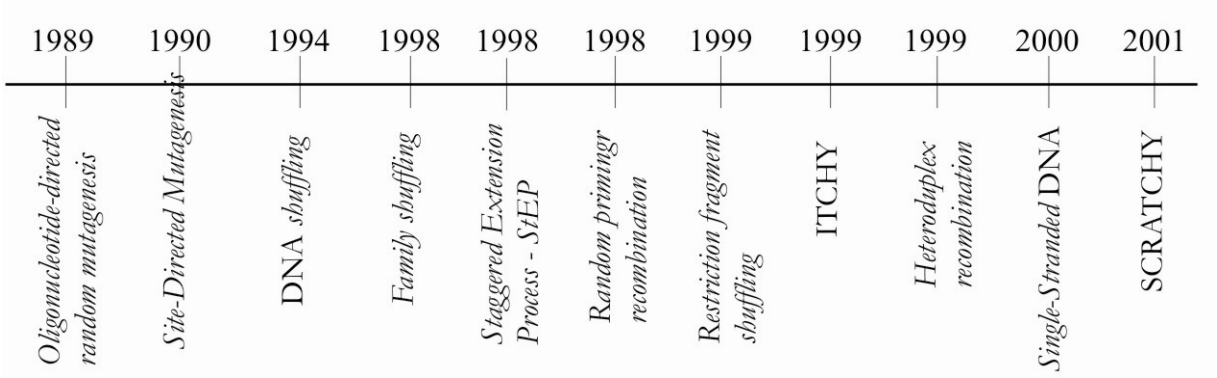


Figura 2.2. Apresentação cronológica de diversas técnicas de evolução *in vitro*.

2.3 *Site-Directed Mutagenesis*

Técnicas de *Site-Directed Mutagenesis* (SDM), ou mutação sítio-dirigida, permitem que nucleotídeos específicos (escolhidos por especialistas humanos) de uma seqüência sejam substituídos. Tais técnicas têm sido empregadas principalmente em estudos que exploram a relação estrutura−função de proteínas e ácidos nucleicos.

O papel estrutural e funcional de resíduos de aminoácidos em uma proteína de interesse pode ser investigado por meio da comparação entre proteínas mutantes que possuem alguns resíduos de aminoácidos mutados e a proteína na sua forma original (*wild-type*) (HERMES et al., 1989). Considere como exemplo o caso da interação enzima-substrato e o problema de identificar os aminoácidos diretamente responsáveis pela ligação entre a enzima e o substrato. A substituição

de um ou mais desses aminoácidos resultaria no não-reconhecimento do substrato pela enzima? A realização de experimentos para responder a essa pergunta requer que a seqüência de nucleotídeos (e aminoácidos) seja conhecida. Além disso, é necessário também que se tenha um conhecimento prévio dos nucleotídeos que estão diretamente relacionados com a função ou atividade em estudo, para poder mutá-los.

Existem diferentes protocolos definidos para SDM. A grande maioria desses protocolos tem por base a reação de PCR (ver Seção 1.4). De uma maneira geral, para se fazer a SDM, *primers* com a mutação desejada são utilizados para a amplificação da seqüência *wild-type*, resultando assim em seqüências com mutações específicas. Segundo Hermes et al. (1989), os métodos ou protocolos utilizados para SDM podem ser classificados de acordo com a localização da mutação desejada, se no início, no meio ou no final do produto da PCR, ou seja, da seqüência resultante.

Para que mutações sejam introduzidas no início ou no final de uma seqüência, a mutação desejada deve ser introduzida no *primer forward* (F) ou no *primer reverse* (R), respectivamente, e uma reação de PCR executada. A Figura 2.3 representa a criação de seqüências cuja mutação é introduzida no início pela utilização de *primer forward* "carregando" a mutação desejada (apenas a fita 3'→ 5' da seqüência a ser amplificada está sendo representada na figura). Note na Figura 2.3 que ocorrem pareamentos entre bases não complementares da seqüência original e do *primer* e são eles os responsáveis pela introdução da mutação desejada nas seqüências resultantes. Este tipo de pareamento é chamado de *mismatch*.

Devido à limitações quanto ao tamanho dos *primers*[6], a inserção de mutações em regiões distantes das extremidades da seqüência é realizada por meio de duas reações de PCR distintas, seguida pela ligação dos produtos resultantes. Nesses casos, além dos *primers forward* (F) e *reverse* (R) correspondentes, respectivamente, ao início e ao final da seqüência que se deseja amplificar, outros dois *primers*, um *forward* (F_m) e um *reverse* (R_m), devem ser construídos contendo as mutações desejadas. Em uma das reações de PCR são utilizados os *primers* F, R_m e a seqüência original a ser mutada. Tal reação resulta na amplificação da primeira metade dessa seqüência, já incluindo a mutação desejada. Na outra reação de PCR, são utilizados os *primers* R e F_m, juntamente com a seqüência original, resultando na amplificação da segunda metade da seqüência contendo a mutação desejada. Ambos os produtos são em seguida submetidos a uma mesma reação de PCR sem a adição de *primers* para que ocorra a remontagem dos fragmentos resultantes de cada uma das reações anteriores. A remontagem das seqüências ocorre devido à existência de

[6] *Primers* de tamanho entre 15 e 20 nucleotídeos são comumente utilizados.

regiões complementares entre os *primers* F_m e R_m. A Figura 2.4 representa esquematicamente as reações envolvidas no processo de SDM, para um caso particular (SDM no meio da seqüência).

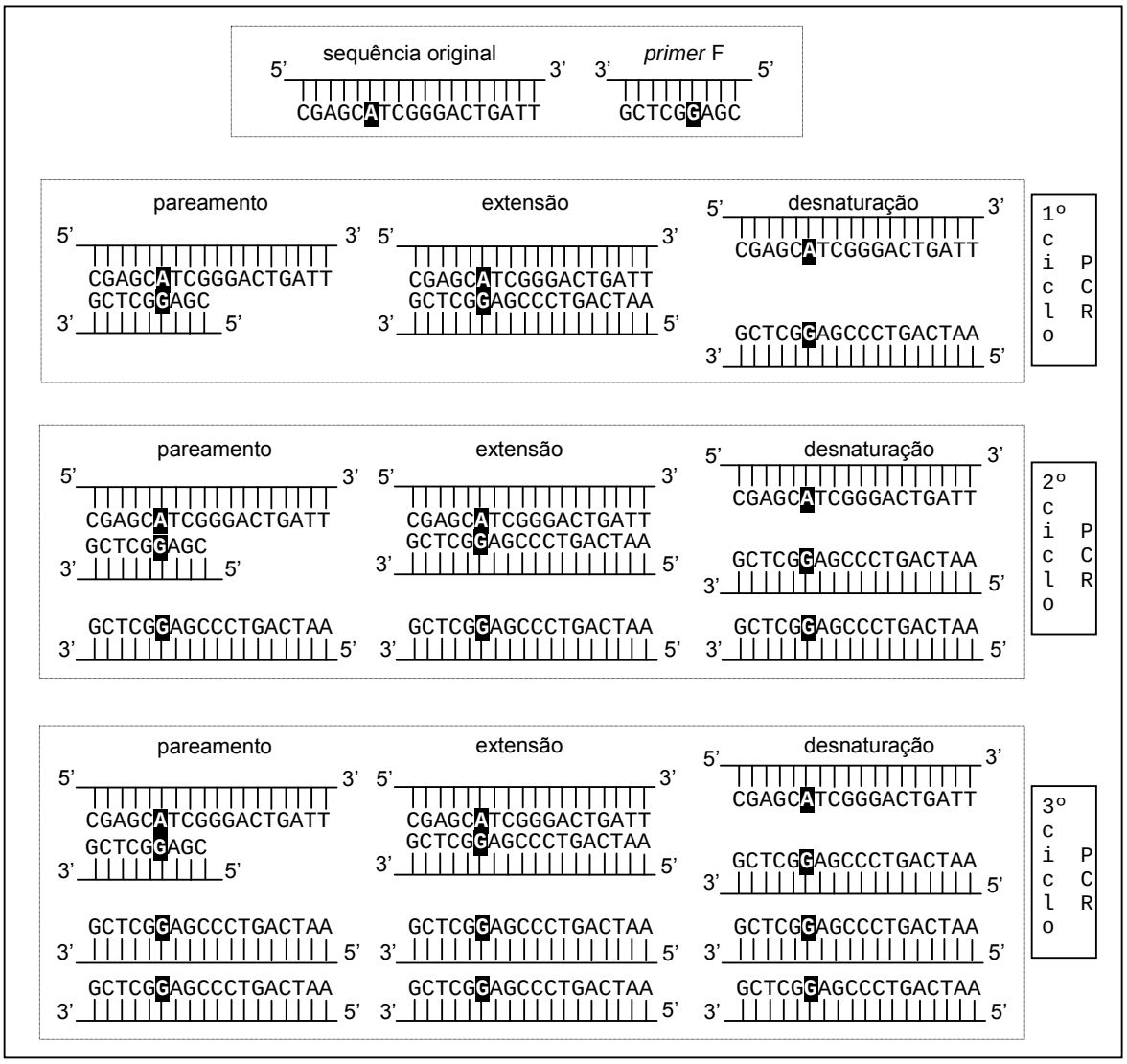


Figura 2.3. Representação esquemática da SDM no início da seqüência. A mutação ocorre pela substituição da Adenina, localizada na sexta posição da seqüência original, pela Guanina nas seqüências resultantes (bases sombreadas de preto). Note que apenas uma das fitas da molécula original é utilizada durante a PCR e que, a cada ciclo, novas seqüências mutantes (fita simples) são produzidas.

Diversos métodos para a realização de mutações sítio-dirigida foram propostos (ZHENG et al., 2004), (URBAN et al., 1997), (KUIPERS et al., 1991) e (LANDT et al., 1990). Um dos mais rápidos e econômicos é o descrito por Landt et al. (1990), e consiste dos seguintes passos:

- Primeira reação de PCR (PCR 1);
- Purificação do produto obtido da PCR 1;
- Segunda reação de PCR (PCR 2);
- Clonagem dos fragmentos obtidos.

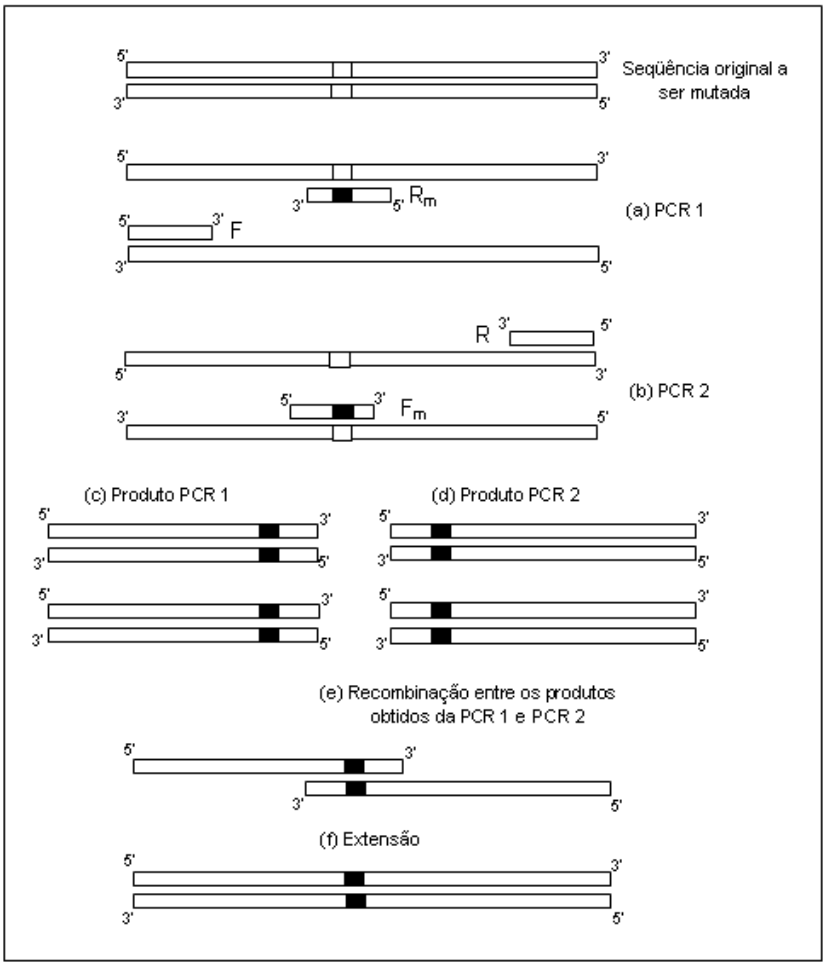


Figura 2.4. Representação esquemática da SDM no meio da seqüência. A mutação irá ocorrer devido ao pareamento incorreto (*mismatch*) entre uma base da molécula de fita simples (representada pelo retângulo branco) e uma base do *primer* (representado pelo retângulo preto). (a) Reação PCR 1 para amplificação da primeira metade da seqüência original resultando em moléculas de fita simples com a mutação desejada. (b) Reação PCR 2 para amplificação da segunda metade da seqüência original resultando em moléculas de fita simples com a mutação desejada. (c) e (d) Produtos resultantes das reações PCR 1 e PCR 2, respectivamente. Os produtos resultantes de ambas as reações são submetidos a uma mesma reação de PCR, sem a adição de *primers*. (e) Recombinação entre os produtos das reações de PCR 1 e PCR 2, por meio do pareamento entre regiões complementares. (f) Extensão dos fragmentos recombinados, resultando nas seqüências com a mutação desejada.

A Figura 2.5, adaptada de Landt et al. (1990), mostra esquematicamente como uma mutação é inserida em um gene a ser clonado. A primeira reação de PCR é utilizada para que sejam produzidos *primers* intermediários, os quais, após um processo de purificação (isolamento/seleção), serão utilizados na segunda reação de PCR. Na primeira reação, apenas um dos dois *primers*, (neste caso o *primer* 1 da Figura 2.5), carrega a mutação desejada e juntos (*primer* 1 e *primer* 2) amplificam apenas o trecho inicial do gene a ser clonado. O produto da primeira reação é então purificado e os *primers* intermediários isolados (*primer* 3 e *primer* 4 da Figura 2.5). Para a segunda reação de PCR são utilizados os *primers* isolados da reação anterior juntamente com um novo *primer* (*primer* 5 da Figura 2.5) complementar à região final do gene. Após a

fragmentação enzimática do produto da segunda reação de PCR, dois tipos de fragmentos são obtidos, sendo eles representados na Figura 2.5 (e) e (f). Apenas os fragmentos do tipo (e) podem ser clonados apropriadamente.

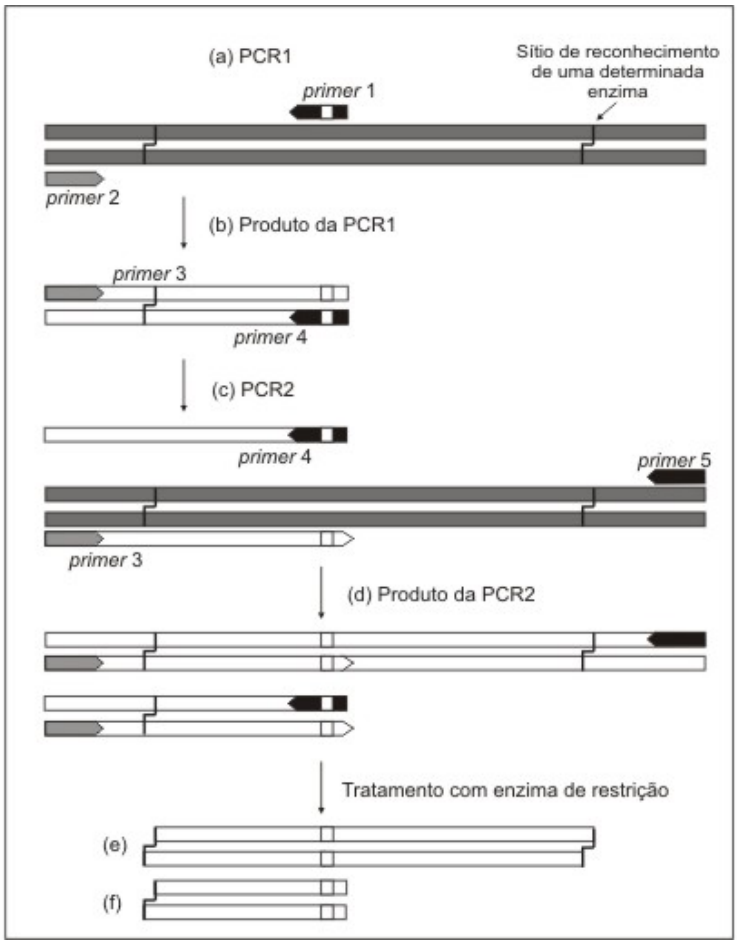


Figura 2.5. Representação esquemática do método de mutação sítio-dirigida. (a) Na primeira reação de PCR a mutação desejada é introduzida (representada pelo pequeno trecho branco no *primer* 1), porém apenas uma porção do gene é amplificada, como mostrado em (b). (c) Na segunda reação de PCR, o produto da primeira reação é utilizado como *primer* juntamente com um novo *primer* para amplificar o gene por inteiro já com a mutação desejada. (d) Como resultado da segunda reação têm-se o gene, bem como uma fração deste amplificados. Após o tratamento do produto desta última reação pela enzima de restrição apropriada, os fragmentos que correspondem ao gene original, agora com a mutação desejada (e), podem ser clonados em vetores apropriados.

2.4 *Oligonucleotide-directed random mutagenesis*

Como visto na Seção 2.3, em experimentos de SDM é necessário um conhecimento prévio da relação estrutura–função da proteína sendo investigada para que se possa direcionar a mutação para os resíduos de aminoácidos que, provavelmente, desempenham um papel fundamental nessa relação. Porém, em muitos casos essa relação pode não ser conhecida ou não estar bem definida,

ou ainda, a seqüência de nucleotídeos da molécula de DNA em estudo pode não ser conhecida, o que limita o uso da técnica de *site-directed mutagenesis*, ou mutação sítio dirigida. Uma alternativa à utilização da mutação sítio-dirigida é a mutação randômica de nucleotídeos. Ao invés de introduzir nas seqüências gênicas mutações previamente estipuladas, uma biblioteca de seqüências randomicamente mutadas é construída. Após a construção dessa biblioteca, as proteínas associadas às seqüências gênicas resultantes podem ser expressas em vetores apropriados e suas respectivas funções analisadas e, em seguida, um estudo da relação seqüência-função feito.

Em experimentos de mutação sítio-dirigida é o pesquisador quem decide qual (ou quais) aminoácido(s) possivelmente desempenha(m) um papel importante na relação estrutura–função e esta escolha é (ou não) confirmada por experimentos realizados posteriormente. Em experimentos de mutação randômica, espera-se que o próprio experimento revele quais substituições são mais importantes. Segundo Hermes et al. (1989), um protocolo de mutagênese randômica deve obedecer aos seguintes critérios:

- a região da seqüência gênica a ser mutada deve ser delimitada;
- todas as bases dentro da região delimitada devem ter a mesma probabilidade de sofrer mutação;
- a probabilidade de substituição de uma determinada base por qualquer uma das quatro bases deve ser a mesma;
- deve ser possível a definição de posições na região delimitada nas quais não devem ocorrer mutações;
- a freqüência de mutação dentro da região delimitada deve ser controlada;
- idealmente, toda seqüência da biblioteca resultante deve conter mutações.

Diversos protocolos para realizar esse tipo de mutação são encontrados na literatura ((JONES, 2005), (MURAKAMI et al., 2002), (SPEE et al., 1993), (CADWELL et al., 1992), (ZHOU et al., 1991), e (LEUNG et al., 1989)). Um dos métodos de mutação randômica comumente utilizado é o *error-prone* PCR (CADWELL et al., 1992), o qual introduz randomicamente mutações durante a reação de PCR por meio da redução da fidelidade da DNA polimerase.

2.5 *Staggered Extension Processes* (StEP)

O método definido como StEP (*Stagerred Extension Process*), proposto por Zhao et al. (1998), consiste basicamente da extensão de *primers* os quais utilizam como molde as seqüências parentais por meio de repetidos ciclos de desnaturação, pareamento e extensão por polimerase. Este método de evolução *in vitro* é visto pelos seus criadores como "um método simples e eficiente para a recombinação e mutagênese *in vitro* de seqüências polinucleotídicas".

Em cada um dos ciclos da reação de PCR, o *primer* sendo estendido pode se parear a diferentes seqüências parentais, de forma que, ao final dos ciclos de desnaturação, pareamento e extensão, a seqüência resultante será composta de fragmentos originários dos diferentes parentais. Na passagem do ciclo i para o ciclo i+1, a possível troca de molde pelo *primer* em extensão é chamada de *switch*. Ciclos curtos de desnaturação, pareamento e extensão são repetidos até que seqüências com o mesmo tamanho dos parentais sejam obtidas.

O sucesso do método, ou seja, a obtenção de seqüências recombinantes em relação aos parentais está basicamente relacionado ao rígido controle da etapa de extensão por polimerase. Longos períodos de extensão durante cada ciclo limitam a ocorrência de recombinações entre os parentais, principalmente quando as mutações entre estes se encontram próximas umas das outras, uma vez que um número menor de *switches* tende a ocorrer.

Além de ser um método de simples execução, outra vantagem é a eliminação da etapa de fragmentação dos parentais por DNase I ou por enzimas de restrição – etapa essa necessária em diversos experimentos de evolução molecular direta, como o DNA *shuffling* (que será visto na seção seguinte), por exemplo – além de utilizar pequenas quantidades de DNA. A Figura 2.6, adaptada de Zhao et al. (1998), representa esquematicamente as etapas do StEP.

A utilização do protocolo StEP, contudo, requer um alto grau de similaridade entre as seqüências parentais bem como um rígido controle dos parâmetros envolvidos na reação de extensão por polimerase como, por exemplo, tempo e temperatura, uma vez que o evento de *switch* é dependente desses dois fatores.

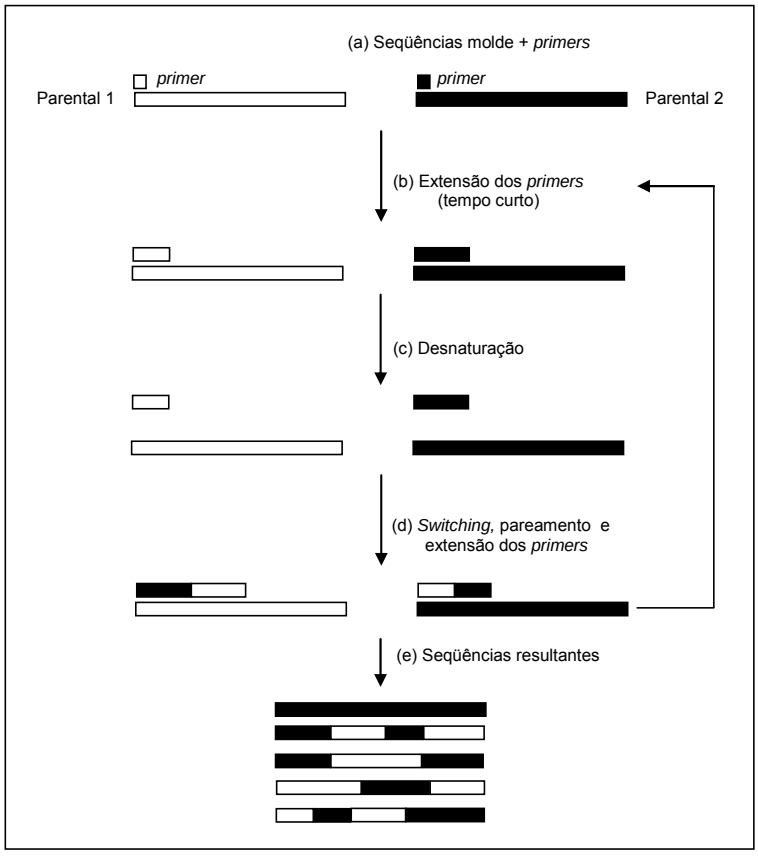


Figura 2.6. Representação esquemática da produção de seqüências recombinantes seguindo a metodologia StEP. Apenas uma das fitas de cada seqüência, e seus respectivos *primers*, são mostrados para simplificar a representação. (a) Pareamento entre os *primers* e as seqüências molde. (b) Extensão dos *primers*. (c) A temperatura da reação é elevada para que os *primers* se soltem dos moldes (desnaturação) e a extensão seja interrompida. (d) Variações de temperatura favorecem o *switch* entre *primers* e seqüência molde. (e) Após a execução cíclica das etapas (b), (c) e (d), tem-se as seqüências resultantes.

2.6 DNA *shuffling*

A técnica de DNA *shuffling* foi desenvolvida inicialmente por Stemmer em 1994 ((STEMMER, 1994a) e (STEMMER, 1994b)) para estudar a evolução molecular dos genes codificadores de β-lactamase. Após seu desenvolvimento, diversas adaptações da metodologia original têm surgido com o objetivo de superar algumas deficiências observadas e aumentar o grau de diversidade das seqüências resultantes.

Um dos pré-requisitos para a aplicação da técnica de DNA *shuffling* é a disponibilidade de seqüências homólogas[7], as quais servirão de base para a construção de um conjunto de moléculas recombinantes, denominado Biblioteca de DNA *shuffling*. Na descrição das etapas do processo de

[7] O termo seqüências homólogas ou homologia é utilizado para referenciar genes que compartilham um ancestral evolucionário em comum revelado pela similaridade de suas seqüências (BROWN, 1999).

shuffling, a homologia entre as seqüências iniciais é fundamental para garantir que fragmentos originários de seqüências distintas recombinem, formando assim um fragmento híbrido que, espera-se, tenha sua funcionalidade melhorada com relação a seus parentais. O método original do DNA *shuffling* pode ser sumarizado pelas seguintes etapas:

1. Seleção dos genes de interesse, ditos parentais;
2. Fragmentação enzimática dos genes;
3. Ciclos de PCR, sem a adição de *primers*, para que ocorra a remontagem dos fragmentos;
4. Amplificação das seqüências remontadas na etapa anterior por meio de PCR com *primers*, a fim de selecionar as seqüências cujo tamanho seja igual a dos parentais.

A Figura 2.7 mostra um esquema geral da metodologia de DNA *shuffling*. Cada uma das quatro etapas deste processo, seleção, fragmentação, ciclos de PCR e amplificação são detalhadas nas subseções 2.6.1, 2.6.2, 2.6.3 e 2.6.4, respectivamente.

2.6.1 Seleção dos genes de interesse

A seleção dos genes de interesse[8] a partir dos quais será construída a biblioteca de DNA *shuffling* é um fator importante o qual, somado a outros fatores envolvidos no processo como, por exemplo, tamanho dos fragmentos, temperatura de pareamento e número de ciclos do PCR, irá influenciar na taxa de diversidade gênica obtida.

As seqüências parentais selecionadas devem ser homólogas, ou seja, devem compartilhar alguma similaridade, para que a técnica de DNA *shuffling* possa ser aplicada com sucesso. Essa exigência decorre da maneira como ocorrem as recombinações entre os fragmentos no processo de remontagem durante os ciclos de PCR. Se por um lado o fator homologia (ou similaridade) é de grande importância para que a técnica de DNA *shuffling* seja empregada, por outro, pode ser responsável por uma baixa taxa de recombinantes na biblioteca resultante.

[8] Também referenciada como seleção dos parentais.

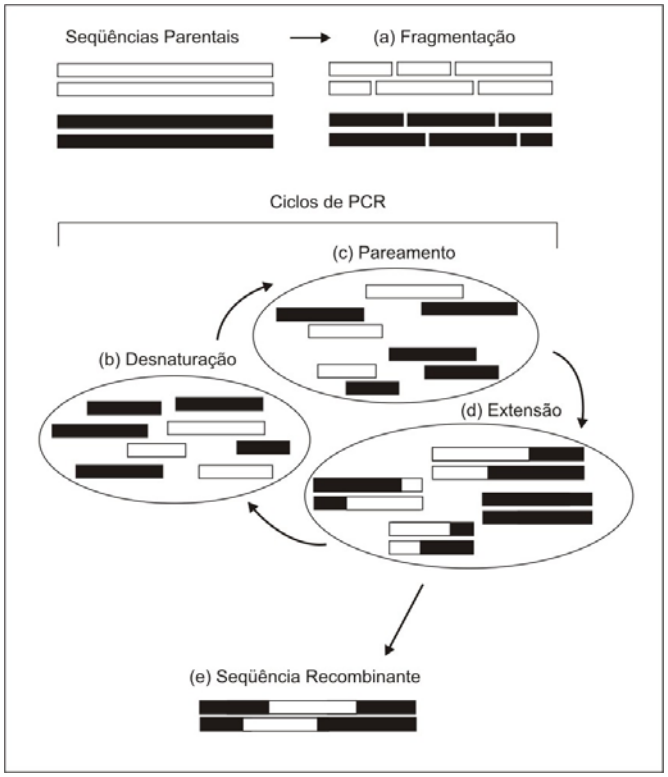


Figura 2.7. Representação esquemática do processo de DNA *shuffling* entre dois parentais. (a) Fragmentação enzimática das seqüências parentais. (b) Desnaturação dos fragmentos resultando em fragmentos de fita simples. (c) Pareamento entre fragmentos de fita simples que compartilham regiões de bases complementares. (d) Extensão por polimerase dos fragmentos pareados. (e) Exemplo de uma seqüência resultante do processo de *shuffling*, a qual é composta pela união de fragmentos originários de ambos os parentais.

2.6.2 Fragmentação

Uma vez escolhidos os genes parentais, eles devem ser fragmentados. A fragmentação pode ser um processo randômico ou não. Para a fragmentação randômica, utiliza-se principalmente a enzima DNase I, a qual produz cortes em posições randômicas em cada uma das fitas que compõem uma molécula de DNA. Não se tem um controle exato sobre o tamanho dos fragmentos produzidos por esse tipo de fragmentação, porém sabe-se que quanto maior o tempo de atuação da enzima sobre o substrato (DNA), maior será o número de cortes produzidos e, desta forma, menores serão os fragmentos resultantes. A fragmentação não randômica pode ser feita através da utilização de enzimas de restrição, de forma que apenas regiões específicas da seqüência de DNA sejam cortadas (ver Capítulo 1, Seção 1.5), sendo possível assim prever o tamanho e o número de fragmentos resultantes desse tipo de fragmentação.

Os fragmentos resultantes deste processo são então purificados por meio de eletroforese em gel de agarose para que aqueles cujo tamanho medido em pares de bases, esteja

compreendido em um intervalo de interesse, sejam selecionados para dar continuidade ao processo. Stemmer (1994a e 1994b) sugere que fragmentos entre 10 e 50 pb sejam utilizados. Contudo, diferentes experimentos relatam a utilização de diferentes tamanhos de fragmentos.

2.6.3 Ciclos de PCR

É nesta etapa que ocorrem as recombinações gênicas, ou seja, fragmentos originários de parentais distintos podem se unir (recombinar) em um único fragmento originando um fragmento híbrido. A obtenção de fragmentos híbridos é o que garante a diversidade da biblioteca sendo construída. Esta etapa corresponde à execução cíclica dos eventos de desnaturação, pareamento e extensão dos fragmentos. A reação de PCR é dependente de parâmetros como tamanho e concentração dos fragmentos, temperatura de pareamento e desnaturação, entre outros. Metzker e Caskey (2001) apresentam um estudo detalhado sobre a reação de PCR incluindo discussões sobre os seus resultados, dependências e limitações. A seguir, são descritos cada um dos três eventos envolvidos na reação de PCR.

Desnaturação

Como a molécula de DNA é composta por duas fitas de nucleotídeos unidas entre si por meio das pontes de hidrogênio formadas entre os pares de bases complementares, os fragmentos resultantes da ação da endonuclease (após a etapa de fragmentação) são fragmentos de fita dupla. Depois de selecionados por meio de eletroforese em gel de agarose, os fragmentos de tamanho desejado são então aquecidos para que as pontes de hidrogênio existentes entre as bases complementares se quebrem, de forma que as fitas duplas se separem em fitas simples; a partir deste ponto os fragmentos de DNA são fitas simples. Este processo é chamado desnaturação e ocorre sob temperaturas próximas de 94ºC, uma vez que, sob essa temperatura, as pontes de hidrogênio das moléculas de DNA são quebradas e a enzima polimerase, que será utilizada posteriormente durante a extensão, ainda mantém sua atividade.

Pareamento

Nesta fase, os fragmentos resultantes da desnaturação irão se parear, ou seja, espera-se que o pareamento entre trechos de bases complementares dos fragmentos ocorra, possibilitando assim que eles sejam novamente transformados em fragmentos de fita dupla, devido sua extensão pela ação de uma polimerase, como descrito na subseção seguinte.

Para que o pareamento entre as bases complementares ocorra, a temperatura é diminuída para valores em torno de 55ºC uma vez que, sob essas temperaturas, o pareamento entre bases complementares é favorecido devido à tendência das moléculas voltarem à sua conformação original, e mais estável, de dupla fita.

As moléculas resultantes do pareamento podem ser de dois tipos: homoduplex, quando os fragmentos que se unem são provenientes do mesmo parental e heteroduplex, quando os fragmentos unidos são provenientes de parentais distintos. Quanto maior for o número de moléculas heteroduplexes formadas, maior será a taxa de recombinação alcançada pelo experimento. A verificação da taxa de homoduplex e heteroduplex gerada pelo *shuffling* pode ser realizada por meio de ferramentas computacionais, como por exemplo, as que realizam o alinhamento entre seqüências resultantes e seqüências parentais. É importante notar que quanto maior a taxa de heteroduplex gerada, maior a diversidade gênica obtida pelo processo e, por conseqüência, maiores são as chances de se obter moléculas com funcionalidades melhoradas, mais específicas ou, ainda, moléculas que apresentam novas funcionalidades em relação às moléculas parentais.

A importância do fator similaridade entre as seqüências escolhidas como parentais na criação da biblioteca de *shuffling* fica evidente nesta etapa, uma vez que a escolha de seqüências pouco similares diminui as chances de formação de moléculas heteroduplexes, pelo fato de seus fragmentos compartilharem pouca complementaridade, dificultando assim o pareamento entre eles.

Extensão

Diversas situações distintas podem ser verificadas como resultado do pareamento entre dois fragmentos. Considere dois fragmentos de DNA de tamanhos m e n quaisquer. De uma maneira geral, sem considerar sua composição, os fragmentos podem assumir uma dentre todas as configurações de pareamento apresentadas na Figura 2.8, sendo (a) $n = m$ e (b) $m \neq n$.

Exceto no caso em que ambos os fragmentos possuem o mesmo tamanho e, além disso, sejam totalmente complementares entre si, como ilustrado na Figura 2.8 (a.1), todas as demais configurações de pareamento entre dois fragmentos necessitam ser estendidas para que se tornem, de fato, um fragmento de fita dupla. A extensão é obtida por meio da atuação de enzimas do tipo DNA polimerase. Como foi visto na Seção 1.4 do Capítulo 1, para que a polimerase atue na extensão dos fragmentos de fita simples, são necessários, além da temperatura

adequada, o molde e o *primer*. Após o pareamento, um dos fragmentos desempenhará o papel de *primer* e o outro o papel de molde para a produção da fita complementar. A extensão, portanto, pode ocorrer sem a adição de *primers*. Por esta razão, esta etapa é também referenciada como PCR sem *primers*.

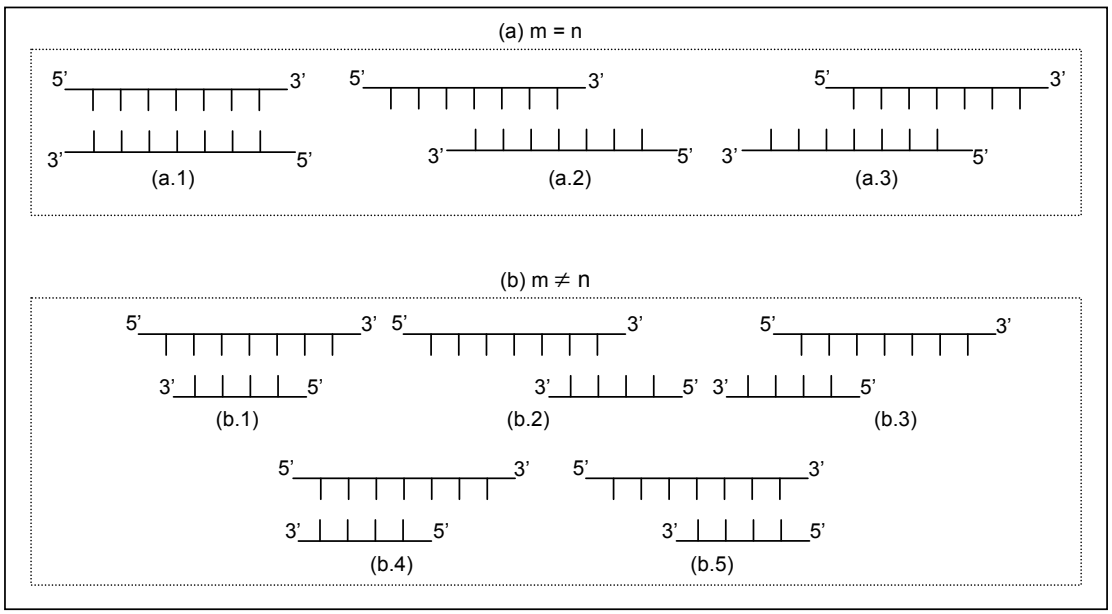


Figura 2.8. Representação dos possíveis pareamentos entre dois fragmentos de tamanhos m e n.

É importante lembrar que a extensão por polimerase só ocorre no sentido 5'→ 3', de forma que, possivelmente, nem todos os fragmentos pareados serão estendidos. Sempre que o pareamento entre dois fragmentos de tamanhos distintos assume a configuração ilustrada na Figura 2.8 (b.1), apenas uma extensão parcial será possível, pois, nestes casos existe somente uma extremidade 3' disponível para a extensão, de forma que um fragmento completo de fita dupla nunca será obtido para este tipo de pareamento. A Figura 2.9 ilustra esta situação de extensão parcial.

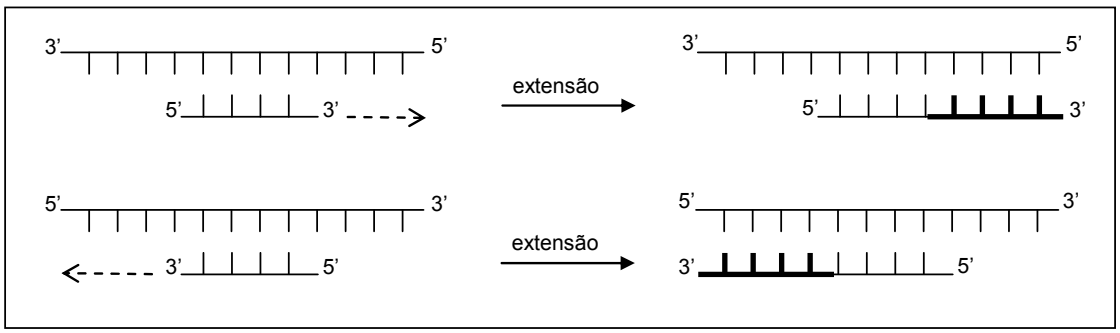


Figura 2.9. Representação de possíveis configurações de pareamento entre dois fragmentos para o qual é possível apenas uma extensão parcial pela polimerase. Para este tipo de configuração, a extensão nunca irá resultar em uma molécula de fita dupla completa.

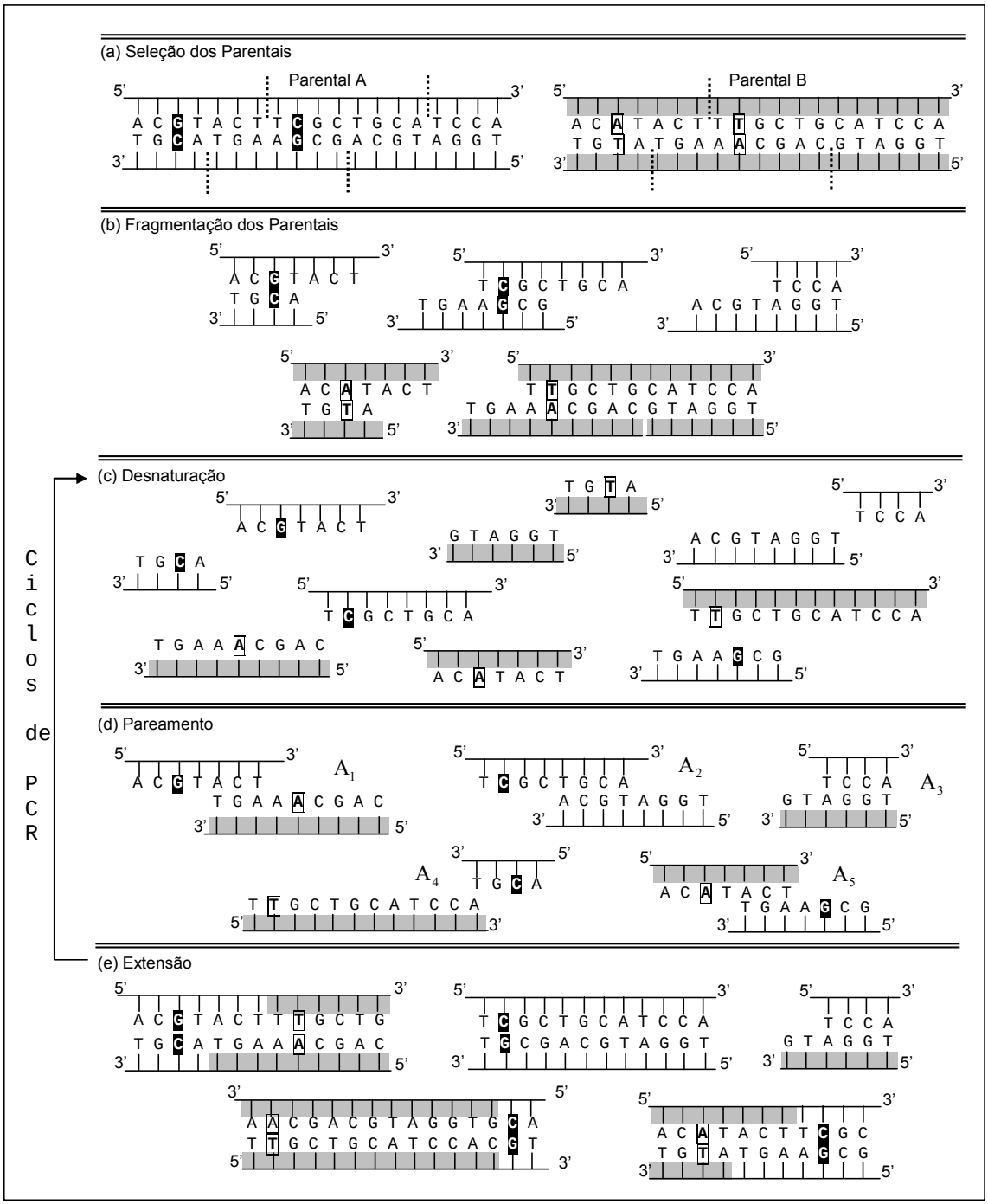


Figura 2.10. Representação esquemática detalhada das etapas do processo de DNA *shuffling* sem a etapa final de amplificação: (a) Seleção dos Parentais. (b) Fragmentação. (c) Desnaturação. (d) Pareamento. (e) Extensão. Os pareamentos indicados por A_1, A_3, A_4 e A_5 correspondem a heteroduplexes, enquanto que em A_2 é formado um homoduplex.

Os pareamentos entre os fragmentos A_1, A_2, A_3, A_4 e A_5 mostrados na figura não são únicos, porém permitem visualizar como os homoduplexes e heteroduplexes são formados e quais fragmentos resultam em diversidade gênica em relação aos parentais. A extensão do heteroduplex formado pelo pareamento A_1 indicado na Figura 2.10 resulta em um fragmento

recombinante de fita dupla no qual a diversidade gênica foi obtida, uma vez que este fragmento contém em sua seqüência dois pares de bases, no caso os pares de bases G–C e T–A (correspondentes às mutações), os quais inicialmente se encontravam separados nos parentais A e B, respectivamente. Situação análoga ocorre com os pareamentos A_4 e A_5. O heteroduplex resultante do pareamento indicado por A_3 na Figura 2.10 não é estendido, uma vez que a extremidade livre tem orientação 5'. Supondo que a extensão deste pareamento A_3 fosse possível, apesar do fragmento estendido resultante ser um heteroduplex, este não contribuiria na geração de diversidade, uma vez que não irá conter em sua seqüência, pares de bases distintos existentes entre as seqüências parentais, como acontece, por exemplo, quando o pareamento A_1 é estendido. Este tipo de heteroduplex (ou cruzamento) é chamado silencioso. O pareamento A_2 é resultante da união de dois fragmentos originários de um mesmo parental, ou seja, é um homoduplex e, desta forma, não introduz diversidade.

2.6.4 Amplificação

Após a execução dos ciclos de PCR, os fragmentos resultantes são submetidos novamente a uma reação de PCR, agora com a adição de *primers*. Os *primers* utilizados correspondem a oligonucleotídeos contendo trechos das extremidades 5' → 3' e 3' → 5' das seqüências parentais. O objetivo de finalizar o processo de DNA *shuffling* realizando uma PCR com *primers* é amplificar os fragmentos recombinados que, possivelmente, possuem o mesmo tamanho dos genes parentais (chamados *full-length*) para que estes possam, posteriormente, ter sua seqüência determinada e/ou serem expressos em vetores de expressão.

Após a amplificação e seleção dos *full-length*, estes fragmentos são clonados, resultando na biblioteca de DNA *shuffling* ou biblioteca de recombinantes. O processo de clonagem foi apresentado do Capítulo 1, Seção 1.5. Os clones da biblioteca serão posteriormente expressos em vetores de expressão, que podem ser, por exemplo, bactérias ou leveduras, para que o produto protéico correspondente seja obtido e ensaios de suas atividades possam ser realizados.

2.7 *Family shuffling*

Diferentemente do DNA *shuffling*, o qual originalmente utiliza apenas genes de uma mesma progênie[9], Crameri e colaboradores (CRAMERI et al., 1998) foram os pioneiros em utilizar a técnica com genes homólogos de diversas espécies; tal aplicação recebeu o nome de *Family*

[9] Geração, prole.

shuffling, porém muitas vezes ela é simplesmente chamada de DNA *shuffling*. A utilização de genes homólogos de diversas espécies aumenta a diversidade funcional inicial do processo, uma vez que os parentais já acumulam em suas seqüências mutações benéficas ocorridas naturalmente ao longo dos anos, e a recombinação destes, por sua vez, pode acelerar ainda mais o processo de evolução *in vitro*.

Nos experimentos descritos por Crameri et al. (1998) foram utilizados quatro genes codificadores de cefalosporinases C originários de quatro espécies microbianas distintas (*Citrobacter freundii, Enterobacter cloacae, Klebsiella pneumoniae* e *Yersinia enterocolitica*). Para avaliar se a diversidade natural presente nas seqüências parentais poderia acelerar ainda mais o processo de evolução *in vitro*, foram realizados experimentos de *shuffling* com cada um dos quatro genes separadamente, bem como outro experimento utilizando todos os quatro genes como parentais. O melhor clone selecionado da biblioteca resultante do *shuffling* dos quatro genes codificadores de cefalosporinases apresentou uma melhora de 540-*fold* de resistência ao antibiótico moxalactam em relação aos parentais *Klebsiellae* e *Yersinia* e uma melhora de 270-*fold* em relação aos parentais *Enterobacter* e *Citrobacter*, enquanto que uma melhora de apenas 8-*fold* foi observada nos experimentos considerando cada uma dos parentais separadamente. Desta forma, os experimentos descritos comprovaram que o *Family shuffling* acelera a evolução *in vitro* de uma maneira mais acentuada do que o *shuffling* como proposto originalmente.

2.8 *Random-priming recombination*

Proposto por Shao et al. (1998), a metodologia do processo descrito como *random-priming recombination* (RPR) utiliza pequenos *primers* randômicos para gerar um grande número de pequenos fragmentos de DNA complementares a diversas regiões das seqüências parentais. Esses fragmentos são então submetidos à etapa de remontagem. Fragmentos que compartilham regiões complementares podem se parear uns aos outros e, em seguida, ser estendidos por uma polimerase. Assim, após um determinado número de ciclos, fragmentos completamente remontados, ou seja, com o mesmo tamanho dos parentais (*full-length*), podem ser obtidos. Os autores afirmam ainda que a RPR apresenta diversas vantagens quando comparada com a técnica de DNA *shuffling*, tais como:

– a possibilidade de utilização de moléculas de fita simples, sem que seja necessário uma etapa intermediária ao processo para a síntese da fita complementar;

– a técnica de DNA *shuffling* requer a fragmentação de moléculas de fita dupla de DNA, o que é, na maioria das vezes, feito com a enzima DNase I. Após a fragmentação, tal enzima deve ser completamente inativada para que não interfira nas etapas seguintes do processo;
– os *primers* utilizados são randômicos e possuem tamanho uniforme; é possível pois, dizer que os *primers* não têm "preferência" por uma região especifica da seqüência parental, de forma que irão se ligar à seqüência parental em diversas posições, garantindo, ao menos em princípio, que todo nucleotídeo da seqüência parental será copiado em uma mesma freqüência durante a extensão;
– a aplicação desta técnica não é dependente do tamanho da seqüência de DNA parental. Fragmentos de DNA menores que 200 bases podem ser utilizados como parentais bem como fragmentos longos, como por exemplo, plasmídeos linearizados e DNAs de fago λ;
– uma quantia de 10 a 20 vezes menor de moléculas iniciais é utilizada pela RPR uma vez que a molécula de DNA será utilizada somente como molde para a extensão dos *primers* randômicos.

A Figura 2.11, adaptada de Shao et al. (1998), mostra uma representação esquemática da metodologia de RPR.

2.9 *Restriction fragment shuffling*

O método de DNA *shuffling* foi utilizado por Kikuchi et al. (1999) para obter recombinantes a partir dos genes *xylE* e *nahH*, cujas seqüências são 84% idênticas. Porém a taxa observada de recombinantes presentes na biblioteca resultante foi menor que 1%, ou seja, a maioria das seqüências remontadas era composta apenas por fragmentos originários de um mesmo parental. Na tentativa de aumentar a taxa de recombinação, um conjunto de enzimas de restrição foi utilizado na fragmentação dos genes parentais ao invés da fragmentação randômica com a enzima DNase I.

Em um primeiro experimento, as enzimas de restrição *Ava*I e *Dde*I foram utilizadas para fragmentar o gene *nahH* e a enzima *Nc*oI utilizada para fragmentar o gene *xylE*. Após a realização do experimento de DNA *shuffling*, uma amostra de dez seqüências resultantes foi seqüenciada e todas elas apresentaram fragmentos originários de ambos os parentais, resultando assim em uma

freqüência de 100% de recombinantes na amostra avaliada. Foram realizados ainda outros experimentos similares, utilizando outras enzimas de restrição, e a taxa de recombinantes obtida mostrou-se elevada novamente.

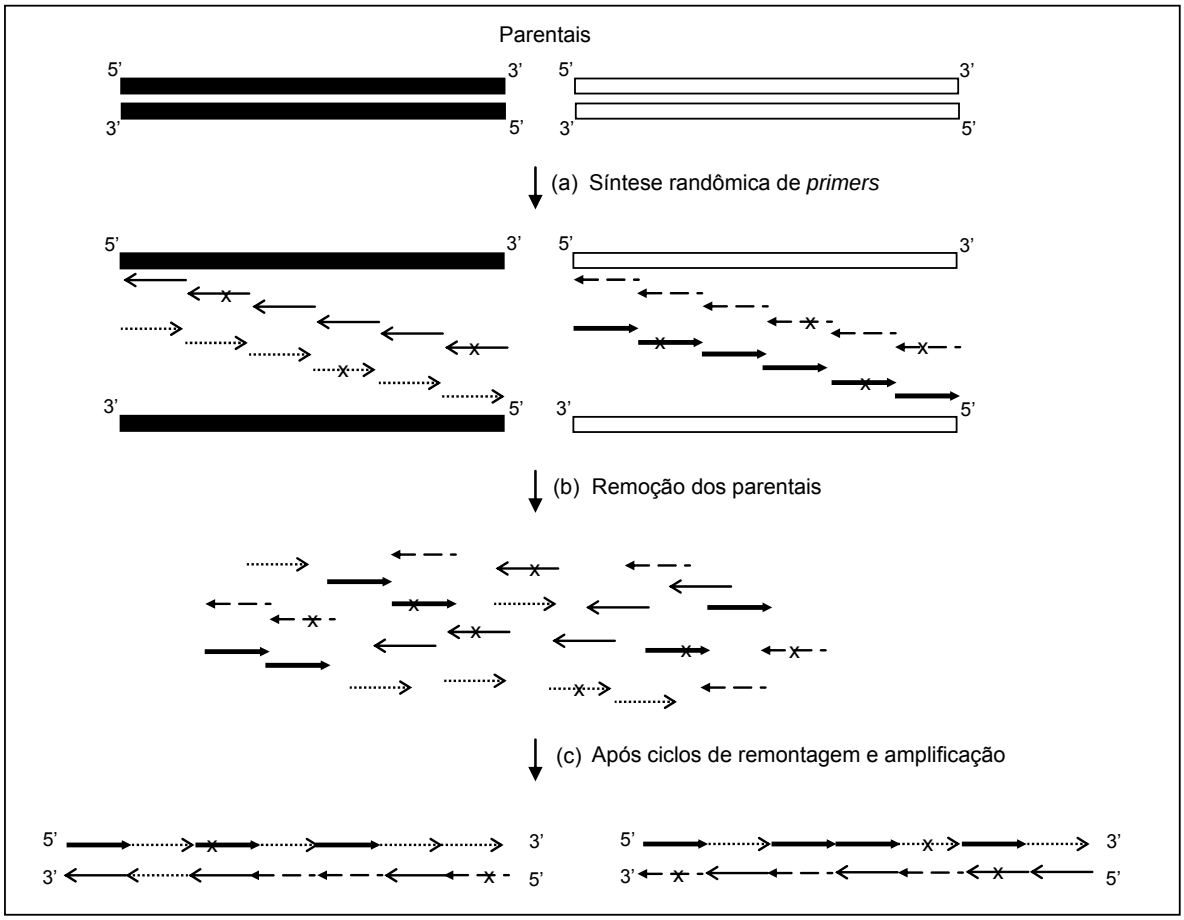


Figura 2.11. Esquema geral da metodologia *Random-priming recombination*. (a) Síntese de pequenos fragmentos de fita simples complementares aos parentais a partir de *primers* randômicos. As posições marcadas com X's representam novas mutações introduzidas pela reação de PCR. (b) Remoção das seqüências parentais. (c) Remontagem e amplificação dos fragmentos resultantes da reação (b).

Apesar do sucesso nos experimentos realizados, a principal limitação desta técnica é a possibilidade de ocorrência de cruzamentos[10] apenas nos locais onde existem os sítios de restrição para as enzimas utilizadas, de forma que a diversidade da biblioteca resultante fica limitada.

[10] O termo *crossover* é também empregado para indicar que houve uma recombinação (cruzamento) entre parentais distintos.

2.10 ITCHY

Como comentado anteriormente, a utilização da metodologia do DNA *shuffling* ou de alguma de suas variantes ((CRAMERI et al., 1998), (ZHAO et al., 1998), (SHAO et al., 1998)), requer que as seqüências a serem recombinadas apresentem alto grau de similaridade (ver Seção 2.6.1). Apesar de tais técnicas terem sido utilizadas com sucesso em diversos experimentos descritos na literatura, elas deixam de explorar uma grande porção do espaço total de recombinações possíveis, devido ao fato de que cruzamentos só podem ocorrer em regiões de homologia entre os parentais. É preciso considerar, entretanto, que cruzamentos localizados em regiões não homólogas entre os parentais podem ser igualmente benéficos.

A metodologia denominada ITCHY (*Incremental Truncation for the Creation of Hybrid enzymes*), desenvolvida por Ostermeier et al. (1999) pode ser utilizada para a criação de biblioteca de seqüências híbridas originárias de duas seqüências parentais independentemente da homologia existente entre elas. A principal etapa dessa metodologia envolve a fragmentação dos parentais com a enzima Exonuclease III (Exo III). Uma das atividades da enzima Exo III é remover nucleotídeos a partir da extremidade 3' OH de uma molécula de fita dupla de DNA, resultando em moléculas com extremidades 5' não pareadas. A Figura 2.12 apresenta uma representação esquemática da atividade da enzima Exo III. Como visto no Capítulo 1, a extremidade 5' da molecula de DNA possui um grupo fosfato, representado por P na Figura 2.12.

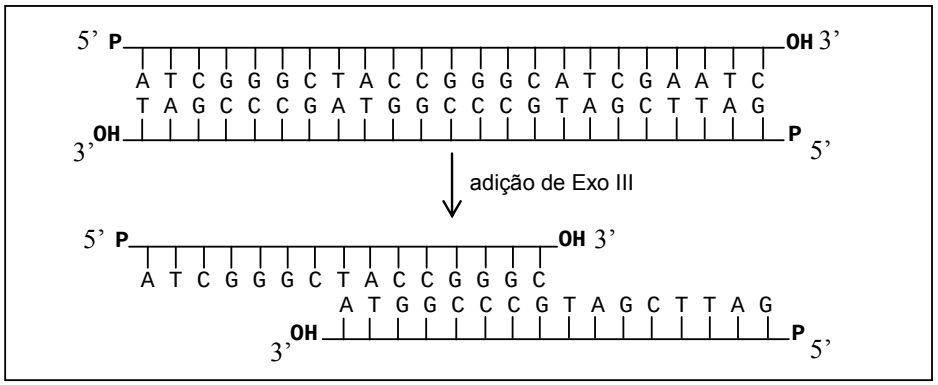


Figura 2.12. Representação da atividade da enzima Exo III, que remove nucleotídeos a partir da extremidade 3' OH.

Outra enzima que desempenha um papel importante na implementação do ITCHY é a Nuclease S1. Tal enzima degrada moléculas de fita simples de DNA ou RNA a partir da extremidade 5'. Desta forma, é possível remover as extremidades de fita simples de fragmentos

tratados com Exo III, para produzir fragmentos *blunt end*, ou seja, fragmentos com extremidades pareadas. Na Figura 2.13 é representada a ação da enzima Nuclease S1 sobre o fragmento tratado com Exo III, mostrado anteriormente na Figura 2.12.

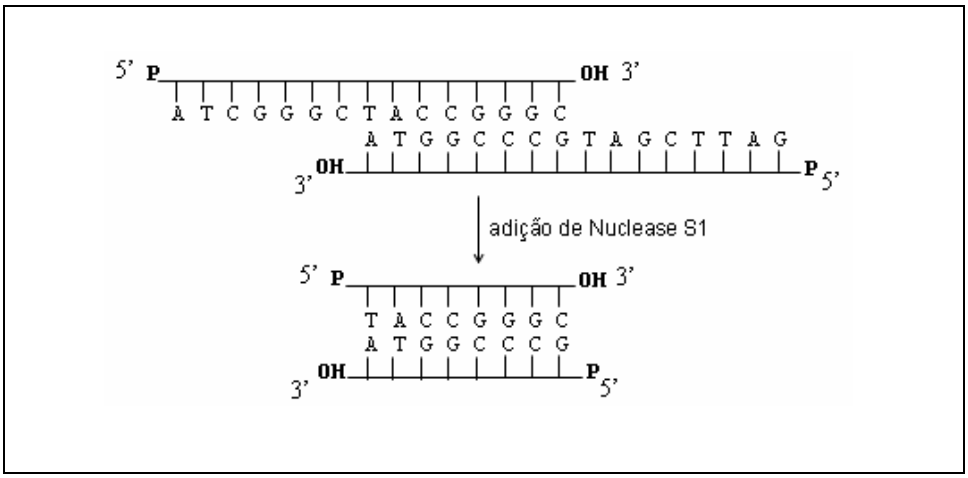


Figura 2.13. Atuação da enzima Nuclease S1, que remove nucleotídeos a partir da extremidade 5', resultando em fragmento *blunt end.*

De uma maneira geral, a metodologia ITCHY envolve a produção de fragmentos truncados em direções opostas e originários de parentais distintos, ou seja, dado o parental 1, os fragmentos produzidos a partir deste parental serão resultantes da remoção de nucleotídeos da sua seqüência na direção 5'→ 3', enquanto que os fragmentos originários do parental 2 serão resultantes da remoção de nucleotídeos da seqüência na direção 3'→ 5'. Para que a remoção dos nucleotídeos ocorra em apenas um das fitas da molécula parental, apenas uma de suas extremidades 3' deve estar livre para sofrer a ação da enzima Exo III. Para tal, os parentais são inseridos em vetores de clonagem apropriados, que contém sítios específicos para determinadas enzimas de restrição, o que irá permitir a "liberação" apenas da extremidade desejada para que a fragmentação de cada parental ocorra.

Sem considerar a inserção dos parentais em vetores, a Figura 2.14 representa esquematicamente a fragmentação de dois parentais, ditos 1 e 2, pela Exo III, seguida do tratamento por Nuclease I e da ligação dos fragmentos resultantes pela enzima Ligase. O fragmento resultante é um híbrido que representa a ocorrência de um cruzamento entre as seqüências parentais.

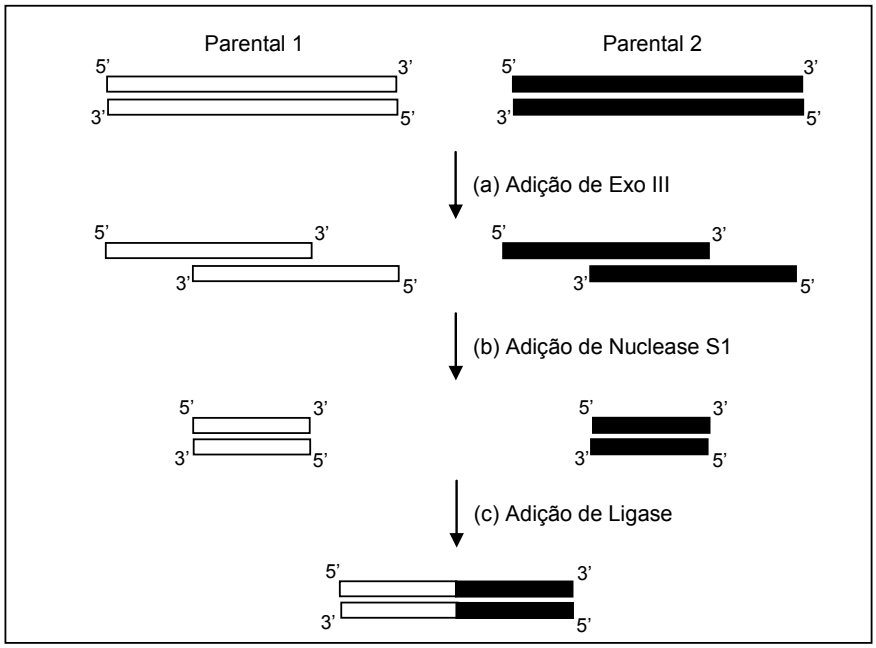


Figura 2.14. Representação esquemática da fragmentação de dois parentais pela enzima Exo III, seguida por tratamento com Nuclease S1 e ligação dos fragmentos por Ligase. (a) Os parentais são tratados com Exo III. (b) Pela ação da Nuclease S1 as extremidades não pareadas são eliminadas. (c) Os fragmentos resultantes da ação da Exo III e da Nuclease S1 são ligados pela enzima Ligase, resultando em um fragmento híbrido.

Para que fragmentos de diversos tamanhos sejam obtidos, a reação de fragmentação dos parentais é controlada e pequenos volumes de material submetido à reação (DNA) são removidos em intervalos de tempo determinados de tal forma que sejam produzidos, teoricamente, fragmentos que diferem em tamanho de apenas um par de bases.

Note que essa metodologia é capaz de produzir seqüências resultantes da ocorrência de apenas um cruzamento entre os parentais, enquanto que outras, como o DNA *shuffling*, por exemplo, podem produzir seqüências com múltiplos pontos de cruzamentos. Porém, ITCHY é capaz de produzir, independente de homologia, um cruzamento em qualquer posição entre duas seqüências parentais. A Figura 2.15, adaptada de Ostermeier et al. (1999), apresenta esquematicamente a criação de uma biblioteca híbrida originária de dois parentais por meio da metodologia ITCHY.

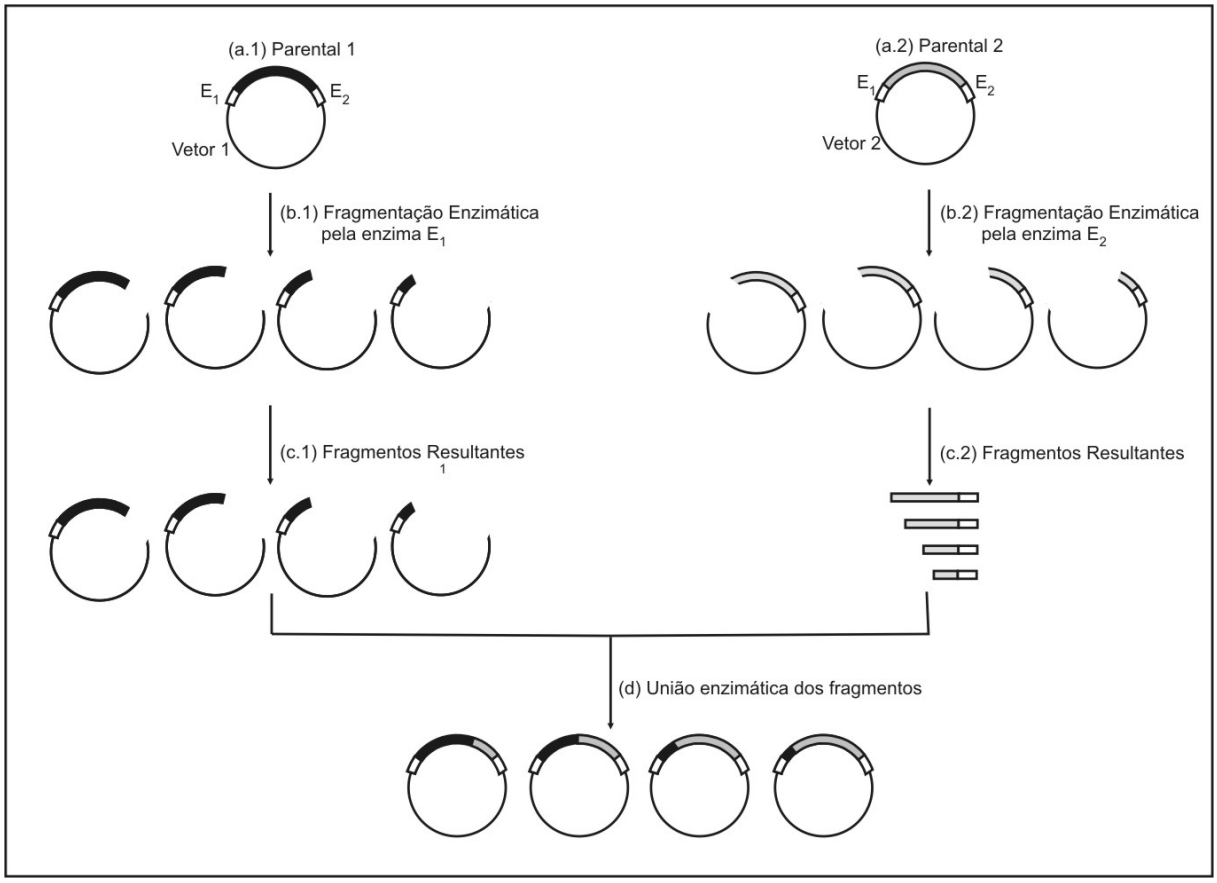


Figura 2.15. Representação esquemática da metodologia ITCHY. (a.1) e (a.2) As seqüências parentais são inseridas em vetores apropriados. (b.1) Os vetores do tipo 1 são fragmentados pela enzima de restrição que reconhece o sítio E_1. (b.2) Os vetores do tipo 2 são fragmentados pela enzima de restrição que reconhece o sítio E_2. Nas reações (b.1) e (b.2) as enzimas Exo III e Nuclease S1 também são adicionadas, e alíquotas de ambas as reações são retiradas em intervalos de tempo determinado para que sejam obtidos fragmentos de diferentes tamanhos. (c.1) e (c.2) Os fragmentos resultantes de ambas as reações são purificados. (d) União dos fragmentos resultantes.

Observe no esquema apresentado na Figura 2.15 (c) e (d) que, devido às enzimas de restrição utilizadas para cada um dos vetores 1 e 2, enzimas que reconhecem os sítios E_1 e E_2, respectivamente, todas as seqüências resultantes (Figura 2.15 (d)) terão em sua extremidade 5' fragmentos originários do Parental 1, e na extremidade 3' fragmentos originários do Parental 2. A utilização das enzimas reconhecedoras dos sítios E_2 e E_1 nos vetores 1 e 2 respectivamente, resultaria em seqüências remontadas em ordem inversa.

Para evidenciar a eficiência da metodologia e comparar seus resultados com os resultados produzidos pelo DNA *shuffling*, foram utilizados fragmentos dos genes purN e GART, originários de *Escherichia coli* e humano, respectivamente, que codificam para a enzima GAR transformilase. Esses fragmentos apresentam apenas 50% de identidade. Dois experimentos foram realizados com essas duas seqüências, um segundo o protocolo ITCHY e outro segundo o DNA *shuffling*.

Uma amostra dos clones ativos de ambas as bibliotecas foi selecionada e analisada. Na amostra originária do ITCHY, foram encontrados cruzamentos em regiões de alta e baixa homologia, enquanto nos clones resultantes do DNA *shuffling* foram encontrados cruzamentos apenas na região de alta homologia entre os parentais, como era de se esperar.

2.11 *Single-stranded* DNA

Nas bibliotecas resultantes de experimentos de DNA *shuffling*, bem como nas resultantes de *Family shuffling*, observa-se a tendência da remontagem dos fragmentos (durante os ciclos de PCR sem *primers*) resultarem em seqüências iguais às parentais. Essa tendência está principalmente relacionada ao fato de que, para que o pareamento entre dois fragmentos ocorra, estes devem compartilhar uma região de complementaridade, o que, por sua vez, favorece a formação de homoduplexes em detrimento à formação de heteroduplexes.

Na tentativa de minimizar a formação de homoduplexes observada no seu trabalho anterior (KIKUCHI et al., 1999), Kikuchi e sua equipe, no trabalho descrito em (KIKUCHI et al., 2000), utilizaram apenas moléculas de fita simples de DNA, denominadas ssDNA (*single-stranded* DNA), de dois genes relacionados (os mesmos genes utilizados em seu trabalho anterior (KIKUCHI et al., 1999) – *xylE* e *nahH*) para realizar o *Family shuffling* e conseguiram verificar uma redução na formação de homoduplexes.

No experimento em questão, apenas uma das fitas do DNA de cada parental foi utilizada, sendo ambas complementares, ou seja, de um dos parentais utilizou-se a fita 5'→ 3' enquanto que do outro se utilizou a fita complementar (3'→ 5'). A Figura 2.16, adaptada de Silva et al. (2003), mostra um esquema simplificado da técnica de *Family shuffling* entre ssDNA de dois parentais distintos.

Paralelamente ao experimento utilizando ssDNA, os autores realizaram outro experimento, com os mesmos genes, utilizando agora moléculas de fita dupla, denominadas dsDNA (*double-stranded* DNA). A única diferença entre os dois experimentos realizados foi a utilização de moléculas de ssDNA ou dsDNA. Uma amostra de cinqüenta seqüências foi randomicamente selecionada da biblioteca resultante de ambos os experimentos e seqüenciada[11]. A análise dessas seqüências revelou que não havia nenhuma seqüência recombinante dentre as resultantes do

[11] Diz-se que uma seqüência foi seqüenciada quando sua composição de bases tiver sido determinada por um processo chamado seqüenciamento.

dsDNA *shuffling*, enquanto que no experimento utilizando ssDNA foram encontradas 14 recombinantes, evidenciando que a utilização de ssDNA favorece a formação de heteroduplexes.

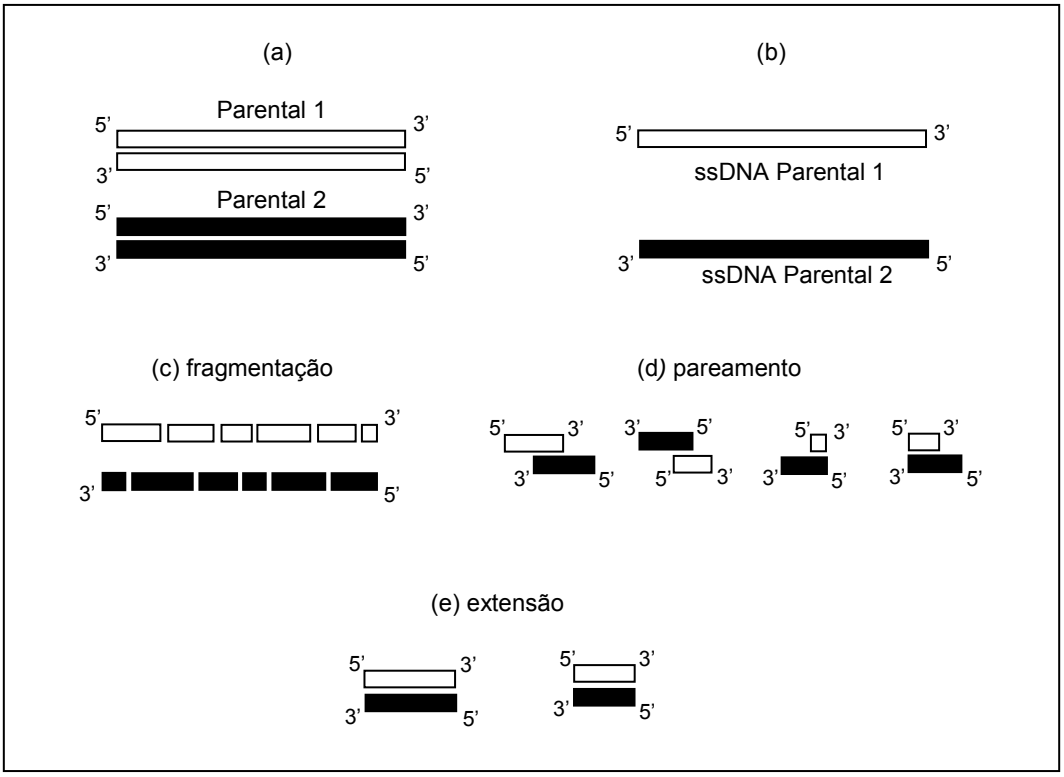


Figura 2.16. Representação esquemática da técnica de *Family shuffling* utilizando apenas ssDNA complementares dos parentais. (a) Moléculas de fita dupla de DNA dos parentais 1 e 2. (b) Fitas simples de DNA (ssDNA) de orientações opostas correspondentes aos parentais. (c) Fragmentação enzimática das ssDNA. (d) Pareamento entre fragmentos originários de parentais distintos. (e) Extensão por polimerase resultando em fragmentos heteroduplexes.

2.12 *Heteroduplex recombination*

Os métodos de recombinação baseados na reação de PCR requerem que quantidades significativas de DNA (ou fragmentos de DNA) estejam disponíveis para a etapa de remontagem e/ou recombinação. Embora, teoricamente, a reação de PCR possa ser utilizada para amplificar até mesmo seqüências muito longas, na prática, a eficiência da amplificação diminui significantemente para este tipo de seqüência, além do número de pontos de mutação introduzidos nesse tipo de amplificação ser potencialmente maior.

Volkov e sua equipe descreveram em Volkov et al. (1999) um método simples para a criação de bibliotecas de seqüências quiméricas de DNA que utiliza a capacidade *in vivo* que algumas células possuem de reparar regiões de não complementaridade (*mismatch*) entre duas seqüências de DNA pareadas para criar novas seqüências compostas por fragmentos de cada um

dos parentais. Essa metodologia, chamada de formação *in vitro* de heteroduplex e reparo *in vivo*, ou simplesmente recombinação de heteroduplex (*heteroduplex recombination)*, combina as vantagens dos métodos *in vitro* e *in vivo* de recombinação e não requer amplificação por PCR do produto recombinado, sendo desta forma, um método mais apropriado para a recombinação de longas seqüências de DNA.

O mecanismo de reparo independente utilizado por algumas células em regiões de *mismatches* está representado esquematicamente na Figura 2.17. O reparo independente de cada um dos *mismatches* pode originar seqüências idênticas às originais (Figura 2.17 (f)) ou seqüências recombinadas (Figura 2.17 (g)) as quais podem ser vistas como o resultado da ocorrência de cruzamento entre os parentais.

Volkov et al. (1999) descrevem duas variantes do método para a recombinação de seqüências parentais homólogas, referenciadas aqui por variante A e B. Ambas iniciam-se com a inserção dos fragmentos parentais (Parental 1 e 2) em vetores de expressão iguais.

Na variante A, os vetores com os insertos referentes aos parentais 1 e 2 são linearizados pelo tratamento com enzimas de restrição específicas e logo após desnaturados. Em seguida, o produto resultante de ambas as reações (reação contendo vetor + parental 1 e reação contendo vetor + parental 2) são misturados e submetidos à temperatura adequada para que ocorra o pareamento entre os fragmentos[12]. A recombinação irá ocorrer quando moléculas originárias de reações distintas se parearem. Note que o pareamento entre moléculas originárias de uma mesma reação irá resultar em moléculas lineares enquanto que o pareamento entre moléculas originárias de reações distintas resultará em moléculas circulares. A Figura 2.18, adaptada de Volkov et al. (1999), representa esquematicamente como ocorre a recombinação entre parentais na variante A.

Uma vez obtidas as moléculas circulares heteroduplexes, estas são utilizadas para a transformação de bactérias[13]. Uma vez transformadas, as bactérias irão crescer em colônias e o reparo das mutações, como apresentado pela Figura 2.17, poderá ocorrer.

Como comentado pelos autores, as primeiras tentativas de geração de bibliotecas de seqüências recombinantes a partir desta metodologia (variante A) resultaram em um baixo número de cruzamentos entre os parentais, ou seja, em uma baixa eficiência de remontagem. Um dos fatores responsáveis por essa baixa eficiência na recombinação pode estar relacionado à presença de *nicks* (regiões de fita simples, ou simplesmente *gaps*) nos heteroduplexes circulares

[12] Temperatura de *annealing* ou temperatura de pareamento.
[13] Após a inserção de um vetor (recombinante ou não) em uma bactéria, diz-se que a bactéria foi transformada.

resultantes (veja Figura 2.18 (c)), os quais são devido à fragmentação enzimática. Na tentativa de minimizar este problema, os heteroduplexes resultantes podem ser tratados com DNA ligase.

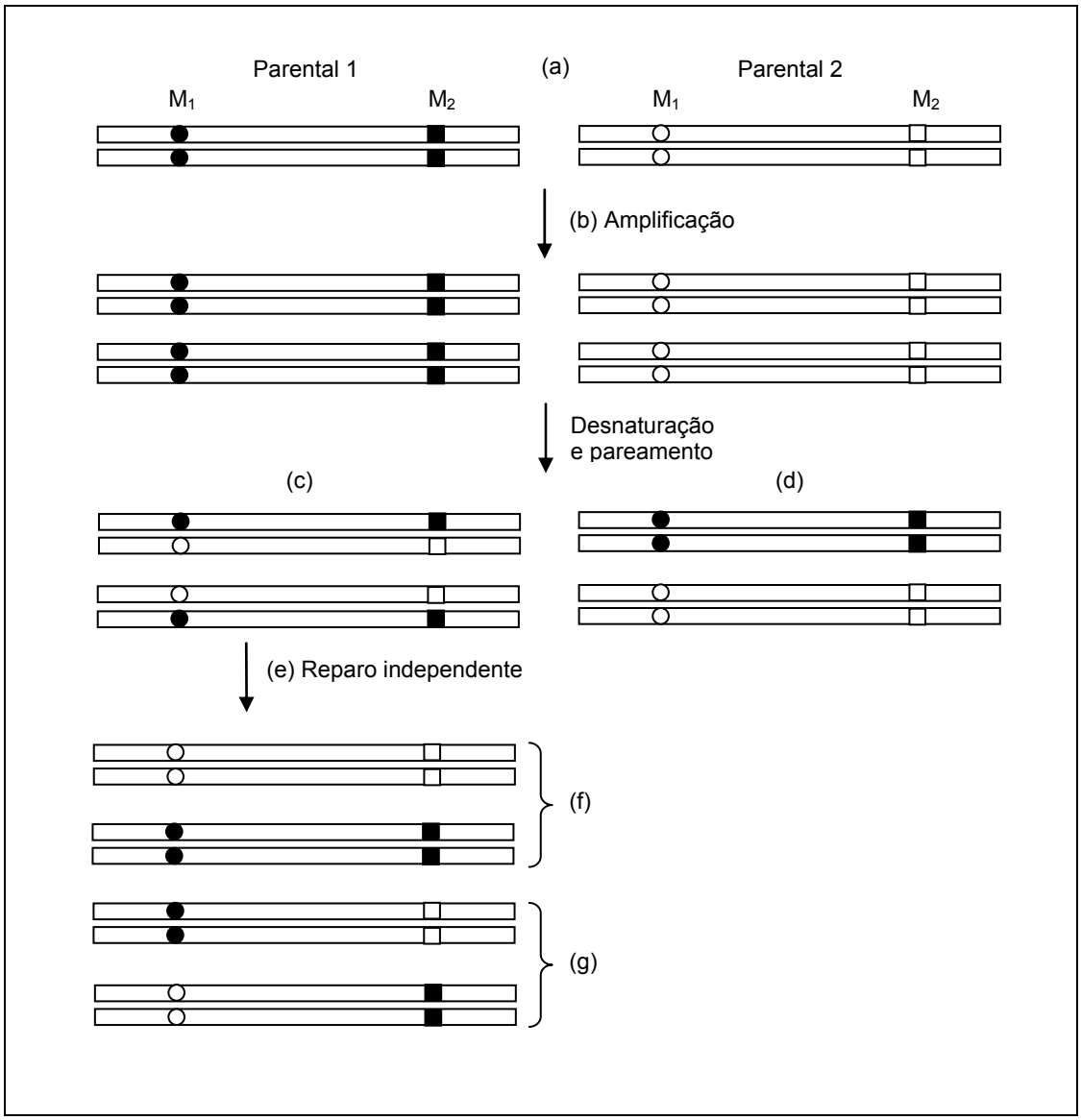


Figura 2.17. Mecanismo de reparo independente. (a) As seqüências parentais 1 e 2 diferem em dois pares de bases M_1 e M_2 os quais estão representados, respectivamente, por círculos e quadrados pretos no Parental 1 e por círculos e quadrados brancos no Parental 2. (b) Amplificação dos parentais. (c) e (d) Seqüências resultantes após a desnaturação e o pareamento. (c) Seqüências pareadas com *mismatches*. (d) Seqüências perfeitamente pareadas. (e) Reparo independente das seqüências com *mismatches* ((c)). (f) Seqüências resultantes do reparo que se tornaram idênticas aos parentais. (g) Seqüências resultantes do reparo, as quais podem ser vistas como o resultado de um cruzamento entre os parentais.

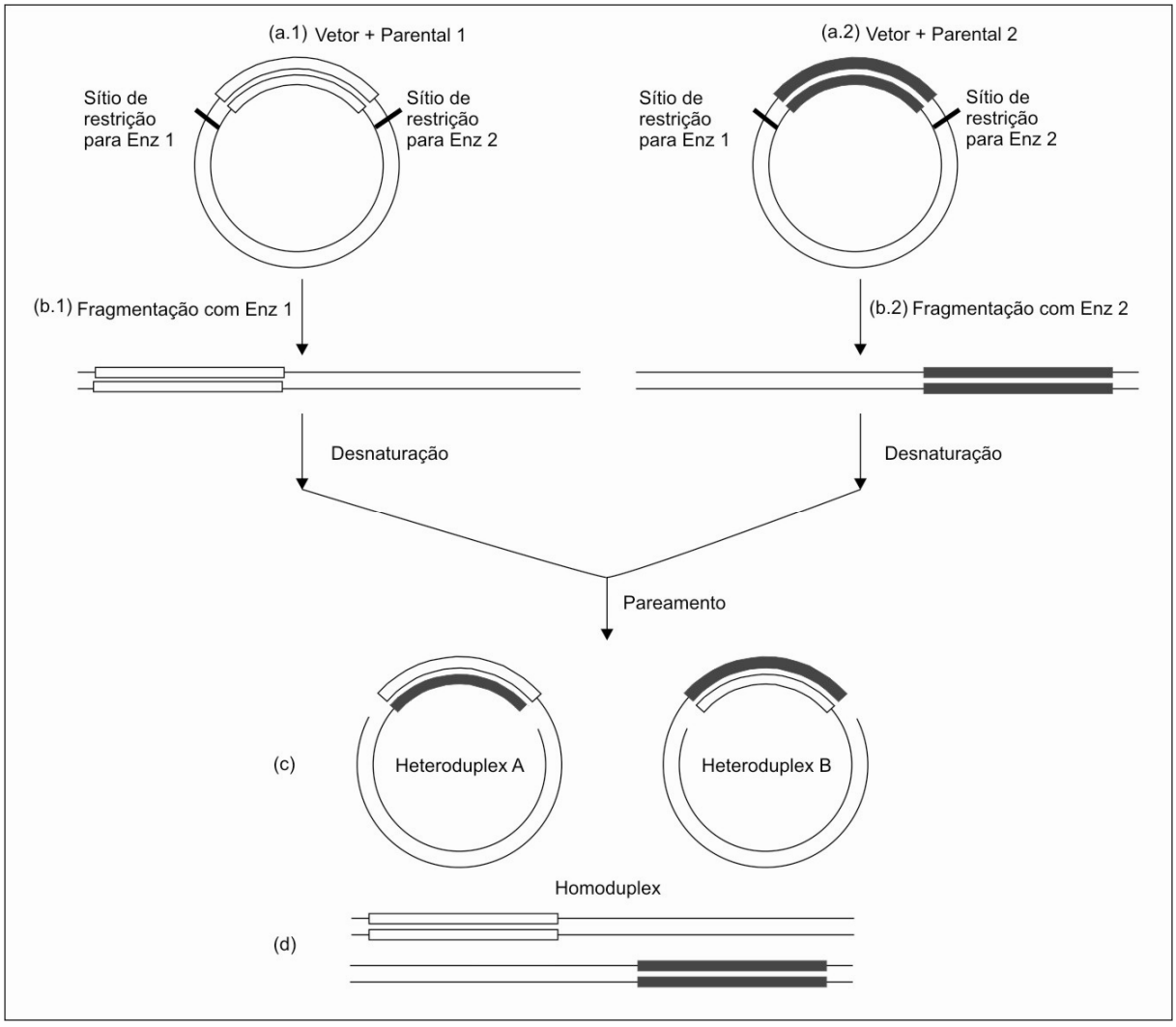


Figura 2.18. *Heteroduplex recombination*, variante A. (a.1) e (a.2) As seqüências parentais são inseridas em vetores de clonagem iguais em duas reações distintas. (b.1) e (b.2) Cada uma das reações é tratada com uma enzima de restrição específica; Enz 1 para reação (a.1) e Enz 2 para a reação (a.2), resultando na linearização das moléculas inicialmente circulares. Em seguida, o produto de ambas as reações é misturado, desnaturado e submetido a temperaturas ideais para que ocorra o pareamento entre os fragmentos. Como resultando deste pareamento, quatro moléculas distintas podem ocorrer, sendo estas: (c) Heteroduplexes e circulares ou (d) Homoduplexes e lineares. Apenas moléculas circulares podem ser eficientemente utilizadas para transformar bactérias.

Uma segunda variante (B) foi descrita com o objetivo de aumentar a eficiência na formação de heteroduplexes remontados e eliminar os possíveis problemas causados pela presença dos *nicks*. Nesta variante, apenas as seqüências parentais são amplificadas por reação de PCR. O produto da amplificação é então desnaturado, remontado e novamente inseridos em vetores de clonagem. Note que nessa variante, a remontagem depende apenas do tamanho das seqüências parentais e de sua identidade, e não mais do pareamento entre as seqüências originárias dos vetores, como na variante A e, além disso, desta forma não é mais possível a ocorrência de *nicks*

nas seqüências circulares resultantes. A Figura 2.19, adaptada de Volkov et al. (1999), ilustra esta variante (denominada variante B) da metodologia de *heteroduplex recombination*.

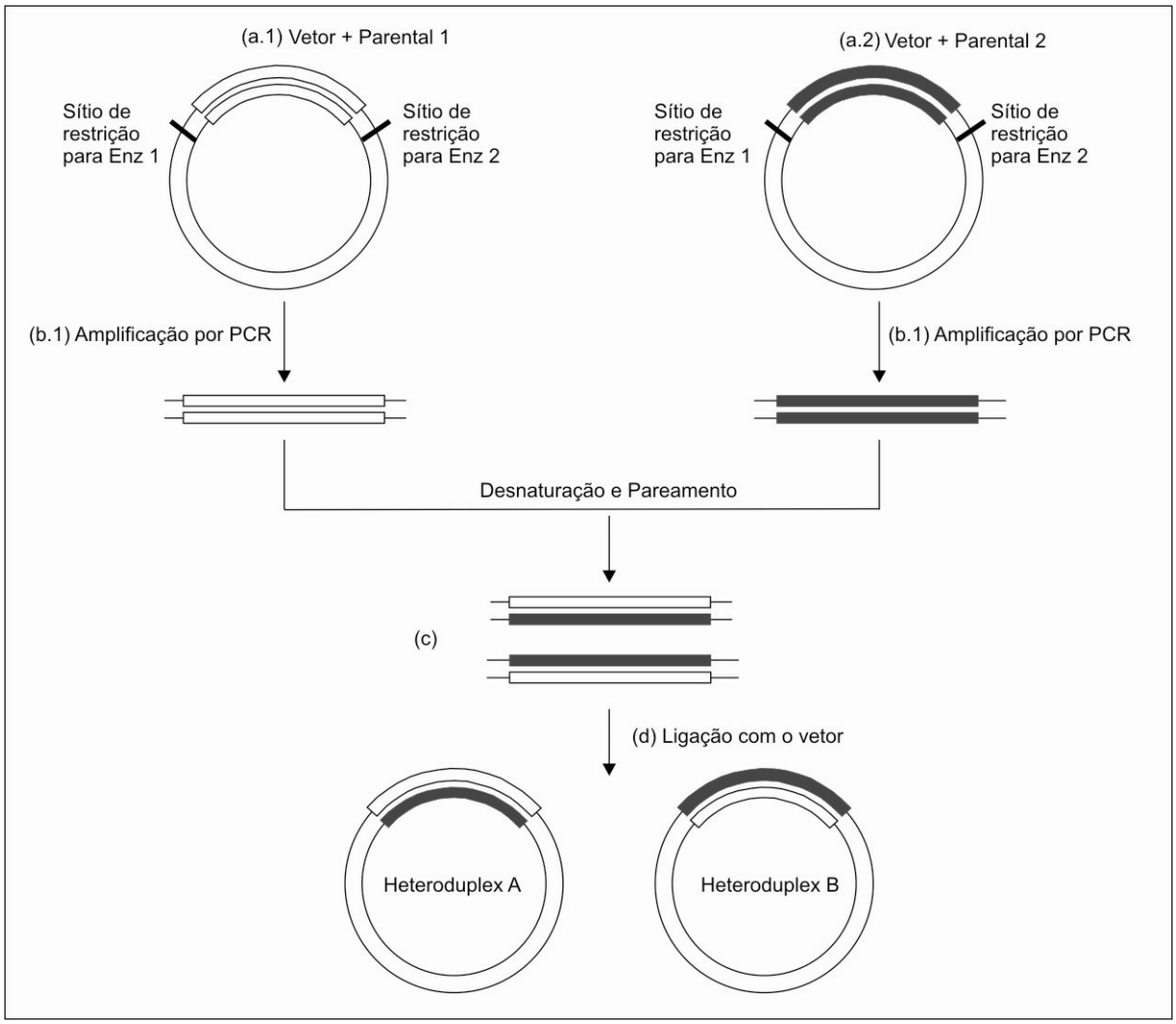


Figura 2.19. *Heteroduplex recombination*, variante B. (a.1) e (a.2) Os fragmentos parentais são inseridos em vetores de expressão iguais em duas reações distintas. (b.1) e (b.2) Reações de amplificação por PCR são executadas para que apenas as seqüências parentais aumentem em número. Em seguida, apenas os fragmentos amplificados são selecionados. (c) Os fragmentos purificados são desnaturados e remontados, resultando, possivelmente, em fragmentos heteroduplexes. (d) Os fragmentos remontados, assim como os vetores que irão receber estes fragmentos, são digeridos com enzimas de restrição e remontados em uma mesma molécula. Os heteroduplexes circulares estão prontos para serem utilizados na transformação de bactérias.

É certo que a variante B também produzirá grandes quantidades de fragmentos remontados idênticos aos fragmentos originais (homoduplexes). A fim de minimizar a ocorrência deste tipo de fragmentos, é possível amplificar apenas fitas opostas de cada um dos complexos (vetor + parental) antes de dar início ao processo de fragmentação enzimática, assim o pareamento ocorrerá necessariamente entre fitas originárias de parentais distintos.

2.13 SCRATCHY

A principal limitação dos métodos de recombinação que não levam em consideração a homologia entre parentais, como, por exemplo, o ITCHY é que eles são capazes apenas de produzir seqüências híbridas contendo um único cruzamento entre os parentais. Este fato limita o potencial de diversidade das seqüências resultantes. Na maioria dos casos de DNA *shuffling* reportados na literatura, múltiplos cruzamentos entre dois ou mais parentais foram necessários para que seqüências com propriedades melhoradas em relação aos parentais fossem obtidas (STEVENSON; BENKOVIC*,* 2002).

A fim de eliminar as limitações da metodologia ITCHY, ou seja, com o objetivo de produzir seqüências resultantes de múltiplos cruzamentos entre os parentais sem que o processo seja totalmente dependente da homologia entre estes, Lutz e coraboradores (LUTZ et al., 2001) utilizaram seqüencialmente as metodologias ITCHY e DNA *shuffling* a fim de criar seqüências com múltiplos cruzamentos entre dois parentais que possuem baixa homologia entre suas seqüências de DNA. Esta nova metodologia é chamada SCRATCHY.

Inicialmente, as duas seqüências parentais são submetidas ao ITCHY para que um conjunto de seqüências resultantes de um único cruzamento entre elas e independente de homologia seja obtido. Note que esses cruzamentos podem ocorrer em qualquer posição entre as seqüências e que os fragmentos resultantes podem ser de qualquer tamanho. As seqüências resultantes são em seguida submetidas ao *shuffling* para que os cruzamentos pré-existentes sejam recombinados, resultando assim em seqüências com múltiplos cruzamentos. A Figura 2.20, adaptada de Stevenson e Benkovic (2002), representa esquematicamente a metodologia SCRATCHY.

Segundo os autores, a união dessas duas técnicas resulta em uma biblioteca de híbridos cuja diversidade de suas seqüências é bem maior do que a diversidade gerada apenas pelo ITCHY ou pelo DNA *shuffling*.

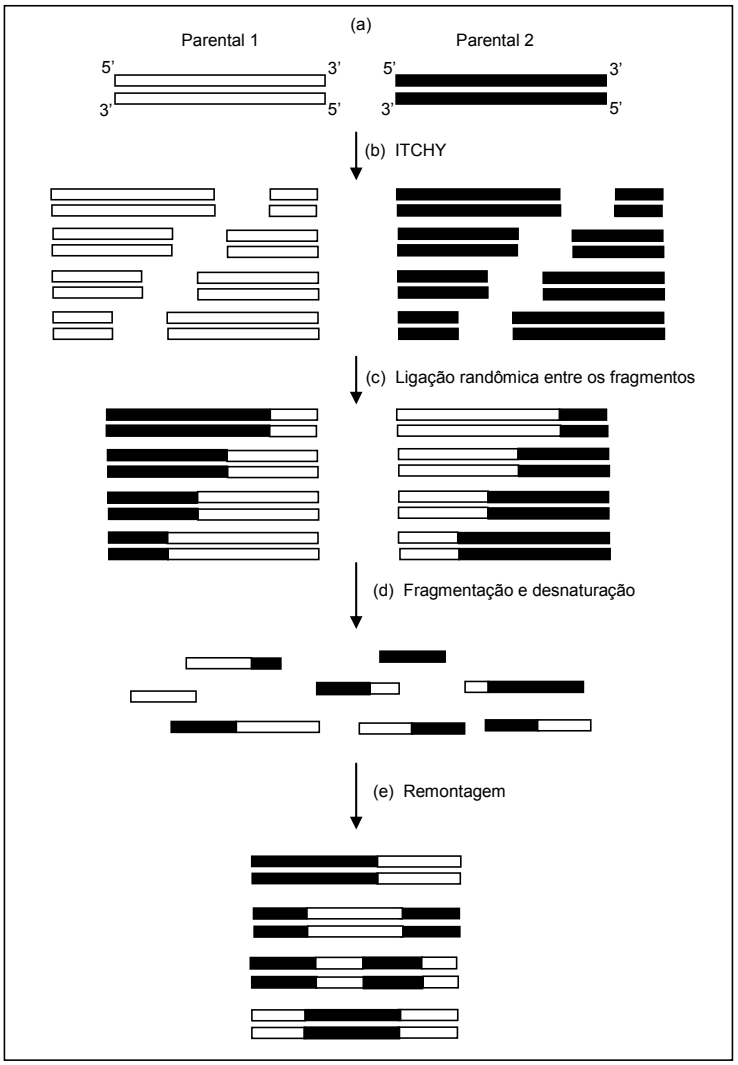


Figura 2.20. Representação esquemática da metodologia SCRATCHY. (a) Seqüência Parental 1 e Parental 2. (b) Fragmentos truncados em ambas as direções dos parentais (5'→ 3' e 3'→ 5') resultantes do ITCHY. (c) Ligação randômica entre os fragmentos resultantes. (d) Fragmentação e desnaturação dos fragmentos. (e) Remontagem dos fragmentos segundo o protocolo de DNA *shuffling*.

2.14 Considerações finais

Ao longo dos anos, diversas técnicas de Evolução Molecular Direta (EMD) foram desenvolvidas como, por exemplo, as descritas neste capítulo. Algumas técnicas surgiram na tentativa de aumentar a eficiência de técnicas anteriores, como é o caso, por exemplo, da técnica *Single Stranded* DNA que, por meio da utilização de apenas uma das fitas da molécula de DNA, tenta diminuir a formação de moléculas do tipo homoduplex em relação a experimentos de DNA *shuffling* e *Family shuffling*. Outras técnicas, entretanto, são específicas para moléculas com determinadas características como é o caso do *Restriction fragment shuffling*, uma vez que, em experimentos deste tipo, a molécula de DNA será fragmentada por enzimas de restrição apenas

em pontos determinados (sítios de restrição para as enzimas utilizadas). Se as moléculas a serem submetidas a experimentos de Evolução Molecular Direta compartilham alto grau de similaridade entre suas seqüências, técnicas cuja recombinação entre os parentais é baseada na homologia entre os fragmentos, como o caso do DNA *shuffling*, *Random-priming recombination*, StEP e ITCHY são mais adequadas. Caso contrário, isto é, se as seqüências a serem recombinadas são pouco similares, técnicas de recombinação como ITCHY são mais adequadas.

Devido à grande aplicabilidade das técnicas de Evolução Molecular Direta em diversas áreas, como por exemplo, famacêutica, médica e industrial, técnicas de EMD estão em constante evolução (BRANNIGAN; WILKINSON, 2002). Dentre algumas das técnicas mais recentes podem ser citadas: TriNEx (BALDWIN et al., 2008) que gera diversidade molecular por meio da substituição randômica de triplas de nucleotídeos; NexT DNA *shuffling* (MÜLLER et al., 2005) que, ao invés de realizar a fragmentação das seqüências parentais utilizando-se de enzimas como a DNase I, descreve a amplificação de trechos randômicos das seqüências parentais para originar os fragmentos a serem submetidos aos ciclos de PCR; *Site-saturation mutagenesis* (CHELLISERRYKATTIL; ELLINGTON, 2004), que permite a criação de bibliotecas de seqüências contendo todas as possíveis mutações em uma ou mais posições específicas de uma seqüência; e *Genome Shuffling* (ZHANG et al., 2002), que descreve a utilização de genomas inteiros ao invés de genes ou fragmentos de DNA em experimentos de DNA *shuffling*. Lutz e Patrick (2004) apresentam uma revisão de um conjunto de outras novas técnicas de Evolução Molecular Direta

Rubin-Pitel e Zhao (2006) apresentam uma revisão de diversos trabalhos que utilizaram diferentes técnicas de evolução molecular direta a fim de alterar propriedades de enzimas de interesse industrial tais como atividade, seletividade, especificidade, estabilidade e solubilidade.

.

3 Capítulo

Revisão de Quatro Modelos para o Processo de DNA *shuffling*

"Siga os bons e aprenda com eles."
Provérbio Chinês

3.1 Introdução

Como comentado anteriormente, a técnica de DNA *shuffling* tem se mostrado uma ferramenta eficiente para a evolução molecular direta de organismos de interesses comerciais, industriais, farmacológicos, agrícolas entre outros ((ROSIC et al., 2007), (GAFVELIN et al., 2007), (LOCHER et al., 2005), (KEENAN, et al., 2004), (CASTLE, et al., 2004), (COSTA, 2004), (STEMMER; HOLLAND, 2003)). Embora haja um maior esforço no sentido de mudar a atividade, especificidade e a estabilidade de enzimas importantes para diversos processos industriais, esta técnica também vem sendo utilizada para melhorar a estabilidade de vetores virais e de anticorpos.

Em seu trabalho, Costa (2004) descreve o *shuffling* entre as proteínas Canacistatina I (proveniente da cana-de-açúcar) e Orizacistatina I (proveniente do arroz). Essas proteínas atuam como inibidoras de protease, participando assim do mecanismo de defesa da planta. O objetivo do DNA *shuffling* foi o de produzir genes recombinantes que codificassem para um inibidor de protease mais eficaz do que os parentais. Um clone recombinante, denominado A10PL3, foi identificado e os testes de atividade inibitória mostraram que o clone A10PL3 teve um aumento

de aproximadamente 8 vezes na sua atividade inibitória com relação a uma protease humana, a catepsina B, quando comparado com os parentais.

O sucesso ou não da técnica de DNA *shuffling* pode ser medido pelo número de moléculas heteroduplexes encontradas na biblioteca resultante, uma vez que estas podem representar moléculas cuja funcionalidade é melhorada em relação aos parentais. Propor otimizações para o DNA *shuffling* é uma tarefa desafiadora devido às dificuldades práticas inerentes bem como à complexidade das reações envolvidas no processo, as quais, segundo Maheshri e Schaffer (2003), dependem de inúmeros parâmetros, tais como:

- concentração, composição e complexidade das seqüências a serem remontadas;
- condições de fragmentação e tamanho dos fragmentos obtidos;
- condições da reação de PCR, onde ocorre a remontagem dos fragmentos, incluindo a temperatura de pareamento, tempo de extensão pela DNA polimerase e número de ciclos da reação de PCR.

Otimizações para as condições das reações envolvidas no DNA *shuffling* são, na maioria das vezes, determinadas empiricamente, envolvendo grande dispêndio de recursos humanos e materiais. Em contrapartida, podem ser encontradas na literatura propostas para a modelagem do processo, subsidiadas por conceitos estatísticos e físicos voltadas para a implementação computacional, com vistas a prover apoio e promover maior eficiência do processo. Particularmente, quatro modelos encontrados na literatura têm se mostrado eficientes sob certos aspectos e têm sido aceitos como possíveis modelos que fornecem informações sobre o processo e sobre o sucesso (ou não) de seus resultados:

- O modelo DRIVeR (proposto por Patrick et al. (2003));
- O moldelo ShuffIt (proposto por Maheshri e Schaffer (2003));
- O modelo eShuffle (proposto por Moore et al. (2001));
- O modelo proposto por Sun (1999).

Esses quatro modelos são revistos nas próximas subseções com o objetivo de evidenciar suas principais características, abrangência, adequabilidade, facilidade de implementação e, principalmente, as contribuições de cada um quanto à modelagem do DNA *shuffling*. As informações apresentadas e discutidas fornecem contexto e subsídio para a ferramenta/modelo propostos nesta Tese, e discutidos nos capítulos 4 e 5.

3.2 O modelo DRIVeR

O principal objetivo desta subseção é apresentar, descrever e discutir em detalhes o modelo DRIVeR (PATRICK et al., 2003), o qual utiliza um ferramental estatístico para estimar o número de variantes distintas em bibliotecas resultantes de um processo de DNA *shuffling*, bem como o número médio de cruzamentos existentes nestas variantes.

3.2.1 Introdução

O modelo DRIVeR foi implementado como um programa em Fortran 77 e está disponível para *download*[14], juntamente com dois outros programas GLUE e PEDEL. Muito embora esta seção focalize apenas o DRIVeR, a descrição do tipo de problema que cada um dos modelos (e sua respectiva implementação) busca resolver é dada a seguir.

GLUE

Problema: Dada uma biblioteca L de tamanho L, que compreende seqüências de comprimento N pares de bases, tal que cada seqüência é escolhida aleatoriamente de um conjunto V de tamanho V de variantes equiprováveis, deseja-se calcular o número esperado de seqüências distintas em L, ou seja, qual a completudo de L. GLUE calcula também o tamanho que uma biblioteca deve ter para garantir uma desejada completude.

PEDEL (*Program for Estimating Diversity in Error-prone PCR Libraries*)

Problema: Dada uma biblioteca L de tamanho L, compreendendo variantes de uma seqüência de N pares de base, nas quais pontos de mutação foram introduzidos, deseja-se calcular o número esperado de seqüências distintas em L.

DRIVeR (*Diversity Resulting from In Vitro Recombination*)

Problema: Dada uma biblioteca L de tamanho L, gerada por recombinações aleatórias de dois genes quase idênticos, que diferem em apenas um pequeno número conhecido de posições de pares de bases (ou códons), deseja-se calcular o número esperado de variantes distintas em L e o número médio de cruzamentos nas variantes.

[14] no endereço www.bio.cam.ac.uk/~blackburn/stats.html

3.2.2 Contextualização

O DRIVeR é um modelo com base estatística, utilizado para estimar a diversidade representada em bibliotecas geradas por meio da recombinação de duas seqüências parentais altamente homólogas, que diferem em apenas poucas (no máximo 20) posições de base ou aminoácidos, caso as seqüências em questão sejam de DNA ou proteínas, respectivamente.

Como discutido no Capítulo 2, o uso de recombinação de seqüências de DNA para evolução direta teve início com o desenvolvimento da técnica de DNA *shuffling* ((STEMMER, 1994a) e (STEMMER, 1994b)), que se baseia em uma reação similar à reação de PCR para a remontagem de seqüências parentais randomicamente fragmentadas. A técnica foi estendida para lidar com uma família de genes de diversas espécies simultaneamente (ver (CRAMERI et al., 1998)). Podem ser encontrados na literatura outros protocolos para evolução molecular direta tais como o RACHITT (ver (COCO et al., 2001)), *shuffling* de fitas simples de DNA (ver (KIKUCHI et al., 2000)) e StEP (ver (ZHAO et al., 1998)). Como comentado por Moore e Maranas (2004, p. 264), "Em todos esses métodos, a geração de cruzamento é dependente do pareamento e da extensão de fragmentos complementares de fita simples, originárias de seqüências parentais distintas (i.e., formação heteroduplex), processo que é tendencioso com relação às posições do cruzamento, que promovem extensões de modo a restabelecer seqüências originais. Isso, por sua vez, tende a promover o desenvolvimento de bibliotecas de DNA combinatoriamente tendenciosas ou, até mesmo, bibliotecas sem qualquer diversidade adicional sobre os parentais. Geralmente, uma severa tendência à remontagem das seqüências parentais (i.e., seqüências remontadas sem a ocorrência de recombinações) é observada quando seqüências com menos de 60% de identidade são recombinadas por meio de protocolos baseados em pareamento".

No modelo DRIVeR, a formação de fragmentos heteroduplex, ou seja, o pareamento entre fragmentos originários de parentais distintos, é descrito como um evento de cruzamento (*crossovers*) entre os parentais, e uma seqüência é dita variante caso ela seja formada como conseqüência da ocorrência de um ou mais cruzamentos.

Lembrando o que foi descrito na Seção 2.6.3, no processo de *shuffling*, após a seleção e fragmentação dos parentais, o conjunto de fragmentos resultantes é submetido aos ciclos de PCR sem a adição de *primers* para que os fragmentos sejam remontados e seqüências com o mesmo tamanho dos parentais sejam obtidas. Um ciclo de PCR consiste de três etapas: desnaturação, pareamento e extensão. O evento de cruzamento é verificado na etapa de pareamento, pois é

neste momento que os fragmentos que compartilham regiões complementares se unem para dar origem, após a extensão, a um único fragmento.

3.2.3 O cruzamento no modelo DRIVeR

No que segue, é discutido, por meio de um exemplo, como o DRIVeR aborda, conceitualiza e modela o evento de cruzamento. Sejam as duas seqüências parentais A e B apresentadas na Figura 3.1 (a), que diferem entre si em dois pares de bases. Vale lembrar que esses pontos de diferença entre as seqüências são também chamados de pontos de mutação entre elas. Sejam os fragmentos resultantes da fragmentação enzimática pela enzima DNase I os mostrados na Figura 3.1 (b) os quais, depois de desnaturados, resultam em fragmentos de fita simples (Figura 3.1 (c)). Para exemplificar a ocorrência de um evento de cruzamento, considere o pareamento apresentado na Figura 3.1 (d) no qual um dos fragmentos é originário do Parental A e o outro do Parental B. Após a extensão dos fragmentos pareados tem-se um fragmento heteroduplex (Figura 3.1 (e)).

A seqüência resultante da extensão (Figura 3.1 (e)) acumula dois pares de bases, destacados em preto e branco, que se encontravam originalmente nos parentais A e B, respectivamente. Este evento caracteriza a ocorrência de um cruzamento entre os parentais. É importante observar também que a possibilidade desse cruzamento (ou recombinação) ocorrer está relacionada ao fato que as mutações entre os parentais foram separadas em fragmentos distintos, pela ação da enzima DNase I.

Como visto no Capítulo 1, uma vez que as duas fitas de uma molécula de DNA são complementares, a molécula pode ser representada apenas por uma das fitas e, por convenção, utiliza-se a fita de orientação 5'→ 3'. Para simplificar ainda mais a notação utilizada nesta seção, considere as duas seqüências parentais A e B da Figura 3.1. Elas serão, deste ponto em diante, representadas apenas por uma de suas fitas; além disso, as bases idênticas entre as seqüências serão omitidas da representação e a localização das mutações entre os parentais será assinalada por símbolos distintos em cada um dos parentais, como mostra a Figura 3.2.

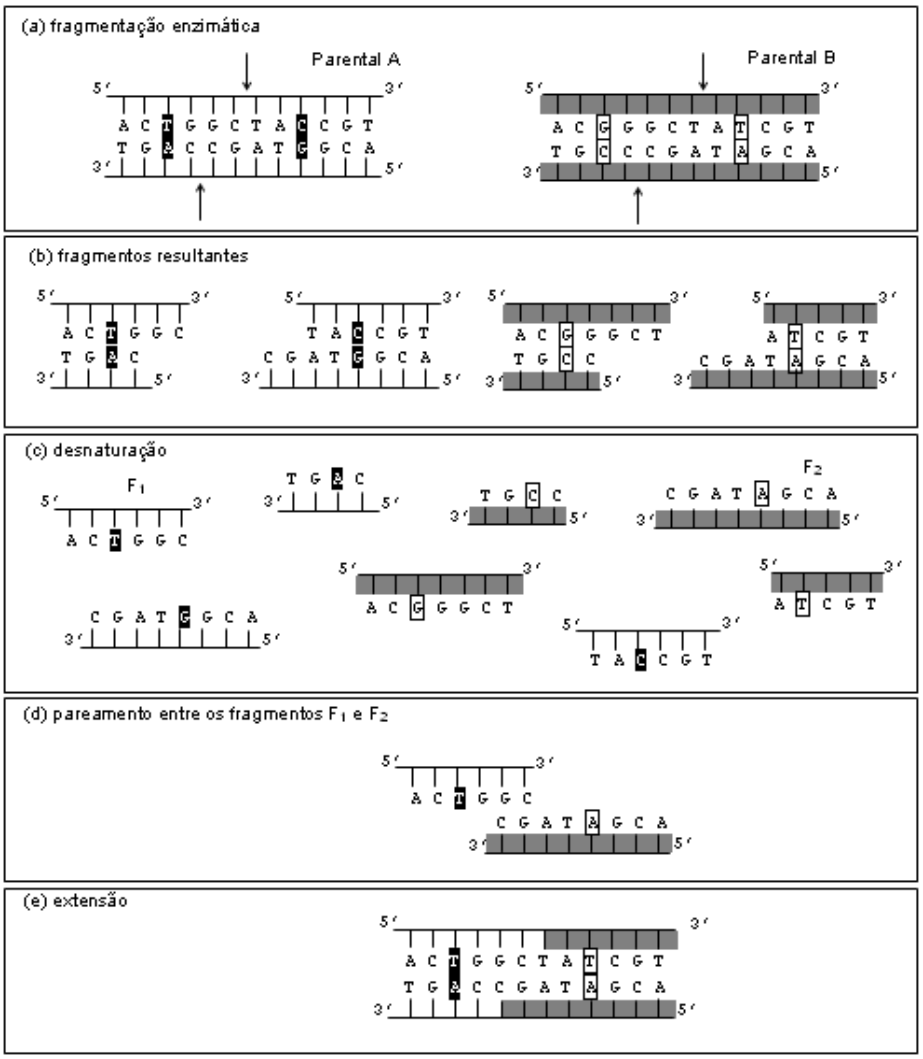


Figura 3.1. Representação do evento de cruzamento entre dois parentais. (a) Parentais A e B que diferem entre si em dois pares de bases identificados no Parental A com fundo preto e no Parental B com fundo branco. Em (a) as setas indicam as posições onde a molécula de DNA será cortada pela enzima de fragmentação. (b) Fragmentos resultantes da ação da enzima. (c) Desnaturação dos fragmentos. (d) Pareamento entre os fragmentos F_1 e F_2, os quais são originários de parentais distintos. (e) Extensão dos fragmentos pareados resultando em um fragmento heteroduplex.

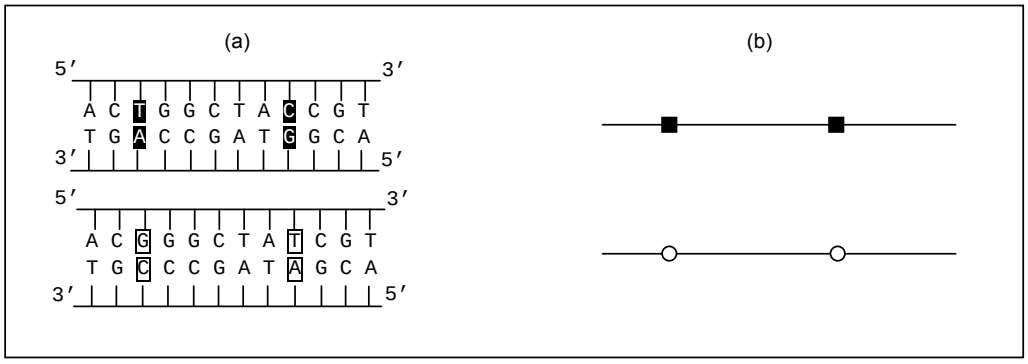


Figura 3.2. Seqüências parentais que diferem em dois pares de bases. (a) Representação por fita dupla. (b) Representação simplificada contendo apenas uma fita e as bases que diferenciam uma seqüência da outra, representadas pelos símbolos ■ e o.

A fragmentação pela enzima DNase I é um processo randômico, de forma que não é possível prever em qual posição, de cada uma das fitas de DNA, ocorrerá o corte. Contudo, considerando duas mutações consecutivas de um parental, apenas duas situações distintas podem resultar da fragmentação randômica: ou o corte produzido pela enzima irá se localizar entre essas duas mutações, ou em um ponto qualquer fora deste intervalo. Apenas quando o corte produzido se localiza entre as duas mutações consecutivas é que se caracteriza a situação em que as mutações poderão ser recombinadas em um único fragmento. A Figura 3.3 ilustra as duas situações descritas, considerando que as moléculas parentais diferem em apenas dois pares de bases.

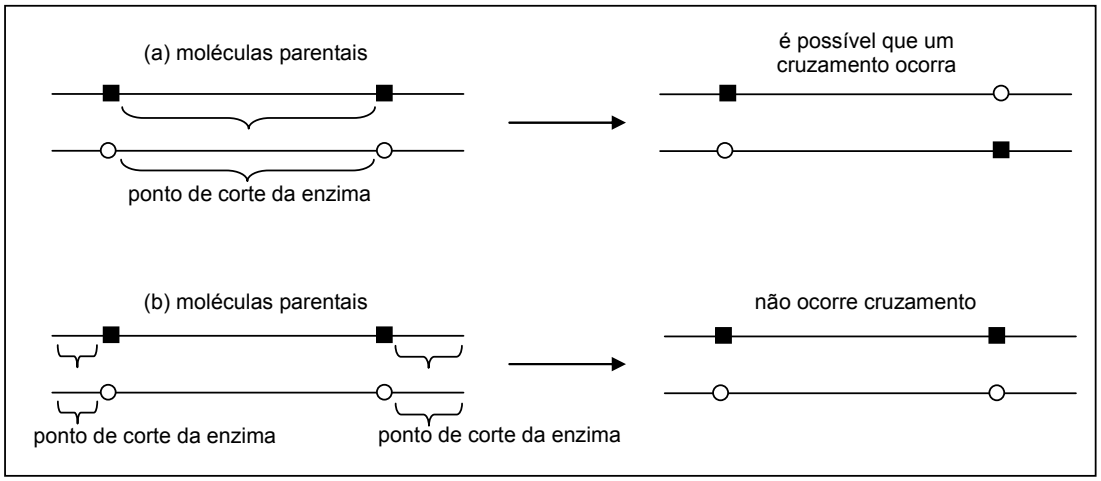


Figura 3.3. Possíveis pontos de corte da enzima DNase I. (a) Caso a enzima produza cortes entre as duas mutações consecutivas, há chance do evento de cruzamento ocorrer entre os fragmentos resultantes. (b) Caso os cortes produzidos pela enzima não estejam localizados entre as mutações, a possível recombinação dos fragmentos não irá gerar um heteroduplex, uma vez que as bases distintas entre as moléculas parentais permaneceram em um mesmo fragmento.

3.2.4 Descrição do modelo DRIVeR

Como discutido previamente, o objetivo da evolução molecular *in vitro* pela técnica de DNA *shuffling* é a produção de moléculas variantes resultantes de um ou mais cruzamentos, as quais podem representar moléculas com características novas ou melhoradas em relação às moléculas parentais. O modelo DRIVeR considera dois fatores como críticos para a obtenção de seqüências variantes na biblioteca de DNA *shuffling*:

– número de heteroduplexes formados durante o processo de pareamento entre os fragmentos;
– distância entre os pares de bases distintos e consecutivos em uma seqüência, uma vez que pares de bases distintos próximos uns dos outros são menos prováveis de se

recombinarem durante o *shuffling* quando comparados àqueles que se encontram mais distantes.

Esses dois fatores estão fortemente relacionados entre si, uma vez que a formação de heteroduplex depende dos cortes realizados pela fragmentação enzimática. Como ilustrado na Figura 3.3, os cortes realizados pela DNase I devem se localizar entre os pares de bases distintos para que esses possam ser remontados em um único fragmento maior. Como a fragmentação é um evento randômico, o modelo DRIVeR avalia qual a probabilidade de que o corte produzido pela enzima se localize na região compreendida entre duas mutações consecutivas. O modelo assume que os pontos de cortes da enzima estão distribuídos ao longo das moléculas parentais segundo a Distribuição de Poisson (FELLER, 1957). No que segue é apresentada a descrição formal do modelo, como feita em Patrick et al. (2003).

Sejam duas seqüências parentais homólogas S_1 e S_2, ambas de tamanho N, que diferem entre si em M pares de bases, representadas respectivamente por A_i e $A_i^{'}$, para $1 \leq i \leq M$. Ambas as seqüências podem ser representadas apenas pelas bases distintas existentes entre elas, ou seja, $S_1 = A_1 A_2 ... A_M$ e $S_2 = A_1^{'} A_2^{'} ... A_M^{'}$, uma vez que as demais bases são iguais. Se considerarmos a ocorrência de todos os possíveis cruzamentos entre os pares de bases distintos e consecutivos das seqüências parentais, tem-se um total de $2^M - 2$ variantes distintas possíveis, sendo que cada uma das variantes distintas k pode ser representada por $D_k = A_1^k A_2^k ... A_M^k$ tal que $A_i^k = A_i$ ou $A_i^{'}$. A Figura 3.4 mostra um exemplo considerando duas seqüências parentais A e B, que diferem em M = 5 pares de bases. A figura mostra também a representação da forma geral de toda variante k, resultante da ocorrência de todos os possíveis cruzamentos entre A e B.

Essa forma de representação de uma variante k ($D_k = A_1^k A_2^k ... A_M^k$) é descrita em termos das posições onde ocorrem as mutações entre os parentais. Porém, como o modelo pretende estimar a probabilidade de ocorrência de cruzamento em uma determinada região, no caso a região compreendida entre duas mutações consecutivas, uma forma de representação mais conveniente é necessária.

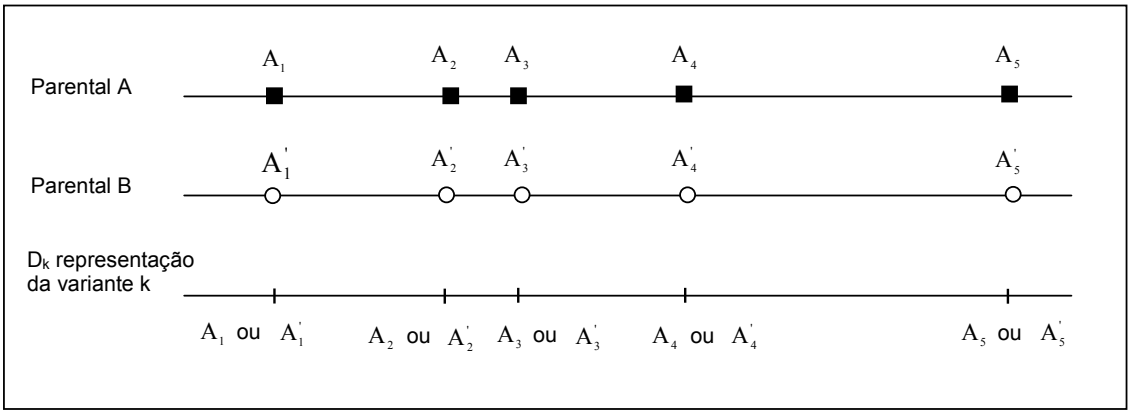


Figura 3.4. Parentais A e B, nos quais cada uma das M = 5 bases distintas entre eles estão assinaladas com um quadrado preto no Parental A e com um círculo branco no Parental B e a representação da forma geral de uma variante k do total de $2^M - 2$ variantes possíveis, resultante da ocorrência de todos os possíveis cruzamentos entre os parentais A e B.

A variante k (D_k) será então representada por uma seqüência binária $B_k = b_1^k b_2^k ... b_{M-1}^k$ na qual $b_i^k = 0$ ou $b_i^k = 1$, i = 1, 2..., M−1, caso tenha ocorrido um número par ou ímpar de cruzamentos entre as bases A_i^k e A_{i+1}^k, respectivamente. Note que a ocorrência de um único cruzamento entre duas bases distintas e consecutivas resulta exatamente na mesma variante caso aconteçam 3, 5, 7, 9,.... cruzamentos nesta mesma posição; o mesmo ocorre para qualquer número par de cruzamentos entre duas bases variantes consecutivas. A Figura 3.5 mostra um exemplo de uma variante k resultante da ocorrência de diversos cruzamentos entre duas seqüências parentais A e B**,** que diferem em M = 6 pares de bases. A figura apresenta também a representação binária B_k da variante resultante.

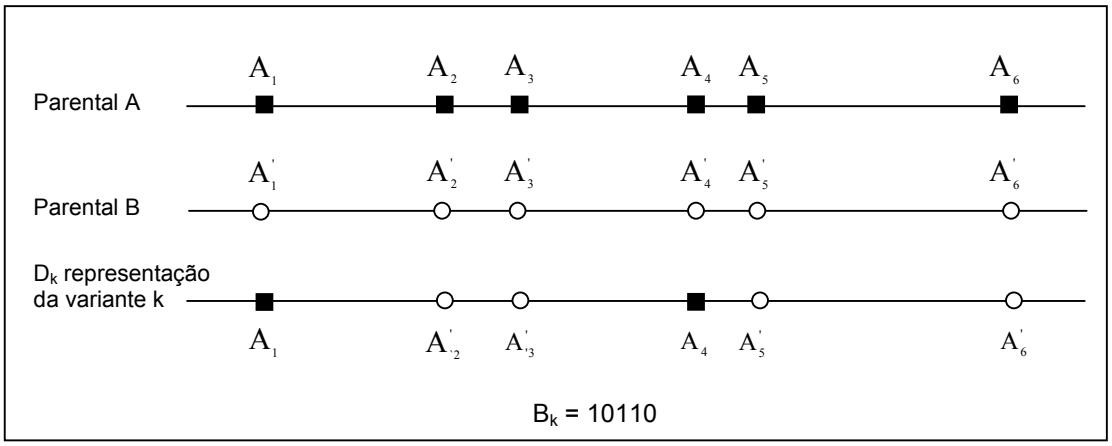


Figura 3.5. Duas seqüências parentais A e B e uma variante k resultante de cruzamentos entre os parentais. B_k é a representação binária da seqüência variante k e é função do número de cruzamentos entre as bases consecutivas que diferem entre si nos parentais.

É importante observar que algumas variantes podem ser representadas por uma mesma seqüência binária B_k. Considere por exemplo a variante $D_k = A_1 A_2 A_3' A_4 A_5' A_6' A_7'$ representada pela seqüência binária $B_k = 011100$. Seja D_k' a seqüência variante inversa a D_k, isto é, $D_k' = A_1' A_2' A_3 A_4' A_5 A_6 A_7$; sua representação binária é $B_k' = 011100$. Tem-se que B_k e B_k' são idênticas e equiprováveis, apesar de representarem variantes distintas. Essa observação é importante e será considerada quando do cálculo da probabilidade de ocorrência de todas as possíveis variantes em uma dada biblioteca. A Tabela 3.1 apresenta um exemplo de todas as $2^5 - 2 = 30$ variantes distintas possíveis resultantes dos cruzamentos entre dois parentais $S_1 = A_1 A_2 A_3 A_4 A_5$ e $S_2 = A_1' A_2' A_3' A_4' A_5'$, bem como a representação binária de cada uma delas, incluindo os parentais S_1 e S_2.

Suponha que o número de cruzamentos por seqüência resultante do *shuffling* seja representado pelo valor λ, e que λ << N (lê-se: λ é muito menor do que N, onde N é o tamanho das seqüências parentais). Assumindo que o número de cruzamentos por seqüência resultante segue a Distribuição de Poisson, a probabilidade P(x) de que uma seqüência resultante tenha exatamente x cruzamentos é dada pela eq. (3.1).

$$P(x) = \frac{e^{-\lambda}\lambda^x}{x!}, x = 0,1,2,... \tag{3.1}$$

O modelo assume também que cruzamentos não ocorrem na posição imediatamente seguinte a uma mutação, uma vez que esse ponto não favorece o pareamento entre dois fragmentos, já que este acontece apenas entre fragmentos que compartilham trechos complementares os quais servem para unir dois fragmentos (veja Figura 3.1 (d)). Desta forma, existem (N − M − 1) pontos possíveis de ocorrer cruzamentos entre as seqüências originais.

A notação n_i, para i = 0, 1, ..., M, representa o número de ligações base–base entre duas mutações consecutivas A_i e A_{i+1}; n_0 e n_M representam o número de ligações base–base antes da mutação A_1 e depois de A_M, respectivamente. Dessa forma, o valor n_i representa o número potencial de pontos em que cruzamentos podem ocorrer entre duas mutações consecutivas A_i e A_{i+1}.

Sejam os parentais:

A = GCGCTAGGAGATCAGTTGAGCTATCGG e

B = GCGTTAGGAGGTCAGTTGCGCTATAGG,

ambos de tamanho N = 27 e que diferem de M = 4 pares de bases. A Figura 3.6 apresenta os valores de n_i para os parentais A e B. Nesta figura, as bases distintas entre ambos os parentais foram substituídas pelo caractere A_i, $1 \leq i \leq M$.

Tabela 3.1. Representação de todas as possíveis variantes do cruzamento entre $S_1 = A_1A_2A_3A_4A_5$ e $S_2 = A'_1A'_2A'_3A'_4A'_5$.

Seqüência	Representação binária	Seqüência	Representação binária
$S_1 = A_1A_2A_3A_4A_5$	0000	$A'_1A'_2A_3A_4A_5$	0100
$S_2 = A'_1A'_2A'_3A'_4A'_5$	1111	$A_1A_2A'_3A'_4A'_5$	0100
$A_1A_2A_3A_4A'_5$	0001	$A_1A'_2A_3A'_4A'_5$	1110
$A_1A_2A_3A'_4A_5$	0011	$A'_1A_2A_3A'_4A'_5$	1010
$A_1A_2A'_3A_4A_5$	0110	$A_1A'_2A'_3A'_4A_5$	1001
$A_1A'_2A_3A_4A_5$	1100	$A'_1A_2A'_3A'_4A_5$	1101
$A'_1A_2A_3A_4A_5$	1000	$A'_1A'_2A'_3A_4A_5$	0010
$A_1A_2A_3A'_4A'_5$	0010	$A_1A'_2A'_3A'_4A'_5$	1000
$A_1A_2A'_3A_4A'_5$	0111	$A'_1A_2A'_3A'_4A'_5$	1100
$A_1A'_2A_3A_4A'_5$	1101	$A'_1A'_2A'_3A'_4A_5$	0001
$A'_1A_2A_3A_4A'_5$	1001	$A'_1A'_2A'_3A_4A_5$	0011
$A_1A_2A'_3A'_4A_5$	0101	$A'_1A'_2A_3A_4A'_5$	0101
$A_1A'_2A_3A'_4A_5$	1111	$A'_1A'_2A_3A'_4A_5$	0111
$A'_1A_2A_3A'_4A_5$	1011	$A'_1A'_2A_3A_4A'_5$	0101
$A_1A'_2A'_3A_4A_5$	1010	$A'_1A'_2A_3A'_4A_5$	0111
$A'_1A_2A'_3A_4A_5$	1110	$A'_1A'_2A_3A'_4A'_5$	0110

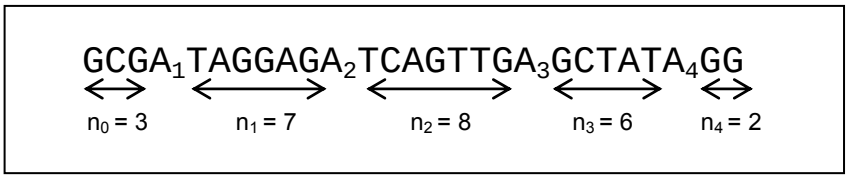


Figura 3.6. Valores de n_i para os parentais A e B.

Da Figura 3.6 tem-se que $\sum_{i=0}^{M} n_i = 3+7+8+6+2 = 26 = N-1$.

A probabilidade de existirem exatamente x cruzamentos (P(x)) entre as bases variantes A_i e A_{i+1} segue também a Distribuição de Poisson, sendo o número esperado de cruzamentos entre duas bases distintas e consecutivas dado por:

$$\frac{(n_i - 1)\lambda}{N - M - 1}$$

A probabilidade de ocorrer um número qualquer de cruzamentos entre A_i e A_{i+1} é dada pela multiplicação das probabilidades $P(b_i = 0)$ e $P(b_i = 1)$, as quais representam a probabilidade de ocorrer um número par ou ímpar de cruzamentos entre essas duas mutações consecutivas, respectivamente. Desta forma, tem-se[15]:

– Probabilidade de ocorrer um número par de cruzamentos entre duas mutações consecutivas A_i e A_{i+1}:

$$\begin{aligned} P(b_i = 0) &= P(x = 0) + P(x = 2) + P(x = 4) + \ldots \\ &= e^{-\frac{(n_i - 1)\lambda}{N-M-1}} * \cosh\left(\frac{(n_i - 1)\lambda}{N - M - 1}\right) \end{aligned} \tag{3.2}$$

– Probabilidade de ocorrer um número ímpar de cruzamentos entre duas mutações consecutivas A_i e A_{i+1}:

$$\begin{aligned} P(b_i = 1) &= P(x = 1) + P(x = 3) + P(x = 5) + \ldots \\ &= e^{-\frac{(n_i - 1)\lambda}{N-M-1}} * \sinh\left(\frac{(n_i - 1)\lambda}{N - M - 1}\right) \end{aligned} \tag{3.3}$$

– Probabilidade de ocorrer um número qualquer de cruzamentos entre duas mutações consecutivas A_i e A_{i+1}:

[15] cosh e sinh nas equações (3.2) e (3.3) representam, respectivamente, o cosseno e o seno hiperbólico, medidos em radianos.

$$P(b_i) = P(b_i = 0) * P(b_i = 1) \tag{3.4}$$

Assim, a probabilidade de uma dada variante k, representada pela seqüência binária B_k ocorrer, corresponde ao produtório dos valores $P(b_i = 0)*P(b_i = 1)$ para todo par A_i e $A_{i+1,}$ para i = 1, 2, ..., M−1. Este produtório é representado na eq. (3.5).

$$P(B_K) = \Pi_i^{M-1} P(b_i^k) \tag{3.5}$$

Pelo fato de duas seqüências inversas e equiprováveis serem mapeadas para um mesmo código binário, a divisão do produtório $P(B_k)$ por 2 fornece a probabilidade relativa, representada por Q_k, da ocorrência de cada uma das 2^M possíveis variantes k.

$$Q_K = \frac{P(B_k)}{2} \tag{3.6}$$

Desta forma, é possível calcular:

– A probabilidade de que uma dada seqüência da biblioteca não seja uma variante:

$$1 - Q_k$$

– A probabilidade de não existir nenhuma seqüência variante em uma biblioteca de tamanho L:

$$(1 - Q_k)^L$$

– A probabilidade de que uma variante k ocorra em uma biblioteca de tamanho L:

$$1 - (1 - Q_k)^L$$

Para uma biblioteca de tamanho L construída pelo *shuffling* entre duas seqüências parentais de comprimento N que diferem em M posições e uma taxa média de cruzamento λ, do total de 2^M variantes possíveis, o número de seqüências variantes esperadas nessa biblioteca, representado por C, é dada pela eq. (3.7).

$$C = \sum_{k=1}^{2^M} 1 - (1 - Q_k)^L \tag{3.7}$$

Uma consideração importante deve ser feita a respeito do valor de λ. Até o momento, foi assumido que o valor λ corresponde ao número real médio de cruzamentos observados em uma amostra de clones da biblioteca de *shuffling*, que foram seqüenciados e analisados. O número médio de cruzamentos observado em uma amostra, porém, pode não corresponder ao número

real de cruzamentos presentes na amostra. Esse fato se deve à ocorrência dos chamados cruzamentos silenciosos**.** Como comentado no Capítulo 2, um cruzamento é dito silencioso quando a seqüência resultante é idêntica a uma das seqüências parentais, mesmo que ela tenha sofrido um evento de cruzamento, ou seja, mesmo que essa seqüência seja formada por fragmentos de parentais distintos. Os cruzamentos silenciosos ocorrem quando o ponto de cruzamento está localizado em um intervalo no qual não existem bases variantes consecutivas. A Figura 3.7 mostra dois exemplos de possíveis pontos de cruzamentos que não produzem seqüências diferentes das originais e, por isso, caso ocorram, serão chamados de cruzamentos silenciosos.

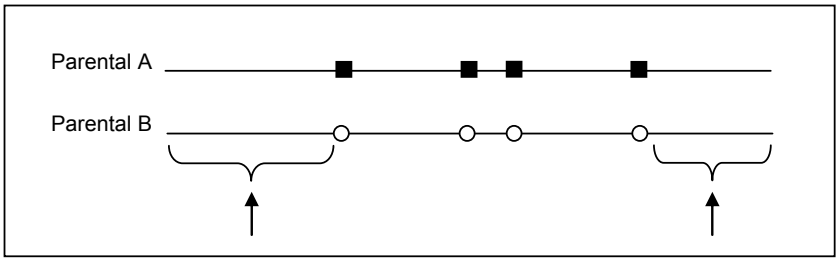


Figura 3.7. Exemplos de pontos de cruzamentos que podem gerar cruzamentos silenciosos.

Devido à potencialidade de ocorrência de cruzamentos silenciosos, o valor λ obtido do seqüenciamento e análise de uma amostra da biblioteca representa um valor observado, denotado por λ^{obs}. Porém, o valor utilizado para os cálculos descritos anteriormente deve, necessariamente, ser o valor real de λ, denotado por λ^{true}. Uma vez informado o valor de λ^{true}, o modelo calcula o valor de λ^{obs}. O valor de λ^{obs} é aproximado pela somatória, para todas as possíveis variantes distintas, da probabilidade de cada variante k ocorrer ($P(B_k)$) vezes o número de cruzamentos existentes nesta variante ($num\_cros(B_k)$), como mostra a eq. (3.8).

$$\lambda^{obs} = \sum_{k} P(B_k) * num\_cros(B_k) \tag{3.8}$$

Deve-se, por tentativa e erro, descobrir qual o valor de λ^{true} que resultará no valor λ^{obs}, e só então utilizar este valor λ^{true} para dar continuidade à utilização do DRIVeR.

Considere duas seqüências fictícias de tamanho N = 1.500 que diferem em M = 9 pares de bases, as quais estão distantes uma da outra por um intervalo fixo n_i = 100 pares de bases. A execução do programa DRIVeR para tamanhos crescentes de L e λ^{obs} permite verificar a influência destes dois parâmetros no número C de variantes esperadas na biblioteca resultante. Tais simulações são apresentadas no Gráfico 3.1.

Observa-se no Gráfico 3.1 que o número esperado de seqüências distintas C cresce com o aumento de L e λ^{obs}, e que, para valores suficientes grandes de L e λ^{obs}, o número de seqüências distintas esperadas alcança o número total de possíveis variantes na biblioteca que, para o caso do exemplo utilizado, é de $2^M = 2^9 = 512$. Porém, valores grandes para L e λ^{obs} nem sempre podem ser alcançados na prática, devido a limitações experimentais relativas ao número de cruzamentos observados na maioria dos experimentos de *shuffling*.

Utilizando o mesmo exemplo anterior, considere um experimento que evidencia a influência das distâncias que separam os pares de bases distintos e consecutivos existentes entre as duas seqüências a serem submetidas ao *shuffling*, no número de variantes esperadas na biblioteca. Sejam N = 1.500, M = 9, $\lambda^{true} = 8$ e os seguintes valores para os intervalos n_i entre os 9 pares de bases variantes: 100, 50, 25, 15, 10, 5, 2. A influência do parâmetro n_i sobre o número de variantes esperadas na biblioteca pode ser observada no Gráfico 3.2.

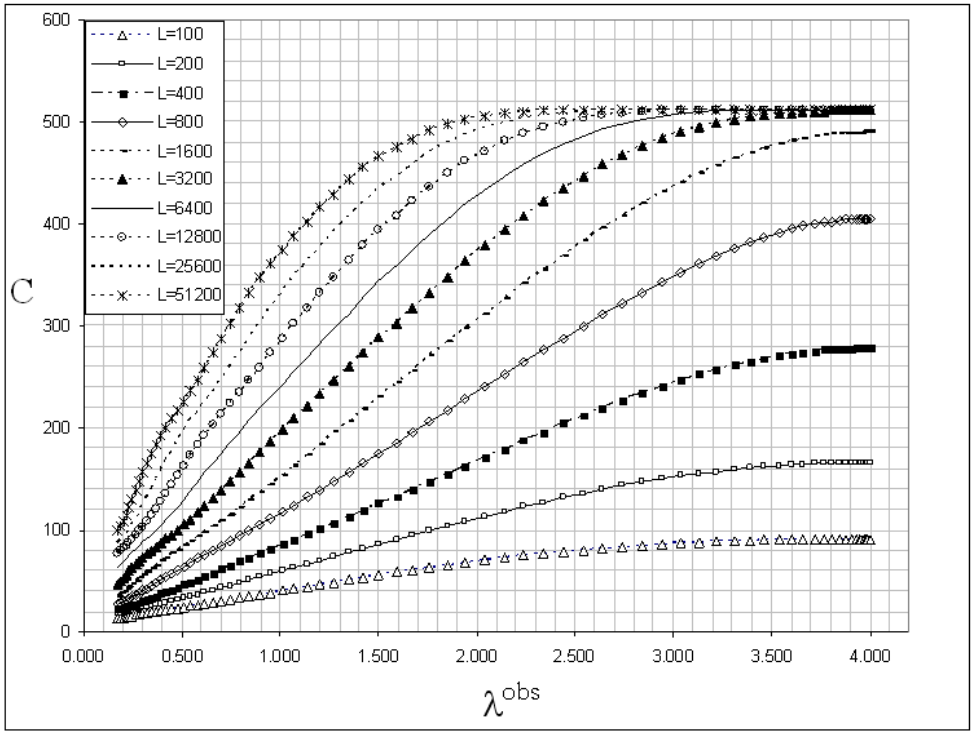


Gráfico 3.1. Influência dos parâmetros L e λ^{obs} no número C de seqüêmcias variantes esperadas em uma biblioteca resultante do *shuffling* de duas seqüências, ambas de tamanho N = 1.500, que diferem em M = 9 pares de bases.

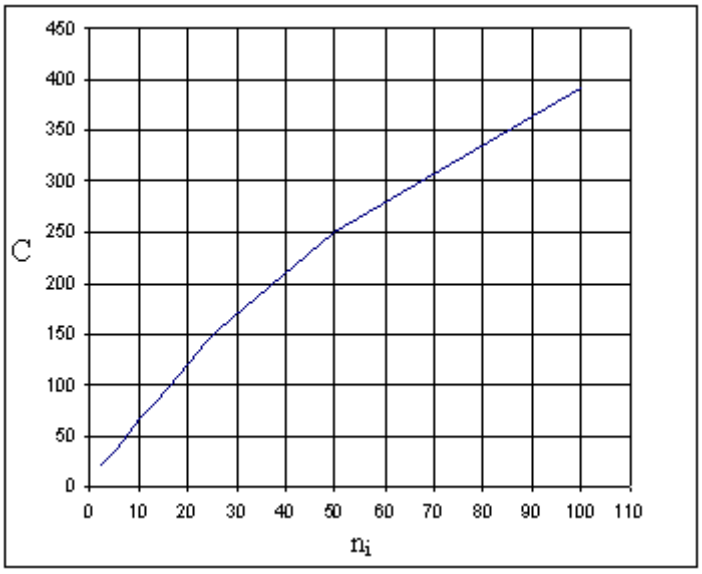


Gráfico 3.2. Influência dos valores do parâmetro n_i no número total C de variantes esperadas em uma biblioteca de *shuffling*, para valores fixos de N = 1.500, M = 9 e $\lambda^{true} = 8$.

Pode-se concluir que quanto maior o intervalo n_i que separa duas bases distintas e consecutivas entre as duas seqüências parentais, maior será o número esperado de variantes na biblioteca resultante uma vez que a probabilidade de um cruzamento ocorrer neste intervalo também aumenta. Contudo, na prática, além do tamanho do intervalo n_i, o tamanho dos fragmentos utilizados para dar início aos ciclos de PCR é um fator determinante na produção de diversidade. Fragmentos com tamanhos superiores ao tamanho mínimo entre todos os valores de n_i limitam a ocorrência de cruzamentos, como discutido mais adiante na Seção 3.5.3.

Informações relevantes sobre o software DRIVeR, principalmente informações relacionadas à sua execução estão descritos no Apêndice A, Seção A.1. A Seção A.2 do Apêndice apresenta o pseudo-código do algoritmo que subsidia o software DRIVeR devidamente comentado para viabilizar o seu completo entendimento.

3.2.5 Usando o DRIVeR

Esta seção condensa os resultados apresentados e discutidos em Montera et al. (2006) na avaliação de três seqüências como candidatas a pares de parentais em experimentos de DNA *shuffling*. O número esperado de variantes distintas presentes na biblioteca de DNA *shuffling* – representado por C – foi utilizado para avaliar a adequabilidade de três seqüências as quais, duas-a-duas, foram testadas como possíveis parentais em experimentos de DNA *shuffling*. As

seqüências utilizadas correspondem a proteínas da família das cistatinas[16] que codificam para inibidores de cisteíno protease. São elas:

– *Oryza sativa*, proveniente do arroz (número de acesso no GenBank[17] NM_190953), referenciada apenas como Oryza;
– *Saccharum officinarum*, proveniente da cana-de-açúcar (número de acesso no GenBank AY119689), referenciada apenas como Cane; e
– *Sorghum bicolor*, proveniente de mudas de sorgo (número de acesso no GenBank X87168), referenciada apenas como Sorghum.

A fim de verificar qual par de parentais, potencialmente, resultaria em uma biblioteca de DNA *shuffling* com um maior número de recombinantes e cujas seqüências apresentem um maior número de cruzamentos entre os parentais, diferentes condições de experimentos de DNA *shuffling* foram simulados utilizando o DRIVeR para os pares de parentais Cane–Oryza, Cane–Sorghum e Oryza– Sorghum.

Primeiramente, o alinhamento entre cada um dos três pares de seqüências foi construído, as mutações entre elas identificadas (segundo o procedimento descrito no Capítulo 4, Seção 4.2.2.1) e a distância entre as mutações consecutivas determinadas, para que o arquivo de entrada necessário à execução do DRIVeR (ver Apêndice A, Seção A.1) fosse gerado para cada caso. O tamanho do alinhamento, o número de mutações, bem como a distância média entre mutações consecutivas para cada um dos pares de seqüência estão sumarizados na Tabela 3.2.

Tabela 3.2. Estatísticas do alinhamento entre os pares de seqüências Cane–Oryza, Cane–Sorghum e Oryza–Sorghum.

Par de parentais	Tamanho do alinhamento (pb)	Número de mutações	Distância média entre mutações consecutivas (pb)
Cane–Oryza	118	11	8,9
Cane–Sorghum	122	12	8,5
Oryza– Sorghum	189	13	12,0

Foram realizadas simulações considerando bibliotecas de DNA *shuffling* de tamanhos L = 1.000, L = 5.000 e L = 10.000, e número real de cruzamentos (λ^{true}) variando de 1 até 10. Os

[16] Cistatinas são proteínas que inibem especificamente cisteíno proteases. Elas ocorrem naturalmente em diversas espécies de vegetais e acredita-se que fazem parte do mecanismo de defesa das plantas contra certos patógenos (COSTA, 2004).
[17] O GenBank é uma base de dados que armazena, entre outros, seqüências de nucleotídeos. As seqüências armazenadas no GenBank são de domínio público. Seu endereço é www.ncbi.nlm.nih.gov.

resultados das simulações para cada um dos três pares de parentais são apresentados na Tabela 3.3. Quanto ao número de cruzamentos observados (λ^{obs}), os resultados não apresentaram variações significativas, como poder ser visto na Tabela 3.3 e, por este motivo, a análise da adequabilidade das seqüências parentais foi feita apenas considerando-se o número esperado de variantes em cada um dos casos. Os resultados das simulações apresentados na Tabela 3.3 foram agrupados segundo o tamanho da biblioteca de DNA *shuffling* para as quais as simulações foram realizadas e estão graficamente representadas nos gráficos 3.3, 3.4, e 3.5.

Tabela 3.3. Resultados estimados pelo DRIVeR considerando o *shuffling* entre os parentais Oryza–Sorgum, Cane–Sorghum e Cane–Oryza.

		Oryza–Sorghum		Cane–Sorghum		Cane–Oryza	
L	λ^{true}	λ^{obs}	C	λ^{obs}	C	λ^{obs}	C
1.000	1	0,76	135,4	0,79	134,5	0,77	120,3
	2	1,40	261,1	1,45	257,3	1,40	225,3
	3	1,95	381,6	2,00	374,0	1,93	324,4
	4	2,42	489,0	2,48	477,0	2,37	411,8
	5	2,83	580,1	2,87	563,7	2,74	486,2
	6	3,18	654,9	3,21	634,3	3,06	547,9
	7	3,48	714,7	3,50	690,4	3,32	598,0
	8	3,74	761,8	3,75	734,2	3,54	638,1
	9	3,97	798,4	3,96	767,9	3,73	669,9
	10	4,17	826,7	4,15	793,8	3,89	694,9
5.000	1	0,76	291,5	0,79	279,7	0,77	238,1
	2	1,40	610,0	1,45	575,9	1,40	469,3
	3	1,95	940,2	2,01	872,3	1,93	688,6
	4	2,42	1.261,5	2,48	1.152,2	2,37	886,0
	5	2,83	1.561,5	2,87	1.406,2	2,74	1.057,3
	6	3,18	1.834,4	3,21	1.631,1	3,05	1.202,4
	7	3,48	2.078,6	3,50	1.926,9	3,32	1.323,5
	8	3,74	2.294,6	3,75	1.995,6	3,54	1.423,7
	9	3,97	2.484,4	3,96	2.139,8	3,73	1.505,9
	10	4,17	2.650,1	4,15	2.262,3	3,89	1.573,1
10.000	1	0,76	934,3	0,79	374,5	0,77	312
	2	1,40	838,1	1,45	771,6	1,40	607,3
	3	1,95	1.305,3	2,01	1.171,8	1,93	884,0
	4	2,42	1.765,0	2,48	1.547,0	2,37	1.124,3
	5	2,83	2.198,8	2,87	1.884,2	2,74	1.324,2
	6	3,18	2.598,2	3,21	2.179,5	3,05	1.485,6
	7	3,48	2.960,6	3,50	2.433,7	3,32	1.613,2
	8	3,74	3.285,0	3,75	2.649,6	3,54	1.712,4
	9	3,97	3.574,6	3,96	2.832,4	3,73	1.788,6
	10	4,17	3.831,9	4,15	2.935,6	3,89	1.846,7

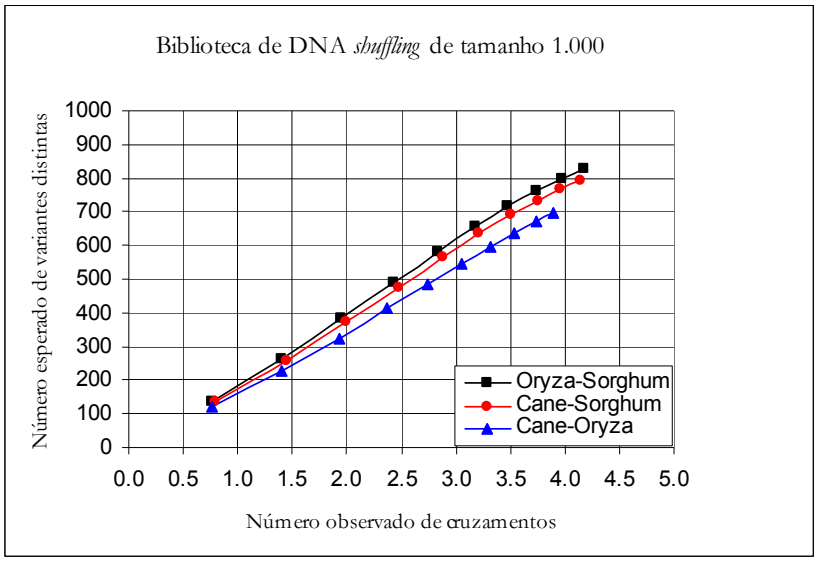


Gráfico 3.3. Número observado de cruzamentos *versus* número esperando de variantes distintas estimados pelo DRIVeR considerando bibliotecas de DNA *shuffling* de tamanho 1.000.

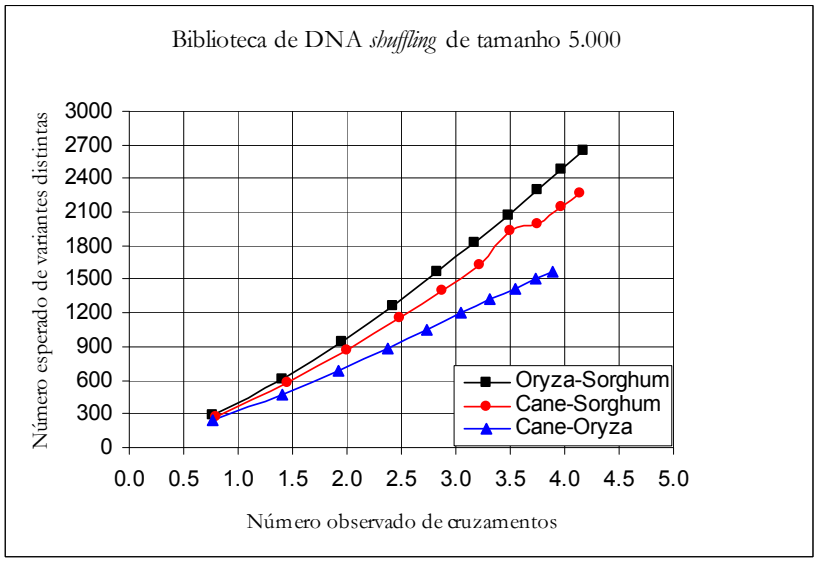


Gráfico 3.4. Número observado de cruzamentos *versus* número esperando de variantes distintas estimados pelo DRIVeR considerando bibliotecas de DNA *shuffling* de tamanho 5.000.

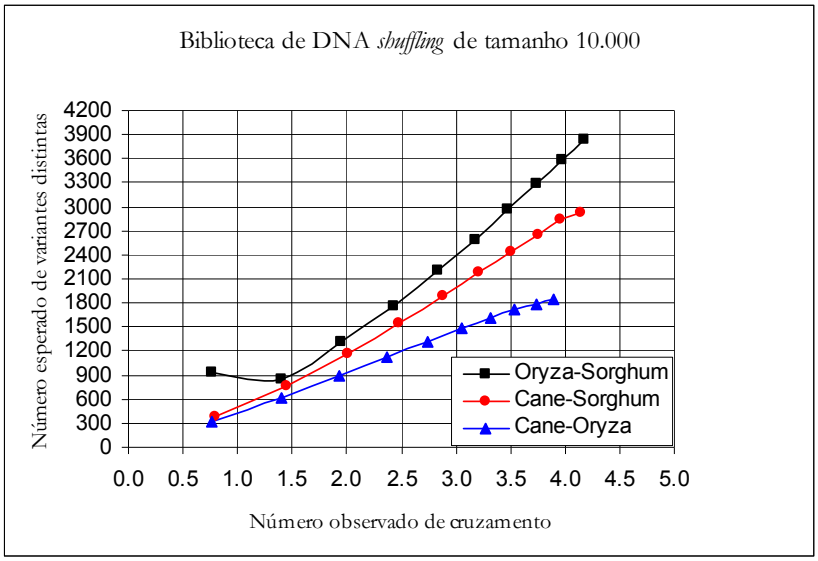


Gráfico 3.5. Número observado de cruzamentos *versus* número esperando de variantes distintas estimados pelo DRIVeR considerando bibliotecas de DNA *shuffling* de tamanho 10.000.

Com base nos resultados estimados pelo DRIVeR é possível inferir que a utilização do par de seqüências Oryza–Shorgum como parentais em experimentos de DNA *shuffling* apresenta vantagens sobre os dois outros pares de parentais avaliados, uma vez que o número de variantes esperadas C na biblioteca resultante do *shuffling* entre esses dois parentais é mais elevado. A vantagem do par Oryza–Shorgum sobre os outros dois pares de parentais pode ser atribuída ao maior número médio de bases que separam as mutações consecutivas (ver Tabela 3.2). Esse fato reforça a hipótese de que quanto maior a distância que separa as mutações consecutivas entre parentais distintos, maiores são as chances destas serem recombinadas em um único fragmento. Contudo, para diferenças muito pequenas entre a distância média que separa as mutações presentes entre pares de parentais (como é o caso dos parentais Cane–Shorgum e Cane–Oryza, cujas distâncias médias entre as mutações é de 8,5 e 8,9, respectivamente), a inferência de que uma maior distância média entre as mutações implica num maior o número de variantes, pode não ser válida, devido, basicamente, a localização das mutações ao longo das seqüências.

Como visto anteriormente na Seção 3.2.4, dadas duas seqüências parentais que diferem entre si em M pares de bases, o *shuffling* entre elas pode resultar num número máximo de $2^M - 2$ variantes distintas. Seja o conjunto de todas essas variantes definido como espaço amostral. Dado o número de variantes distintas esperadas em uma biblioteca de *shuffling* e o número de mutações entre os parentais, a cobertura do espaço amostral é definida como sendo a porcentagem de variantes obtidas do total de variantes possíveis, e pode ser calculada como:

$$\%C = \frac{C * 100}{2^M - 2}$$

Além do número absoluto de variantes distintas estimados pelo DRIVeR em cada um dos casos, o cálculo da porcentagem de cobertura traz informações relevantes sobre a diversidade que, de fato, foi atingida por uma biblioteca de DNA *shuffling*. A Tabela 3.4 mostra o tamanho do espaço amostral, o número total de variantes distintas obtidas bem como a porcentagem de cobertura atingida pelo *shuffling* entre cada um dos três pares de parentais avaliados, considerando as estimativas do DRIVeR para valores de $\lambda^{true} = 4$ e L = 1.000.

Tabela 3.4. Porcentagem de cobertura do espaço amostral para cada um dos pares de parentais considerando $\lambda^{true} = 4$ e L = 1.000.

	Oryza–Sorghum	Cane–Sorghum	Cane–Oryza
Espaço amostral	$2^{13} = 8.192$	$2^{12} = 4.096$	$2^{11} = 2.048$
C	489	477	411,8
%C	6%	11,6%	20%

Observe na Tabela 3.4 que, apesar dos valores de C serem relativamente próximos para todos os três pares de seqüências, a cobertuda do espaço amostral varia significativamente. Apesar do par Oryza–Sorghum apresentar a menor cobertura do espaço amostral, este par pode, ainda, ser considerado como o mais adequado ao processo de DNA *shuffling*, uma vez que irá produzir um número maior de seqüências variantes.

3.3 O modelo ShuffIt

O modelo ShuffIt proposto por Narendra Maheshri e David Schaffer (2003), assim como o modelo eShuffle proposto por Moore et al. (2001), descrito na Seção 3.4, baseiam-se em conceitos da termodinâmica e da cinemática envolvidos no processo de DNA *shuffling*.

3.3.1 Introdução

O modelo de Maheshri−Schaffer permite que informações relevantes tais como a eficiência da remontagem e o número de cruzamentos sejam avaliados a cada etapa do processo, uma vez que o conjunto de fragmentos sendo remontados pode ser acompanhado durante cada um dos n ciclos da reação de PCR.

3.3.2 Descrição do modelo

Antes do início do experimento de *shuffling*, os parentais são amplificados para que quantidades suficientes de DNA sejam obtidas para a realização do experimento. Sendo assim, inicialmente o modelo considera a existência de m cópias de cada parental e que estas já se encontram desnaturadas, ou seja, o conjunto de moléculas a serem submetidas ao *shuffling* é representado por moléculas de DNA de fita simples, denotadas por ssDNA (*single-stranded* DNA). Em todos os resultados apresentados no trabalho de Maheshri−Schaffer, foram considerados inicialmente a existência de 150 a 500 moléculas de ssDNA cuja fragmentação resultou em um intervalo de 1.500 a 5.000 fragmentos de ssDNA.

O algoritmo implementado para simular a fragmentação das moléculas é baseado na distribuição de Poisson, a qual resulta em uma distribuição exponencial do comprimento dos fragmentos produzidos, típica da fragmentação produzida pela enzima DNase I[18]. Os fragmentos produzidos possuem um valor médio de comprimento, denotado por AFS (*Average Fragment Size*). Assim como nos experimentos de *shuffling*, os fragmentos são purificados para que apenas aqueles cujo tamanho esteja compreendidos entre um valor mínimo MFS (*Minimum Fragment Size*) e máximo XFS (*Maximum Fragment Size*) sejam selecionados para dar continuidade ao processo.

O conjunto de fragmentos purificados (selecionados) é então submetido aos ciclos de PCR, para que a remontagem dos fragmentos ocorra. O modelo contempla a ocorrência de três possíveis eventos durante os ciclos de PCR:

- as moléculas de ssDNA colidem, duas-a-duas, randomicamente entre si;
- dada uma colisão, deve-se decidir se as moléculas irão se parear e, em caso afirmativo, em qual arranjo, ou seja, qual sobreposição entre os dois fragmentos ocorrerá;
- moléculas pareadas (ou hibridizadas) são estendidas na direção 5'→ 3'.

Sejam dois fragmentos S_1 e S_2 do tipo ssDNA. Caso S_1 e S_2 hibridizem, uma molécula de fita dupla, dita dsDNA (*double-stranded* DNA), é formada. Tal evento é descrito pela eq. (3.9).

$$S_1 + S_2 \leftrightarrow D \tag{3.9}$$

[18] Quanto maior o tempo de exposição do DNA à ação da enzima DNase I, maior será o número de cortes produzidos na molécula de DNA e, portanto, menor será o tamanho dos fragmentos resultantes.

Durante os ciclos de PCR considera-se a existência de duas classes distintas de fragmentos: uma na qual estão inseridos os fragmentos do tipo ssDNA e outra contendo os fragmentos do tipo dsDNA.

Para decidir se, e como o pareamento entre duas moléculas ssDNA que colidiram ocorrerá, todas as possíveis configurações nas quais essas duas moléculas podem parear-se devem ser testadas. Contudo, apenas sobreposições de tamanho mínimo igual a 7 bases são consideradas por este modelo. A decisão por uma ou outra configuração de pareamento é tomada com base na energia livre produzida por cada uma destas configurações. A variação de energia livre (ΔG) é calculada segundo o modelo NN (*Nearest-Neighbor*) utilizando os valores de ΔH e ΔS proposto por Allawi e SantaLucia (1997) (ver Tabela C.2, Apêndice C) com modificações, para que se possa considerar os efeitos da ocorrência de *mismatches* nas regiões de sobreposição e da concentração de sal utilizada na reação de DNA *shuffling*. Detalhes do modelo NN são descritos na Seção 3.4.2. Nesse modelo não são consideradas a ocorrência de eventos de pareamentos com *gaps*, como o ilustrado na Figura 3.8. Fragmentos que não conseguiram se parear são retornados ao conjunto de ssDNA para colisões adicionais.

```
G C T T G A T G C G C G
          | | * | |
          C G - G C G A G T T C
```

Figura 3.8. Exemplo do pareamento entre dois fragmentos de DNA no qual um *gap* ocorreu.

A influência das condições experimentais sobre as probabilidades de pareamento é descrita por um fator α que depende: da constante de equilíbrio K definida para uma dada variação de energia livre (ΔG) em determinada temperatura T, dada por $\Delta G_T = G_{dsDNA} - G_{ssDNA}$; da concentração inicial total (C_T) de fragmentos do tipo ssDNA; da constante dos gases R, que tem valor 8,314 $J.K^{-1}.mol^{-1}$ (ou 0,08206 $atm.L.K^{-1}.mol^{-1}$ ou 1,9872 $cal.K^{-1}mol^{-1}$); e da identidade desses fragmentos, representada por b, sendo b = 4 se os fragmentos são distintos e b = 1 se os fragmentos são complementares, como mostra a eq. (3.10).

$$\alpha = \frac{1}{KC_T b} + 1 \text{ sendo } K = \exp\left(\frac{-\Delta G_T}{RT}\right) \tag{3.10}$$

A probabilidade de um pareamento ocorrer é definida por:

$$X = \alpha - \sqrt{\alpha^2 - 1} \tag{3.11}$$

Assim, a probabilidade de pareamento decresce de 1 para 0 a medida que α aumenta de ≈ 1 para ∞ (infinito).

A extensão que ocorre quando duas moléculas ssDNA formam um par híbrido é 100% fiel, ou seja, o algoritmo de extensão implementado não considera os possíveis erros (*mismatches*) inseridos na molécula sendo estendida por uma polimerase.

Quando novos pareamentos não são mais observados[19], a etapa de colisão entre as moléculas de ssDNA é suspensa. Nesse momento, as moléculas que hibridizaram e foram estendidas (dsDNA) são desnaturadas e inseridas novamente no conjunto de moléculas ssDNA, juntamente com aquelas que não hibridizaram para que uma nova etapa de colisões seja repetida e fragmentos maiores sejam obtidos. Este processo é repetido até que novos pareamentos não sejam mais observados durante as colisões. Ao final de cada uma das etapas de colisões, é possível obter métricas relevantes como número de cruzamentos e eficiência da remontagem. Por fim, é feita a modelagem da PCR com *primers*. Nessa simulação, *primers* correspondentes às extremidades das seqüências parentais são utilizados para selecionar as seqüências remontadas que podem representar seqüências completamente remontadas (*full-length*).

O modelo Maheshri−Schaffer permite que a formação e a distribuição dos cruzamentos sejam analisadas a cada ciclo de PCR, sendo possível assim avaliar também os cruzamentos ocorridos entre fragmentos que não foram remontados completamente e como esses últimos podem influenciar no resultado final do experimento de *shuffling*.

Para realizar a validação do modelo, os autores realizaram experimentos de DNA *shuffling in vitro* e *in silico*, com um fragmento de DNA de tamanho 769 pb contendo o gene gfp *(Green Fluorescent Protein)*. No experimento *in vitro*, a fragmentação do parental foi realizada pela enzima DNase I, sendo descartados os fragmentos menores do que 25 pares de bases. A distribuição dos tamanhos dos fragmentos produzidos pela simulação *in silico* da fragmentação do parental (utilizando o processo de Poisson), segundo os autores, mostraram-se de acordo com a distribuição dos tamanhos dos fragmentos obtidos *in vitro* (dados não apresentados no trabalho).

Fragmentos resultantes da fragmentação de tamanhos compreendidos em diferentes intervalos foram submetidos a diferentes reações de PCR sem *primers* a fim de avaliar a influência do tamanho dos fragmentos em sua remontagem. Foram executados 30 ciclos de PCR sem

[19] Isto é, se após 10 tentativas, novos pareamentos não ocorreram.

primers para promover a remontagem, e os fragmentos resultantes foram analisados por meio de eletroforese em gel de agarose. O tamanho dos fragmentos obtidos foi determinado pela análise desta imagem digitalizada. Simultaneamente, foi realizada a simulação *in silico* da remontagem.

Os gráficos apresentados na Figura 3.9, adaptados de Maheshri e Schaffer (2003), foram apresentados pelos autores para que os resultados experimentais fossem comparados diretamente com os resultados obtidos do modelo. Nestes gráficos, os resultados dos 30 ciclos de PCR foram comparados com os resultados obtidos de 14 ciclos de execução do programa. A unidade para a medida de intensidade do gel é arbitrária. Com relação às concentrações iniciais (medidas em ng/μl) e aos tamanhos médio (AFS) e mínimo (MFS) dos fragmentos, medidos em pares de bases (pb), em cada um dos quatro experimentos apresentados na Figura 3.9 tem-se:

– Figura 3.9 (a): concentração inicial dos fragmentos = 8, AFS = 45 e MFS = 25;

– Figura 3.9 (b): concentração inicial dos fragmentos = 120, AFS = 45 e MFS = 25;

– Figura 3.9 (c): concentração inicial dos fragmentos = 8, AFS = 50 e MFS = 50;

– Figura 3.9 (d): concentração inicial dos fragmentos = 120, AFS = 50 e MFS = 50;

Considerando a concentração inicial dos fragmentos de 8 ng/μl e tamanho médio de 25 pb (Figura 3.9 (a)), ambos os experimentos, *in vitro* e *in silico* produziram um pico na intensidade do gel em torno de 750 pb, valor este próximo ao tamanho do gene gfp. Quando a concentração dos fragmentos foi aumentada para 120 ng/μl (Figura 3.9 (b)) um pico um pouco mais largo em torno de valor 750 pb foi observado tanto experimentalmente como pelo modelo. Note que para esses dois experimentos, o modelo foi capaz de capturar corretamente a tendência do pico (mais fino e mais largo) observado *in vitro*. Considerações similares podem ser observadas nos gráficos da Figura 3.9 (c) e (d), nos quais fragmentos maiores foram utilizados.

O Apêndice B, Seção B.1 descreve os detalhes necessários à execução do software ShuffIt, apresentando uma breve descrição de cada um dos arquivos que compõem a implementação do modelo bem como a descrição e a localização das variáveis que devem, necessariamente, ser modificadas a fim de que a execução do ShuffIt represente uma reação de DNA *shuffling* em particular.

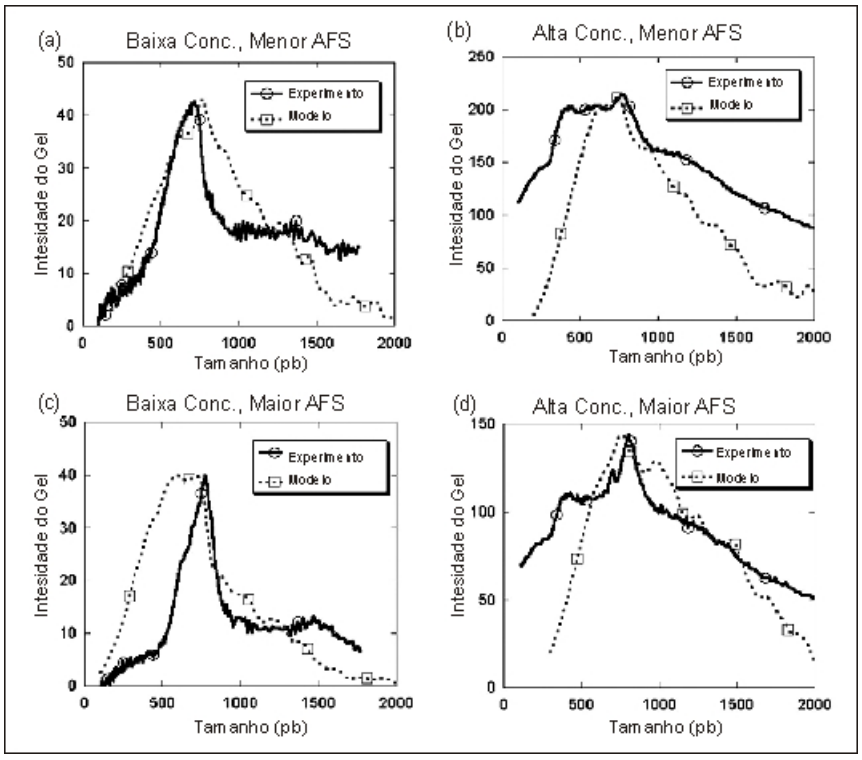


Figura 3.9. Gráficos comparativos entre os resultados *in silico* (modelo) e *in vitro* (experimento) do DNA *shuffling* do gene gfp.

3.4 O modelo eShuffle

Moore e colaboradores (2001) desenvolveram uma ferramenta computacional chamada eShuffle para avaliar a geração de cruzamentos em experimentos de DNA *shuffling*. A ferramenta é baseada em um modelo proposto pelos autores que considera o valor da variação da energia livre de Gibbs (ΔG) resultante de cada possível evento de pareamento entre fragmentos distintos, para determinar qual deles é o mais provável de acontecer.

3.4.1 Introdução

Diferentemente do DRIVeR, no modelo eShuffle, o conhecimento da seqüência de bases que compõem cada um dos parentais é importante, uma vez que o cálculo da energia livre depende das bases que irão se sobrepor em uma determinada região de pareamento. O modelo permite avaliar como o comprimento dos fragmentos, a temperatura de pareamento, a identidade entre as seqüências e o número de seqüências parentais influenciam no número, tipo e distribuição dos cruzamentos ao longo das novas seqüências remontadas resultantes do *shuffling*. O modelo

considera basicamente duas etapas do processo de *shuffling*: o pareamento e a remontagem dos fragmentos.

A modelagem proposta para o evento de pareamento envolve conceitos de termodinâmica bem como utiliza os parâmetros do modelo *Nearest Neighbor* aplicado a ácidos nucleicos, ou simplesmente NN (CROTHERS; ZIMM, 1964). Os conceitos de termodinâmica utilizados bem como o modelo NN estão descritos na Seção 3.4.2. As seções 3.4.3 e 3.4.4 descrevem o modelo proposto para os eventos de pareamento e para a remontagem dos fragmentos, respectivamente.

3.4.2 Conceitos de termodinâmica e o modelo *Nearest–Neighbor*

O modelo eShuffle proposto por Moore e colaboradores assume que durante a etapa de pareamento os fragmentos competem entre si pela escolha do fragmento ao qual irão se parear. Esta competição é quantificada pelas leis do equilíbrio termodinâmico, que permitem inferir:

- qual fração dos fragmentos irá se parear a uma dada temperatura;
- como os eventos de pareamento estarão distribuídos entre aqueles fragmentos que envolvem diferentes tamanhos de seqüências complementares, ou regiões de sobreposição;
- qual a proporção de eventos de pareamento que irão envolver *mismatches*, ou seja, qual a porção dos fragmentos que não compartilham regiões inteiramente complementares irão se parear.

A termodinâmica envolvida no evento de pareamento entre fragmentos de DNA pode ser analisada por meio da utilização dos parâmetros do modelo *Nearest-Neighbor*, que descrevem a contribuição da entalpia e da entropia de cada par de bases vizinhas existentes em uma região de sobreposição. Como referência básica para a descrição dos conceitos de termodinâmica apresentados a seguir utilizou-se Russell (2006).

A entalpia H mede o conteúdo de energia térmica de um sistema, cuja variação, representada por ΔH, mede o calor liberado ou absorvido por este sistema para uma dada transformação ocorrida sob pressão constante. A entropia S mede a desordem de um sistema e sua variação (ΔS) revela se o sistema sofreu aumento ou diminuição da desordem após a ocorrência de determinada reação. As variações de entalpia e entropia de um sistema ou reação são estabelecidas pelas equações (3.12) e (3.13), respectivamente.

$$\Delta H = H_{produtos} - H_{reagentes} \tag{3.12}$$

$$\Delta S = S_{produtos} - S_{reagentes} \tag{3.13}$$

nas quais:

$H_{produtos}$ = entalpia dos produtos

$H_{reagentes}$ = entalpia dos reagentes

$S_{produtos}$ = entropia dos produtos

$S_{reagentes}$ = entropia dos reagentes

ΔH = variação de entalpia do sistema

ΔS = variação da entropia do sistema

Com base nos valores das equações (3.12) e (3.13) tem-se:

$\Delta H < 0$, indicativo que a reação libera calor, e por isso é dita exotérmica

$\Delta H > 0$, indicativo que a reação absorve calor, e por isso é dita endotérmica

$\Delta S < 0$, indicativo que o sistema, após a reação, se torna menos desordenado

$\Delta S > 0$, indicativo que o sistema, após a reação, se torna mais desordenado

Uma reação ocorre mais espontaneamente sempre que:

– mais calor liberar, ou seja, quanto menor for o seu ΔH;
– mais desordenado se tornar o sistema após a reação, ou seja, quanto maior for o seu ΔS.

É importante notar, entretanto, que a espontaneidade de uma reação depende da entropia total, isto é, da entropia do sistema e de sua vizinhança, e não apenas da variação associada ao sistema. Se for considerado, por exemplo, o universo como vizinhança, o cálculo se torna impraticável. Como solução, considera-se que as reações ocorrem sob pressão e temperatura constantes pois, desta forma, a entropia da vizinhança depende somente da quantidade de calor transferido da reação para a vizinhança e da temperatura na qual esse calor é transferido, representados pelas variáveis calor absorvido pela vizinhança (calor_viz) e temperatura (T), como mostra a eq. (3.14)

$$\Delta S_{vizinhança} = \frac{calor\_viz}{T} \tag{3.14}$$

Sob pressão e temperatura constantes, o calor absorvido pela vizinhança corresponderá ao calor liberado pelo sistema. Se o sistema libera calor para a vizinhança, ΔH é negativo. Desta forma, a eq. (3.14) pode ser reescrita como a eq. (3.15).

$$\Delta S_{\text{vizinhança}} = \frac{-\Delta H_{\text{sistema}}}{T} \tag{3.15}$$

Assim, é possível calcular a entropia total como a soma das entropias do sistema mais a entropia das vizinhanças:

$$\Delta S_{\text{total}} = \Delta S_{\text{sistema}} + \Delta S_{\text{vizinhança}}$$

$$\Delta S_{\text{total}} = \Delta S_{\text{sistema}} - \frac{\Delta H_{\text{sistema}}}{T}$$

$$-T\Delta S_{\text{total}} = \Delta H_{\text{sistema}} - T * \Delta S_{\text{sistema}} \tag{3.16}$$

A fim de facilitar a definição de reação espontânea, utiliza-se a função de Energia Livre de Gibbs, denotada por G, que estabelece uma relação entre entalpia e entropia:

$$G = H - T * S \tag{3.17}$$

Analogamente, define-se a variação da energia livre de Gibbs (ΔG) como:

$$\Delta G = \Delta H - T * \Delta S \tag{3.18}$$

para uma reação que ocorre à temperatura constante T, expressa em Kelvin[20] (K). Alternativamente, é possível definir a variação da energia livre de Gibbs pela eq. (3.19):

$$\Delta G = R * T * \ln(T) \tag{3.19}$$

É possível agora definir a espontaneidade de uma reação com base apenas no valor da variação de sua energia livre. As relações entre variação de energia livre e espontaneidade de uma reação, considerando-se pressão (P) e temperatura (T) constantes, estão sumarizadas na Tabela 3.5.

[20] A conversão de graus Kelvin (K) para graus Celsius (C) é dada por C = K – 273,15.

Tabela 3.5. Relação entre a variação da energia livre de Gibbs e a espontaneidade de uma reação.

ΔG (P e T constantes)	Reação
< 0	Espontânea
= 0	Em equilíbrio
> 0	Não espontânea

Uma vez definida a energia livre de Gibbs, é possível apresentar o modelo *Nearest-Neighbor* (NN). O modelo NN, descrito para ácidos nucleicos, assume que a estabilidade da ligação entre um determinado par de bases depende da identidade e da orientação dos pares de bases vizinhos a este em uma região de sobreposição (SANTALUCIA, 1998). Segundo o modelo, dois pares de bases vizinhos contribuem com um determinado valor de variação de entalpia (ΔS) e entropia (ΔH) para o cálculo da energia livre total correspondente à região de pareamento. Mais precisamente, de acordo com o modelo NN, a energia livre deve ser calculada como a soma de três termos distintos (FREIER, 1986):

– a energia livre de iniciação de uma hélice (fita dupla de DNA) associada com a formação do primeiro par de base da fita dupla;
– soma da energia livre associada com cada um dos pares de bases subseqüentes, sendo este evento denominado propagação; e
– energia associada à simetria das moléculas, ou seja, se estas são ou não complementares entre si.

Considere os pares de bases complementares A–T e C–G. A notação utilizada para representar a ocorrência consecutiva dos pares de bases $\begin{matrix} 5'\,AC\,3' \\ 3'\,TG\,5' \end{matrix}$ em uma região de sobreposição (fita dupla de DNA) é AC/TG. Dadas as quatro bases A, C, G e T que compõem a molécula de DNA, dezesseis pares de bases distintos e perfeitamente pareados, isto é, sem *mismatches*, podem ocorrer em uma região de sobreposição, são eles: AA/TT, TT/AA, AT/TA, TA/AT, CA/GT, TG/AC, GT/CA, AC/TG, CT/GA, AG/TC, GA/CT, TC/AG, CG/GC, GC/CG, GG/CC, CC/GG. Pares de bases com *mismatches* também podem ocorrer em uma região de sobreposição.

A Figura 3.10 apresenta dois exemplos de regiões de sobreposição entre dois pares de fragmentos de DNA distintos, nas quais os pares de bases vizinhos que contribuem para a estabilidade do pareamento estão destacados dois a dois.

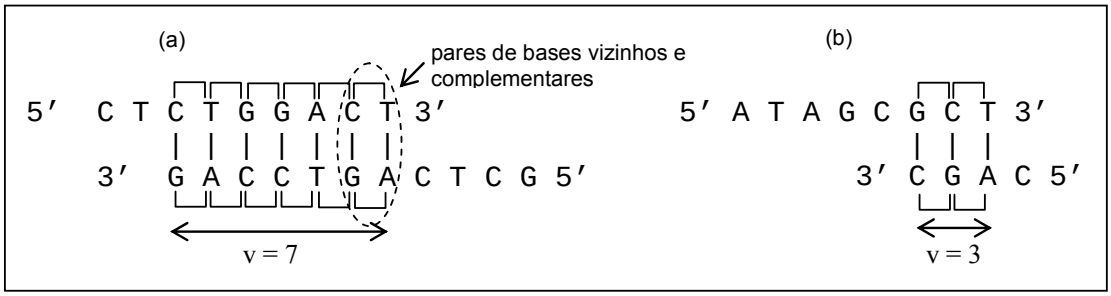


Figura 3.10. Duas regiões de sobreposição entre dois pares de fragmentos de DNA distintos. Os pares de bases vizinhos e consecutivos estão indicados pelos símbolos ⊓ e ⊔.

Seja v o tamanho de uma região de sobreposição entre duas seqüências de fita simples de DNA. A variação da energia livre de Gibbs para um determinado recombinante cuja região de sobreposição tem tamanho v é dada pela eq. (3.20):

$$\Delta G = \sum_{i=1}^{v-1}\left(\Delta H_{i,i+1} - T\Delta S_{i,i+1}\right) + \left(\Delta H_{\text{iniciação}} - T\Delta S_{\text{iniciação}}\right) + \left(\Delta H_{\text{simetria}} - T\Delta S_{\text{simetria}}\right) \tag{3.20}$$

$$\Delta G = \sum_{i=1}^{v-1}\left(\Delta G_{i,i+1}\right) + \Delta G_{\text{iniciação}} - \Delta G_{\text{simetria}}$$

A Tabela 3.6 apresenta os valores de ΔS e ΔH propostos por Allawi e SantaLucia (1997), Sugimoto et al. (1996), SantaLucia et al. (1996) e Breslauer (1986), para uma temperatura constante de 310K (ou 37ºC) e 1 M NaCl para os dezesseis pares de bases sem *mismatches*. A Tabela 3.6 traz ainda os valores sugeridos em cada uma das referências para os parâmetros *iniciação* e *simetria*. Valores de ΔS e ΔH para pares de bases com *mismatches* podem ser encontrados nas seguintes referências: (ALLAWI; SANTALUCIA, 1998), (ALLAWI; SANTALUCIA, 1997) e (HE et al., 1991).

Tabela 3.6. Valores de variação de entropia (ΔH) e entalpia (ΔS) para os pares de bases vizinhos.

	ALLAWI; SANTALUCIA 1997		SUGIMOTO et al., 1996		SANTALUCIA et al., 1996		BRESLAUER et al., 1986	
Par NN	ΔH	ΔS	ΔH	ΔS	ΔH	ΔS	ΔH	ΔS
AA/TT TT/AA	-7,9	-22,2	-8,0	-21,9	-8,4	-23,6	-9,1	-24,0
AT/TA	-7,2	-20,4	-5,6	-15,2	-6,5	-18,8	-8,6	-23,9
TA/AT	-7,2	-21,3	-6,6	-18,4	-6,3	-18,5	-6,0	-16,9
CA/GT TG/AC	-8,5	-22,7	-8,2	-21,0	-7,4	-19,3	-5,8	-12,9
GT/CA AC/TG	-8,4	-22,4	-9,4	-25,5	-8,6	-23,0	-6,5	-17,3
CT/GA AG/TC	-7,8	-21,0	-6,6	-16,4	-6,1	-16,1	-7,8	-20,8
GA/CT TC/AG	-8,2	-22,2	-8,8	-23,5	-7,7	-20,3	-5,6	-13,5
CG/GC	-10,6	-27,2	-11,8	-29,0	-10,1	-25,5	-11,9	-27,8
GC/CG	-9,8	-24,4	-10,5	-26,4	-11,1	-28,4	-11,1	-26,7
GG/CC CC/GG	-8,0	-19,9	-10,9	-28,4	-6,7	-15,6	-11,0	-26,6
Iniciação C/G	0,1	-2,8	0,6	-9,0	0	-5,9 ± 0,8	0	-16,77
Iniciação T/A	2,3	4,1	0,6	-9,0	0	-9,0 ± 3,2	0	-20,13
Simetria*	0	-1,4	0,0	-1,4	0	-1,4	0	-1,34

Unidades dos parâmetros: ΔH é dado em kcal/mol e ΔS é dado em cal/Kmol.
*a simetria só se aplica a seqüências que são complementares entre si (*self*-complementares). Caso contrário o valor do parâmetro simetria é zero.

Como exemplo, considere o cálculo da energia livre da região de sobreposição mostrada na Figura 3.10 (a), considerando que a reação ocorre à temperatura de 310K e pressão constante, e os valores de ΔS e ΔH sugeridos por Allawi e SantaLucia (1997).

$$\Delta G = (\Delta H_{CT/GA} - T * \Delta S_{CT/GA}) + (\Delta H_{TG/AC} - T * \Delta S_{TG/AC}) + (\Delta H_{GG/CC} - T * \Delta S_{GG/CC}) +$$

$$(\Delta H_{GA/CT} - T * \Delta S_{GA/CT}) + (\Delta H_{AC/TG} - T * \Delta S_{AC/TG}) + (\Delta H_{CT/GA} - T * \Delta S_{CT/GA}) +$$

$$(\Delta H_{iniciação} - T * \Delta S_{iniciação}) + (\Delta H_{simetria} - T * \Delta S_{simetria})$$

$$= (-7800 - 310 * -21{,}0) + (-8500 - 310 * 22{,}7) + (-8000 - 310 * -19{,}9) +$$

$$= (-8200 - 310 * -22{,}2) + (-8400 - 310 * 22{,}4) + (-7800 - 310 * 21{,}0') +$$

$$= (100 - 310 * -2{,}8) + (0 - 310 * -1{,}4)$$

$$= -8982 \text{ cal} = -8{,}982 \text{ kcal}$$

3.4.3 O modelo de pareamento

Uma vez apresentados alguns dos conceitos de termodinâmica necessários, bem como o modelo NN, é possível descrever o modelo de pareamento proposto por Moore et al. (2001). Na descrição desse modelo, os fragmentos também podem ser chamados de *templates* e dois fragmentos que se parearam formam um complexo chamado duplex.

Durante o experimento de DNA *shuffling*, após as seqüências parentais terem sido fragmentadas, e estes fragmentos posteriormente desnaturados, estes irão se parear a fim de que a remontagem dos fragmentos ocorra. É preciso lembrar que a possibilidade de pareamento de um dado fragmento, provavelmente, não será única. Como exemplo, considere o *template* A = TGATGCGCGCTA que compartilha regiões complementares com os fragmentos F_1 = GCGAGCTGAG, F_2 = GCTCTGTTGC, F_3 = GATGAGCTC e F_4 = GGATAGCTA. Diferentes tamanhos de sobreposições, contendo ou não *mismatches*, podem ocorrer entre o *template* e cada um destes fragmentos, como mostra a Figura 3.11.

Dadas as quatro possibilidades de pareamentos entre o *template* A e os fragmentos F_1, F_2, F_3 e F_4, o modelo assume que o pareamento mais provável de acontecer é o que resultar em uma menor energia livre, uma vez que este tipo de reação é mais espontânea. Genericamente, uma sobreposição de tamanho v entre dois fragmentos quaisquer será representada como mostrado na Figura 3.12.

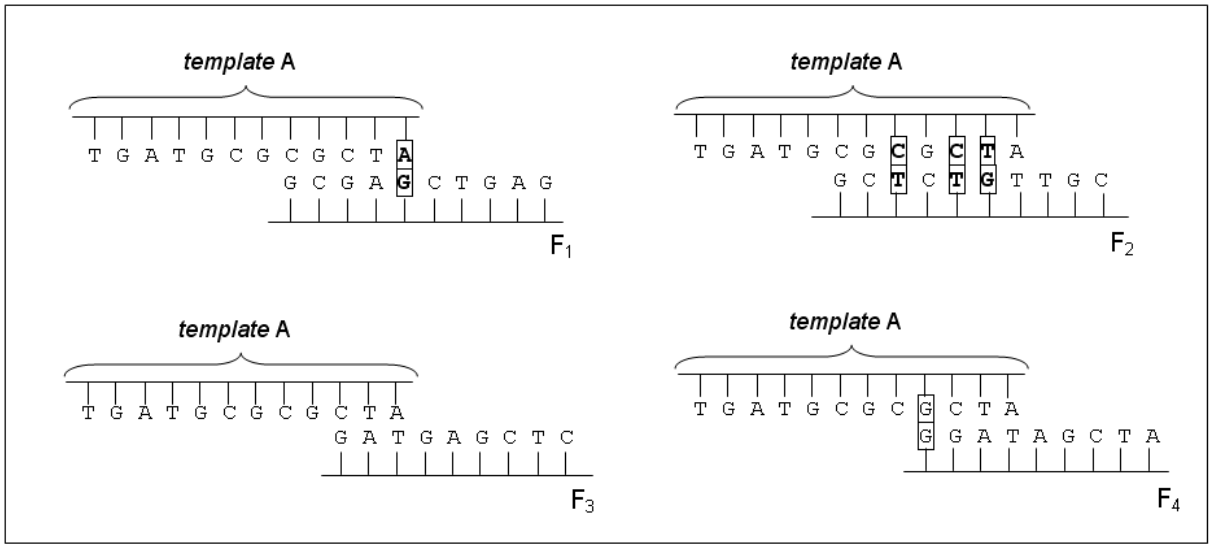


Figura 3.11. Diferentes pareamentos entre o *template* A e cada um dos fragmentos F_1, F_2, F_3 e F_4, com diferentes tamanhos de sobreposição, contendo ou não *mismatches* (destacados com por um retângulo).

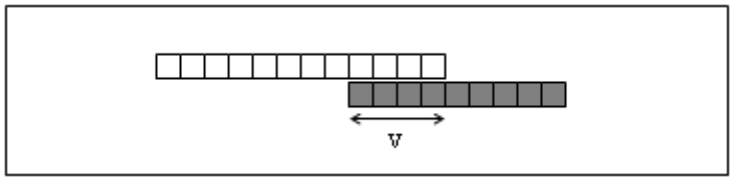


Figura 3.12. Representação genérica do pareamento entre dois fragmentos de DNA com uma região de sobreposição de tamanho v.

Uma vez que a energia livre associada a um evento de pareamento pode ser calculada, as diferentes possibilidades de pareamento entre o *template* e cada fragmento podem ser estimadas sob diferentes temperaturas pelo modelo proposto. Considere que a reação de pareamento de um *template* A e um fragmento F, formando o duplex AF, seja representada por:

$$A + F \leftrightarrow AF \tag{3.21}$$

Na situação de equilíbrio, a proporção entre a concentração do reagente e do produto se mantém constante. Essa proporção é expressa pela constante de equilíbrio K(T), que relaciona a fração molar dos *templates*, fragmentos (reagentes) e duplexes (produtos) em diferentes temperaturas. A constante de equilíbrio K, em uma dada temperatura T, para a reação representada na eq. (3.21) é dada por:

$$K(T) = \frac{x_{AF}}{x_A x_F} = \exp\left(\frac{\Delta G(T)}{RT}\right) \tag{3.22}$$

Considere que x representa a fração molar dos elementos na reação e x^0 a concentração inicial dos elementos na mistura, de forma que $x_A = x_A^0 - x_{AF}$ representa a concentração dos *templates* (A) que não se pareiaram enquanto $x_F = x_F^0 - x_{AF}$ representa a concentração dos fragmentos (F) que não se pareiaram a um *template* A.

Seja a(T) a curva de pareamento definida como a fração dos *templates* que se pareiaram a uma temperatura T. Essa curva é dada pela eq. (3.23).

$$a(T) = \frac{x_{AF}}{x_A^0} = \frac{x_A^0 - x_A}{x_A^0} = \frac{x_A^0}{x_A^0} - \frac{x_A}{x_A^0} = 1 - \frac{x_A}{x_A^0} \tag{3.23}$$

A temperatura de *melting* T_m, ou temperatura de pareamento, é definida como a temperatura na qual metade dos *templates* está na forma de híbridos, ou seja, estão formando duplexes. A utilização da fórmula (3.23), descrita pelo modelo de energia livre, para o cálculo da temperatura de *melting* tem mostrado bons resultados quando comparados com os resultados obtidos pela utilização de fórmulas determinadas empiricamente.

Observa-se que, em geral, quanto maior é a região de sobreposição entre dois fragmentos maior é a temperatura de *melting*, enquanto que pequenas regiões de sobreposição, ou a ocorrência de *mismatches*, ou ainda um baixo conteúdo de bases CG, contribuem para diminuir a temperatura de *melting*.

A eq. (3.21) considera que apenas um fragmento F concorre pelo pareamento com um determinado *template* A. Porém, como exemplificado na Figura 3.11, existem fragmentos de diferentes tamanhos, origens e regiões de sobreposição[21] competindo por um mesmo *template*. A eq. (3.24) reescreve a eq. (3.21) para acomodar essas informações,

$$A + F_{mv} \leftrightarrow AF_{mv} \tag{3.24}$$

na qual m indica o parental do qual o fragmento F é originário e v indica o tamanho da sobreposição entre o fragmento F e o *template* A durante o pareamento. Dessa forma, a seletividade de um determinado fragmento F, originário do parental m e cuja sobreposição com o *template* A tem tamanho v, também é dependente das concentrações de todos os outros fragmentos (originários dos diferentes parentais) que também podem se parear com o *template*. Esta seletividade é dada pela eq. (3.25), sendo ela dependente da temperatura.

[21] Com ou sem *mismatches*.

$$s_{mv}(T) = \frac{x_{AF_{mv}}}{\sum_{v', m'} x_{F_{m'v'}}} \tag{3.25}$$

A diferença de energia livre entre as possíveis escolhas para a ocorrência de um evento de pareamento e a concentração relativa dos fragmentos na mistura determinam qual o pareamento é dominante em uma dada temperatura. Por exemplo, em altas temperaturas, o pareamento entre fragmentos que possuem grande sobreposição com o *template* e no qual não ocorrem *mismatches* são dominantes sobre todos os outros possíveis híbridos, devido à alta entalpia que estes proporcionam à reação. Com a diminuição da temperatura, há um favorecimento da ocorrência de eventos de pareamento entre *templates* e fragmentos que compartilham uma menor região de sobreposição e até mesmo entre regiões que se sobrepõem com alguns *mismatches*.

Como em experimentos de DNA *shuffling* os tamanhos dos fragmentos utilizados estão distribuídos em um intervalo conhecido, e devido à seletividade para o pareamento ser dependente da temperatura, deve-se considerar que híbridos são formados ao longo de todo um intervalo decrescente de temperaturas que vai desde a temperatura de desnaturação até a temperatura mínima utilizada na etapa de pareamento, e não apenas em uma dada temperatura fixa. Dessa forma, para representar a seletividade total S_{mv} de um dado fragmento originário do parental m cuja sobreposição com o *template* A tem tamanho v ao longo de um intervalo de temperaturas, é necessário integrar a seletividade $S_{mv}(T)$ com relação à temperatura, como mostrado na eq. (3.26).

$$S_{mv} = \int_{T_{pareamento}}^{T_{desnatuação}} s_{mv} \frac{d_a(T)}{dT} dT \tag{3.26}$$

Dado um conjunto de fragmentos competindo por um dado *template* A sob um determinado intervalo entre a temperatura de desnaturação e temperatura de pareamento, S_{mv} quantifica a seletividade global do *template* A sobre um fragmento F.

3.4.4 Remontagem dos fragmentos

O processo de remontagem dos fragmentos é modelado como sucessivos eventos de pareamentos. Nesse momento, a seletividade do evento de pareamento é dependente apenas da complementaridade entre o *template* e os fragmentos que competem entre si pela hibridação, sendo que o *template* corresponde ao último fragmento remontado até o momento. Para

simplificar a apresentação da modelagem dessa etapa de remontagem dos fragmentos, apenas fragmentos de um mesmo tamanho L serão considerados.

O objetivo principal da modelagem da remontagem dos fragmentos é responder a seguinte pergunta: qual é probabilidade que uma seqüência completamente remontada, com B nucleotídeos, contenha x cruzamentos? Esta probabilidade é representada por $\prod^{x}$.

O pareamento de um fragmento cuja última base encontra-se na posição i – 1 com um fragmento de tamanho L, cuja sobreposição entre eles tem tamanho v[22], implica que o fragmento resultante terá o tamanho (i – 1)+(L – v), e esta será a nova posição i – 1 na qual o próximo fragmento deve se hibridar dando continuidade a remontagem sucessiva dos fragmentos. A Figura 3.13, adaptada de Moore et al. (2001), mostra um esquema do processo de remontagem dos fragmentos.

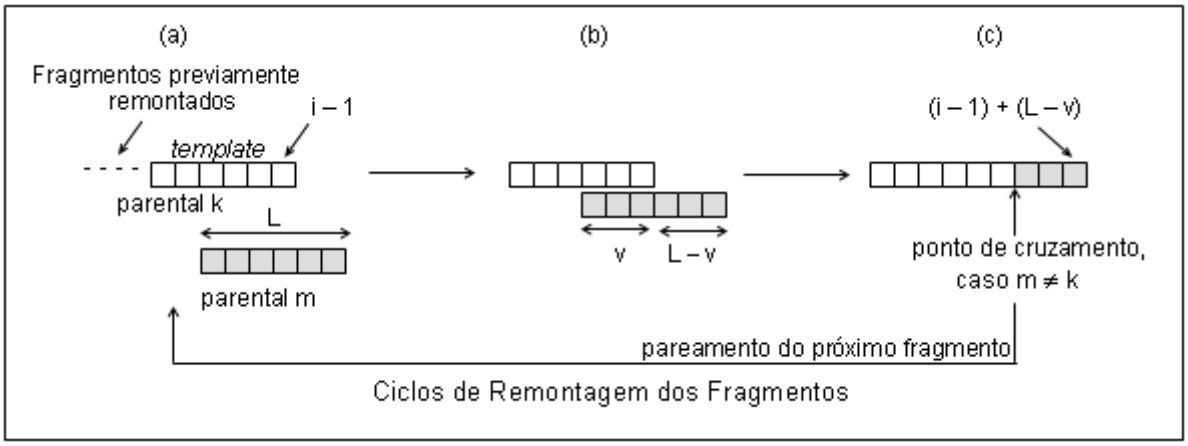


Figura 3.13. Esquema representativo do processo de remontagem dos fragmentos. No ciclo de remontagem dos fragmentos, o fragmento resultante em (c) passa a ser o *template* em (a) ao qual outro fragmento irá se parear, e assim sucessivamente.

O tamanho v da sobreposição pode ser qualquer valor entre 1 e L – 1. A sobreposição v = 0 ou v = L não são possíveis, uma vez que:

– v = 0 implica que não há sobreposição entre os fragmentos, logo o pareamento não pode ocorrer;
– v = L implica numa sobreposição total entre os dois fragmentos, evento não caracterizado como cruzamento, uma vez que não irá ocorrer a extensão dos fragmentos pela polimerase.

[22] O modelo implementado pelo eShuffle considera que sobreposições de tamanho mínimo igual a 2 podem resultar no pareamento entre os fragmentos.

A Figura 3.14 representa todas as possíveis sobreposições que podem gerar cruzamentos entre dois fragmentos de origens m e k, ambos de tamanho L, caso m e k representem parentais distintos.

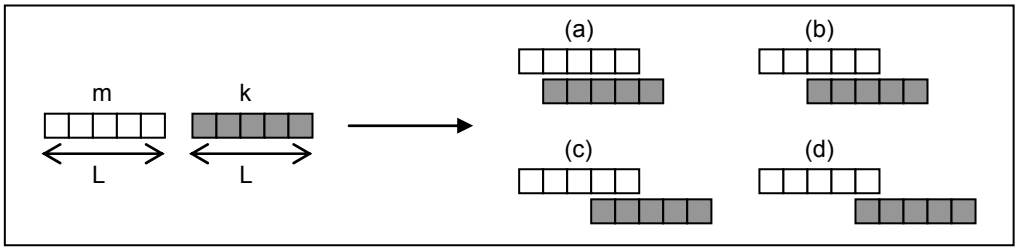


Figura 3.14. Possíveis sobreposições entre dois fragmentos originários dos parentais m e k, ambos de tamanho L, os quais podem resultar em cruzamentos, caso $m \neq k$.

Considere inicialmente P_{ik}^{x} como sendo a probabilidade de que a remontagem, a partir de uma posição qualquer i até a posição B de uma seqüência de DNA, contenha exatamente x cruzamentos, dado que o fragmento que termina na posição i – 1 é originário do parental k. Quando um fragmento F_m originário do parental m pareia-se a um fragmento F_k originário de um parental k, se m = k, ocorre a formação de homoduplex, se $m \neq k$, um heteroduplex é formado, ou seja, um cruzamento ocorreu entre os parentais m e k. Caso o fragmento do parental k tenha se pareado a um fragmento originário do parental m na posição i, e $m \neq k$, para o cálculo de P_{ik}^{x} é necessário calcular ainda a probabilidade de que x – 1 cruzamentos ocorram no restante da seqüência remontada até a posição B.

Como as probabilidades de ocorrer uma sobreposição de tamanho v = 1, 2, ..., L – 1 são mutuamente exclusivas e as mesmas dependem da seletividade do fragmento em relação ao *template* em uma dada temperatura, temos que a probabilidade P_{ik}^{x} é dada pela eq. 3.27.

$$P_{ik}^{x} = \sum_{v=1}^{L-1} S_{kv} P_{i+L-v,k}^{x} + \sum_{m \neq k} \sum_{v=1}^{L-1} S_{mv} P_{i+L-v,m'}^{x-1} \qquad (3.27)$$

$$\forall\, x > 0,\ \forall\, i > L \text{ e } \forall\, k$$

O primeiro termo no cálculo de P_{ik}^{x} representa o caso onde o primeiro fragmento remontado não é originário do parental m e sim do parental k, desta forma, a primeira remontagem não gera cruzamento. Este fato exige que na remontagem dos fragmentos restantes, ocorram exatamente x cruzamentos. Diferentemente, o segundo termo considera que o primeiro

fragmento remontado é originário do parental m e que m ≠ k, de maneira que um primeiro cruzamento ocorreu e, por isso, são necessários apenas à ocorrência de x – 1 cruzamentos nas remontagens seguintes.

Sabendo como calcular P_{ik}^{x}, é possível determinar o valor de $\prod^{x}$. Inicialmente no processo de remontagem, qualquer que seja o fragmento inicial, este possui tamanho L, o que implica dizer que qualquer que seja a sobreposição do fragmento que se pareou a este, os novos nucleotídeos serão inseridos a partir da posição i = L + 1. Além disso, no estado inicial da remontagem, temos que a probabilidade de que o fragmento a dar início a uma nova seqüência sendo remontada seja originário do parental m se iguala à concentração relativa (C_m) deste parental na mistura, o que implica que a probabilidade de que a seqüência remontada (*full-length*) contenha x cruzamentos, corresponde à probabilidade de que ocorram *x* cruzamentos após a posição L + 1, dada pela eq. (3.28).

$$\hat{O}^{x} = \sum_{m} C_m P_{L+1,m}^{x}, \; x = 0,1,2,... \tag{3.28}$$

As seguintes condições limitantes são consideradas para o cálculo de $\prod^{x}$ para garantir que nenhum cruzamento ocorre após a posição i = B:

- $P_{ik}^{0} = 1 \quad \forall i > B \text{ e } \forall k$
- $P_{ik}^{x} = 0 \quad \forall x > 0, \forall i > B \text{ e } \forall k$

Assim, o modelo proposto de remontagem, permite estimar qual a fração dos fragmentos remontados irão conter x = 0, 1, 2, ..., cruzamentos.

Informações relevantes sobre como executar o software eShuffle, estão descritos no Apêndice C, Seção C.1. A Seção C.2 do Apêndice apresenta o pseudo-código do algoritmo que subsidia o software eShuffle. Por fim, na Seção C.3 é apresentado um estudo sobre a influência da utilização de diferentes valores do modelo NN encontrados na literatura sobre as estimativas do eShuffle.

3.5 O modelo Sun

Diferentemente do modelo proposto por Patrick et al. (2003), implementado pelo software DRIVeR; do modelo proposto por Moore et al. (2001), implementado pelo software eShuffle; o do moldelo proposto por Maheshri e Schaffer (2003) e implementado pelo software ShuffIt; O modelo matemático proposto por Fengzhu Sun (1999) para modelar o DNA *shuflling* não é disponibilizado via software. Em contato com o autor, Sun informou que não dispunha de nenhuma implementação que pudesse fornecer (informação pessoal)[23]. Desta forma, a descrição que segue baseou-se apenas na publicação original do modelo proposto.

3.5.1 Introdução

Fengzhu Sun (1999) propôs um modelo matemático para modelar o DNA *shuflling* e estudou o progresso de experimentos desse tipo com base no modelo. O modelo proposto consiste de duas partes principais. A primeira delas refere-se à aplicação do modelo de Lander−Waterman (LANDER; WATERMAN, 1988) para o mapeamento físico dos clones por meio de *fingerprinting* de clones randômicos a fim de modelar a distribuição das regiões que podem ser remontadas pelo *shuffling*, ou seja, das regiões onde cruzamentos podem ocorrer. A segunda parte do trabalho apresenta um modelo para a recombinação de fragmentos de DNA originários de parentais distintos. Cada uma das partes do modelo serão descritas separadamente nas seções 3.5.2 e 3.5.3.

3.5.2 O modelo de distribuição de clones randômicos

O modelo de clones randômicos, proposto por Lander−Waterman (LANDER; WATERMAN, 1988) foi inicialmente aplicado em projetos de mapeamento físico de DNA por meio de *fingerprinting*. Nesse tipo de projeto, deseja-se determinar a seqüência de ácidos nucleicos de um determinado organismo. Devido a limitações inerentes ao processo de clonagem e de seqüenciamento, seqüências muito longas não podem ser clonadas diretamente e seqüenciadas. Desta forma, para a determinação da composição de seqüências longas, elas são primeiramente fragmentadas em seqüências menores, amplificadas e, em seguida, sua composição determinada pela utilização de técnicas de seqüenciamento. Os fragmentos seqüenciados são, em seguida, remontados para que a seqüência completa do organismo seja determinada.

[23] SUN, F. Modeling DNA Shuffling. Mensagem recebida por lumontera@gmail.com em 20 de Abril de 2007.

Considere um conjunto de fragmentos a serem remontados. A remontagem é feita basicamente pela união de fragmentos que se sobrepõem, ou seja, fragmentos que compartilham regiões complementares. Desta forma, fragmentos que possuem regiões complementares unem-se para formar um fragmento maior. O agrupamento de um conjunto de fragmentos é denominado ilha. Após a análise e o agrupamento em ilhas dos fragmentos do conjunto inicial, inúmeras ilhas podem ter sido formadas. Regiões desconexas entre as ilhas são denominadas oceanos. Como exemplo, considere a remontagem de uma seqüência de DNA de um determinado organismo cujo tamanho é M pares de bases. A Figura 3.15 representa três ilhas (a), (b) e (c) formadas pela união de três, dois e um fragmento, respectivamente, remontadas a partir do seqüenciamento de fragmentos randômicos de uma biblioteca de clones, bem como os oceanos existentes entre elas.

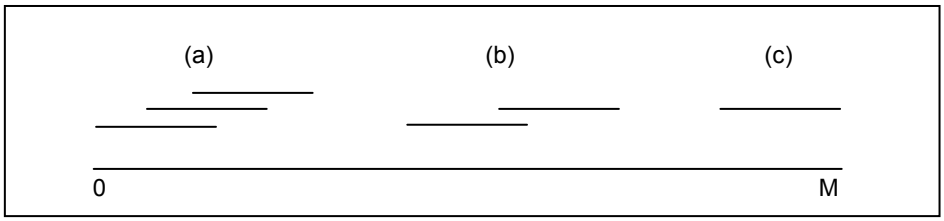


Figura 3.15. Ilhas e Oceanos. Ilhas formadas por: (a) três, (b) dois e (c) um fragmento. As regiões de espaços (*gaps*) entre as ilhas são os oceanos.

Supondo que a seqüência completa do organismo de interesse (genoma) esteja perfeitamente representada em uma biblioteca de clones, e que todos os fragmentos utilizados na construção da biblioteca tenham o mesmo tamanho, a localização dos clones randomicamente selecionados e seqüenciados [24] ao longo do genoma original foi modelada segundo o Processo de Poisson, para o qual as seguintes variáveis são definidas:

- M corresponde ao tamanho da seqüência genômica a ser determinada, em pares de bases;
- L corresponde ao tamanho dos clones;
- N corresponde ao número de clones;
- C = L∗N/M corresponde ao número esperado de clones que irão cobrir um determinado ponto randômico ao longo do genoma;
- Q corresponde ao tamanho necessário de sobreposição entre fragmentos para que estes sejam inseridos em uma mesma ilha;

[24] Como comentado anteriormente, clones são cópias de um mesmo fragmento de DNA.

– θ = Q/L corresponde à fração de sobreposição necessária para que dois fragmentos sejam unidos em uma mesma ilha.

Por simplicidade, assume-se que o tamanho dos clones seja 1, (L / L = 1) e, desta forma, o tamanho do genoma g passa a ser g = M / L e o número esperado de clones que cobrem um determinado ponto randômico c = N / g. O genoma completo encontra-se no intervalo (0, g). Assume-se que os fragmentos randômicos estão distribuídos ao longo do DNA genômico de acordo com a distribuição de Poisson (eq. (3.29)) com o parâmetro c; assim, a probabilidade de que existam k fragmentos cuja extremidade direita encontra-se no intervalo de tamanho t é dado por:

$$P(k) = \frac{e^{-ct} ct^{k}}{k!} \tag{3.29}$$

Alguns dos resultados desse modelo podem ser utilizados no contexto do DNA *shuffling*. Pode-se dizer que as ilhas representam regiões que podem ser remontadas pelo *shuffling* e os oceanos regiões que não podem ser remontadas. É possível assumir também que o tamanho esperado de uma ilha corresponde ao tamanho médio das regiões que serão remontadas durante o *shuffling*. Desta forma, os resultados apresentados no trabalho do Lander−Waterman foram reescritos no contexto do problema de DNA *shuffling* e estão apresentados no Teorema 1.

Teorema 1. Seja θ a fração de comprimento que dois fragmentos devem compartilhar a fim de parearem e, em seguida, serem estendidos por uma polimerase, N o número de fragmentos amostrados e c a cobertura destes fragmentos. Tem-se que:

(i) O número esperado de regiões que podem ser remontadas por meio do DNA *shuffling* é dado por:

$$N * \exp(-c(1-\theta)) \tag{3.30}$$

(ii) O número esperado de regiões remontadas a partir de j fragmentos (j ≥ 1) por meio de DNA *shuffling* é dado por:

$$N * \exp(-2c(1-\theta)) * (1-\exp(-c(1-\theta)))^{j-1} \tag{3.31}$$

(iii) O número esperado de regiões remontadas a partir de pelo menos dois fragmentos é dado por:

$$N * \exp(-c(1-\theta)) - N * \exp(-2c(1-\theta)) \tag{3.32}$$

(iv) O tamanho esperado de uma região remontada a partir do DNA *shuffling* (medido em pares de bases) é dado por:

$$\lambda = (\exp(c(1-\theta)) - 1)/c + \theta \tag{3.33}$$

Como se sabe, o principal objetivo do *shuffling* é recombinar, em um único fragmento, mutações presentes em parentais distintos. Por simplicidade, considere que as seqüências de DNA (fita simples) de dois parentais a serem submetidos ao *shuffling* diferem em apenas duas bases, ou seja, existem apenas duas mutações entre os parentais, ditas M_1 e M_2, e que essas mutações estão separadas por t pares de bases, como ilustrado na Figura 3.16.

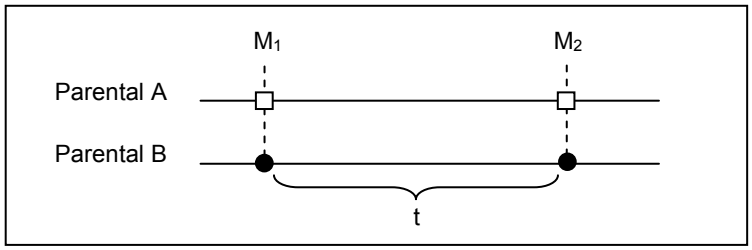


Figura 3.16. Parentais A e B, os quais diferem entre si em apenas duas bases, ditas mutações M_1 e M_2 que estão separadas por t bases.

A fim de obter moléculas com ambas as mutações M_1 e M_2, sendo cada uma delas originária de um parental distinto, é necessário que a região compreendida entre M_1 e M_2 seja remontada em um único fragmento durante o processo de DNA *shuffling*. A região compreendida entre essas duas mutações será chamada de região alvo e denotada por Tar (*Target*). O objetivo é determinar qual a probabilidade de que essa região seja remontada. Como comentado em Sun (1999), "infelizmente não existe uma fórmula explícita para o cálculo desta probabilidade". Apesar de não apresentada no trabalho, o autor afirma ter utilizado uma fórmula aproximada para o cálculo de tal probabilidade para determinados valores de c, θ e t. Um algoritmo foi proposto, porém não apresentado no trabalho, e os valores encontrados para a probabilidade de remontagem de uma determinada região Tar, sendo t = 25, c compreendido entre 4 e 10 e valores de θ igual a 0, 0,25 e 0,50 estão apresentados na Tabela 3.7.

Tabela 3.7. Probabilidade de que uma região alvo de tamanho t = 25 pb seja remontada pelo DNA *shuffling*.

	Cobertura c						
θ	4	5	6	7	8	9	10
0,00	0,15	0,44	0,69	0,86	0,94	0,98	0,99
0,25	0,00	0,01	0,18	0,41	0,62	0,77	0,87
0,50	0,00	0,00	0,00	0,00	0,02	0,08	0,19

A partir dos dados mostrados na Tabela 3.7 é possível verificar que a fração mínima θ de sobreposição necessária para que dois fragmentos se hibridizem e, em seguida, sejam estendidos assume um papel muito importante na probabilidade de que uma determinada região alvo seja remontada pelo processo de *shuffling*. Considerando, por exemplo, uma cobertura c igual a 10, a probabilidade de que uma região alvo cujo t = 25 seja remontada é de 0,99 quando uma fração mínima de sobreposição[25] entre os fragmentos é exigida comparada com apenas 0,19 quando se faz necessário que cerca de metade do comprimento dos fragmentos se sobreponham (θ = 0,50) para que o pareamento e a extensão ocorram.

Desta forma, a fim de garantir uma maior probabilidade de remontagem em uma determinada região alvo, as condições de um experimento de DNA *shuffling* devem ser tais que pequenas regiões de sobreposição entre fragmentos sejam necessárias para que o pareamento e a extensão ocorram. Baixas temperaturas de pareamento favorecem a ocorrência desse evento entre regiões complementares de menor tamanho, além de permitir que ocorram *mismatches* nessa região. Tal condição pode resultar em um número maior de mutações durante o processo de *shuffling*. A Figura 3.17 mostra uma região de pareamento entre dois fragmentos onde um *mismatch* ocorreu.

Figura 3.17. Exemplo de pareamento com *mismatch* (*).

[25] O autor não especifica qual o valor dessa fração mínima de sobreposição, contudo, a representa na Tabela 3.7 pelo valor 0,00.

Um balanço entre o aumento da probabilidade de ocorrência de pareamento nas regiões alvo e, ao mesmo tempo, a minimização da ocorrência de tais mutações deve ser considerado em experimentos de DNA *shuffling*.

É de interesse determinar também o número de fragmentos remontados que cobrem uma determinada região alvo Tar. Considerando apenas os fragmentos na direção 5'→ 3', existem k fragmentos que cobrem a região alvo se, e somente se, existe uma sobreposição de tamanho mínimo θ entre cada par de fragmentos f_i e f_{i+1}, sendo $1 \leq i \leq k - 1$, e não existe nenhum outro fragmento que se sobreponha ao fragmento f_k com tamanho mínimo θ. A Figura 3.18 mostra a remontagem por k = 4 fragmentos da região alvo compreendida entre duas mutações M_1 e M_2, separadas por t pares de bases.

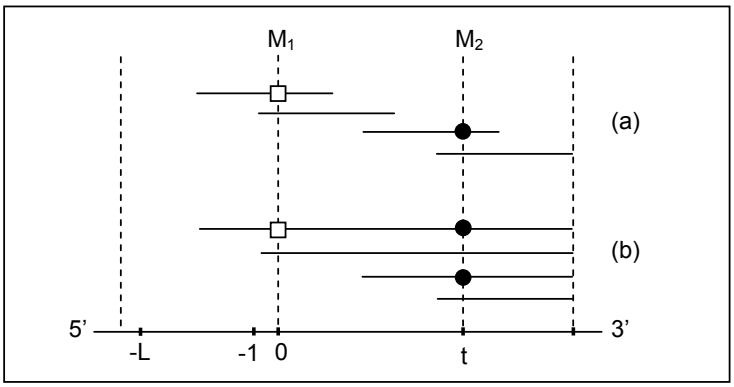


Figura 3.18. Esquema de remontagem dos fragmentos. (a) Sobreposição entre fragmentos com extremidades complementares. (b) Após a extensão das extremidades 3' dos fragmentos complementares, tem-se uma ilha.

Como demonstrado em Lander−Waterman, é possível afirmar que, dada uma região alvo específica, a probabilidade de que essa região seja remontada resultando em uma ilha composta de k fragmentos é dada por.

$$\exp(-c(1-\theta))(1-\exp(-c(1-\theta)))^{k-1} \tag{3.34}$$

3.5.3 O modelo de recombinação

Na seção anterior foi descrito um modelo para estimar a distribuição randômica de fragmentos ao longo de um genoma bem como as regiões que podem ser remontadas por meio do DNA *shuffling* deste genoma. Até então, as diferenças entre organismos não foram consideradas. Porém, uma das principais utilizações do DNA *shuffling* é recombinar mutações presentes em organismos

distintos a fim de obter moléculas híbridas. O modelo apresentado nesta seção descreve a recombinação entre fragmentos originários de DNA de diferentes espécies.

Sejam duas seqüências parentais A e B, originárias de duas espécies distintas 1 e 2, cujas seqüências de DNA diferem em apenas dois pares de bases, representadas por M_1 e M_2 e que a distância entre elas seja t pares de bases, como representado anteriormente na Figura 3.16. Seja a fração dos fragmentos originários da espécie 1 representada por α e da espécie 2 representada por $1 - \alpha$. Um fragmento randomicamente amostrado pode se parear com um fragmentos da espécie 1 com probabilidade α e com um fragmento da espécie 2 com probabilidade $1 - \alpha$. Se denotarmos por 0 a localização da mutação M_1 e por t a localização da mutação M_2 (como mostrado na Figura 3.18), um determinado fragmento remontado que cobre a região (0, t) terá ambas as mutações com probabilidade p_2, uma das mutações com probabilidade p_1, e nenhuma mutação com probabilidade p_0, sendo p_2, p_1 e p_0 dadas pelas equações (3.35), (3.36) e (3.37), respectivamente.

$$p_2 = \alpha(1-\alpha) \tag{3.35}$$

$$p_1 = (1 - p_0 - p_2) = (1 - 2p_2) = (1 - 2(\alpha(1-\alpha))) = a^2 + (1-\alpha)^2 \tag{3.36}$$

$$p_0 = p_2 \tag{3.37}$$

Observe que, quando o tamanho L do fragmento é maior que a distância t que separa as duas mutações, M_1 e M_2 podem não ser recombinadas independentemente, assim, nessas condições, a fração de moléculas resultantes do *shuffling* contendo ambas mutações pertencentes a um mesmo parental irá aumentar, enquanto que fração de moléculas remontadas cujas mutações consecutivas são originárias de um mesmo parenta irá diminuir caso o tamanho L dos fragmtnso utilizados na remontagem seja < t. Tais situações estão esquematizadas na Figura 3.19.

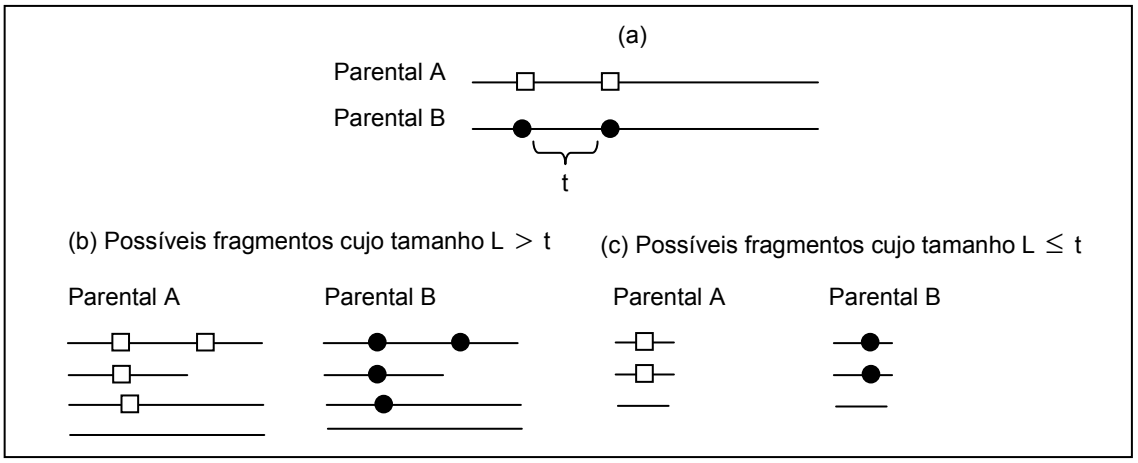


Figura 3.19. Distintos tamanhos de L. (a) Dois parentais A e B que diferem em dois pares de bases distantes entre si por t pares de bases. (b) Fragmentos resultantes da fragmentação enzimática cujo tamanho é L > t. (c) Fragmentos resultantes da ação enzimática cujo tamanho é L ≤ t.

Considerando fragmentos de tamanho L qualquer, novas probabilidades para o número de mutações presentes em um fragmento remontado devem ser calculadas. Com base no esquema apresentado na Figura 3.19, temos que, a fim de possuir as duas mutações M_1 e M_2 (originárias de parentais diferentes) em um fragmento remontado, essa remontagem deve iniciar-se com um fragmento contendo M_1, cuja probabilidade é α, e sua extremidade 5' deve estar no intervalo (−L, t − L) (cuja probabilidade é t / L), e o fragmento a se parear com este último deve carregar a mutação M_2 (probabilidade $1 - \alpha$). Assim, as probabilidades p_2, p_1 e p_0 descritas anteriormente pelas equações (3.35), (3.36) e (3.37), devem ser reescritas como mostrado nas equações (3.38), (3.39) e (3.40), respectivamente.

$$p_2 = t\alpha(1-\alpha)/L \tag{3.38}$$

$$p_1 = (1 - p_0 - p_2) = (1 - 2p_2) = 1 - 2(t\alpha(1-\alpha)/L) = 1 - (2t\alpha(1-\alpha))/L \tag{3.39}$$

$$p_0 = p_2 \tag{3.40}$$

Considerando ainda as duas espécies de DNA de diferentes origens que diferem entre si em dois pares de bases[26], e que ambas as mutações são benéficas em relação à função desempenhada por cada uma das moléculas de DNA, é de interesse que o *shuffling* entre essas seqüências resulte em moléculas que apresentem em suas seqüências ambas as mutações. Assim, o cálculo da probabilidade de que a biblioteca resultante contenha ao menos uma de tais moléculas é de interesse.

[26] Duas mutações.

Seja F a variável que descreve a existência, na biblioteca resultante do *shuffling*, de pelo menos uma seqüência cuja cobertura se encontra na região alvo e F_k a variável que descreve a existência de k seqüências desse tipo. A variável E descreve a ocorrência de pelo menos uma seqüência remontada na qual as duas mutações estão presentes. O objetivo então é estimar a probabilidade P(E).

Da eq. (3.34) tem-se que a probabilidade de ocorrência do evento $F_{k,}$ que é condicionada a ocorrência do evento F, é dada pela eq. (3.41) e a probabilidade de ocorrência do evento E, que também é condicionada a ocorrência de F_k é dada pela eq. (3.42).

$$P(F_k \mid F) = \exp(-c(1-\theta))(1-\exp(-c(1-\theta)))^{k-1} \tag{3.41}$$

$$P(E \mid F_k) = 1 - p_2^k, \text{ para } k = 1,2,3,... \tag{3.42}$$

Desta forma, tem-se que a probabilidade P(E) é calculada como mostra a eq. (3.43).

$$P(E) = \sum_{k=1} P(E \mid F_k) \tag{3.43}$$

Uma comparação com os resultados práticos do experimento realizado por Stemmer (1994b) e os resultados produzidos pelo modelo descrito, confirmam que o modelo é capaz de encontrar valores muito próximos aos encontrados experimentalmente. Um dos experimentos descritos em Stemmer (1994b) foi o *shuffling* entre duas seqüência de 1 kb (1.000 pares de bases) que contém o gene codificador da lacZα. Após a fragmentação com a enzima DNase I, apenas fragmentos de tamanho variando entre 10 e 50 pares de bases foram utilizados para a remontagem. O objetivo do *shuffling* era unir em um único fragmento duas regiões distintas (M_1 e M_2), as quais estão separadas por 75 pb em ambas as seqüências. A taxa de seqüências remontadas que possuíam ambas as mutações foi de 24%, contra um valor de 25% predito pelo modelo de recombinação de fragmentos descrito nesta seção.

3.6 Considerações Finais

Foram encontradas na literatura quatro propostas de modelos para o processo de DNA *shuffling*: o modelo proposto por Patrick et al. (2003); o modelo proposto por Maheshri e Schaffer (2003); o modelo proposto por Moore et al. (2001); e o modelo proposto por Sun (1999). Os modelos segundo Patrick, Maheshri–Schaffer e Moore foram implementados e disponibilizados pelos

softwares DRIVeR, ShuffIt e eShuffle, respectivamente. Nenhum software foi implementado para viabilizar o modelo proposto por Sun. Cada um dos modelos detalhados neste capítulo utiliza-se de abordagens distintas para modelar as diferentes etapas do processo de DNA *shuffling*. Contudo, todos os modelos têm o objetivo comum de estimar a diversidade da biblioteca resultante do *shuffling* entre seqüências parentais por meio do número médio de cruzamentos nas seqüências resultantes.

A descrição/detalhamentos dos modelos, além de proporcionar o melhor entendimento das características e abordagem utilizadas por cada um deles, colaborou para que uma descrição detalhada da utilização dos softwares fosse feita, além de permitir que dois deles, o eShuffle e o DRIVeR, fossem reescritos na linguagem de programação C e uma interface gráfica para a utilização de cada um deles fosse adicionalmente implementada, como apresentado no Capítulo 4.

4 Capítulo

Proposta e Implementação do ISAS, uma Ferramenta Computacional para Apoio ao Processo *in vitro* de DNA *shuffling*

"A dúvida é o principio da sabedoria."
Aristóteles

4.1 Introdução

Como visto no Capítulo 2, dentre as inúmeras metodologias descritas para a evolução molecular direta de moléculas em laboratório, o DNA *shuffling* tem sido utilizado com sucesso em diversos experimentos descritos na literatura (WANG et al., 2007), (NI et al., 2005), (OLIVA, 2004), (FRED et al., 1999), (CHANG et al., 1999) e (NESS et al., 1999).

A determinação do tamanho dos fragmentos, número de ciclos de PCR, tempo de cada ciclo de PCR, temperatura de pareamento e demais condições nas quais um determinado experimento de DNA *shuffling* deve ser realizado, a fim de garantir o maior número possível de recombinantes, na maioria das vezes, fica por conta de um especialista humano e é feito de forma empírica. Em contrapartida, podem ser encontradas na literatura algumas modelagens computacionais que visam otimizar o processo ou, pelo menos, algumas de suas etapas, a fim de aumentar sua eficiência, resultando assim em bibliotecas com maior diversidade genética.

Com o objetivo de minimizar o tempo e recursos gastos em experimentos de DNA *shuffling* e maximizar a qualidade dos resultados obtidos, a simulação *in silico* de tal experimento pode ser vista como uma ferramenta de grande importância uma vez que, por meio dela, é possível

verificar (entre outros) a influência de parâmetros como similaridade entre as seqüências parentais, distância entre as mutações presentes nos parentais e tamanho dos fragmentos a serem recombinados, sobre os resultados esperados do *shuffling*, produzindo assim informações e otimizações a serem empregadas quando da realização *in vitro* do DNA *shuffling*.

A ferramenta ISAS – *Interactive Software for Assisting DNA Shuffling Processes*[27], proposta e implementada durante o trabalho de pesquisa realizado, é um sistema computacional interativo que tem como objetivo principal auxiliar usuários que pretendem conduzir experimentos de DNA *shuffling* bem como aqueles que já realizaram tal experimento e necessitam analisar os resultados obtidos. Dentre as diversas funcionalidades implementadas e disponibilizadas, o ISAS permite a análise da adequabilidade de seqüências candidatas a seqüências parentais a serem utilizadas em experimentos de DNA *shuffling*, auxilia na análise da biblioteca resultante a fim de facilitar a identificação de seqüências recombinantes, além de disponibilizar três ferramentas para a simulação *in silico* de experimentos de DNA *shuffling*.

4.2 O sistema ISAS – arquitetura e principais funcionalidades

O sistema ISAS é constituído por três subsistemas: 1) *Sequence Basics*, 2) *Pre Shuffling* e 3) *Post Shuffling*, cujas funcionalidades são apresentadas na Figura 4.1.

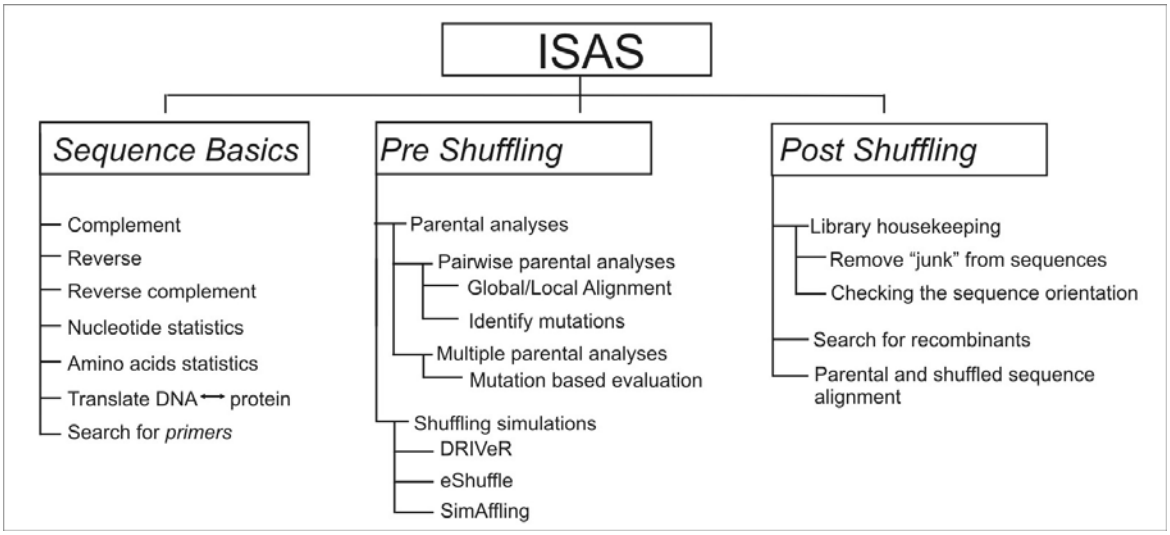


Figura 4.1. Arquitetura do sistema ISAS e principais funcionalidades.

[27] Optou-se por manter alguns dos nomes em inglês para evitar a duplicação de trabalho – alguns resultados deste trabalho de doutorado já foram publicados e submetidos para publicação em revistas da área ((MONTERA; NICOLETTI, 2008), (MONTERA et al., 2008b), (MONTERA et al., 2006), (MONTERA et al., 2008c)).

Para o desenvolvimento do ISAS foi adotado uma abordagem modular com vistas a facilitar sua expansão, por meio da incorporação de novas funcionalidades, à medida que o trabalho progredia. Cada um dos subsistemas *Sequence Basics*, *Pre Shuffling* e *Post Shuffling* são descritos nas seções 4.2.1, 4.2.2 e 4.2.3 respectivamente.

4.2.1 O subsistema *Sequence Basics*

O subsistema *Sequence Basics* agrupa um conjunto de procedimentos relacionados, principalmente, à análise de seqüências de DNA. Esse subsistema, entretanto, não implementa nenhuma inovação relacionada à manipulação e ao tratamento de seqüências de DNA.

A descrição do subsistema *Sequence Basics* é feita a seguir, tendo como referência a sua tela principal, mostrada na Figura 4.2. Cada um dos botões disponíveis na barra superior da tela manipula informações arquivo ↔ memória, como segue:

- *Open Sequence File* → abre o arquivo de entrada contendo a seqüência de DNA a ser analisada;
- *Past from Clipboard* → permite que o usuário transfira uma seqüência copiada de um outro aplicativo ou arquivo;
- *Clear* → remove a seqüência de DNA atual;
- *Save to File* → permite que o usuário salve em arquivo a seqüência de DNA.

A seqüência de entrada para esse subsistema deve estar no formato FASTA[28] ou texto ASCII. O subsistema automaticamente verifica a consistência dos dados, analisando se a seqüência de DNA é uma seqüência válida, ou seja, se a seqüência informada é composta apenas por caracteres do alfabeto {A, C, G, T}. Caso não seja uma seqüência válida, uma mensagem de erro aparece na tela.

[28] O formato FASTA exige que seqüência de DNA ou proteína se inicie com o símbolo '>' seguida do identificador da seqüência, como pode ser observado na Figura 4.6.

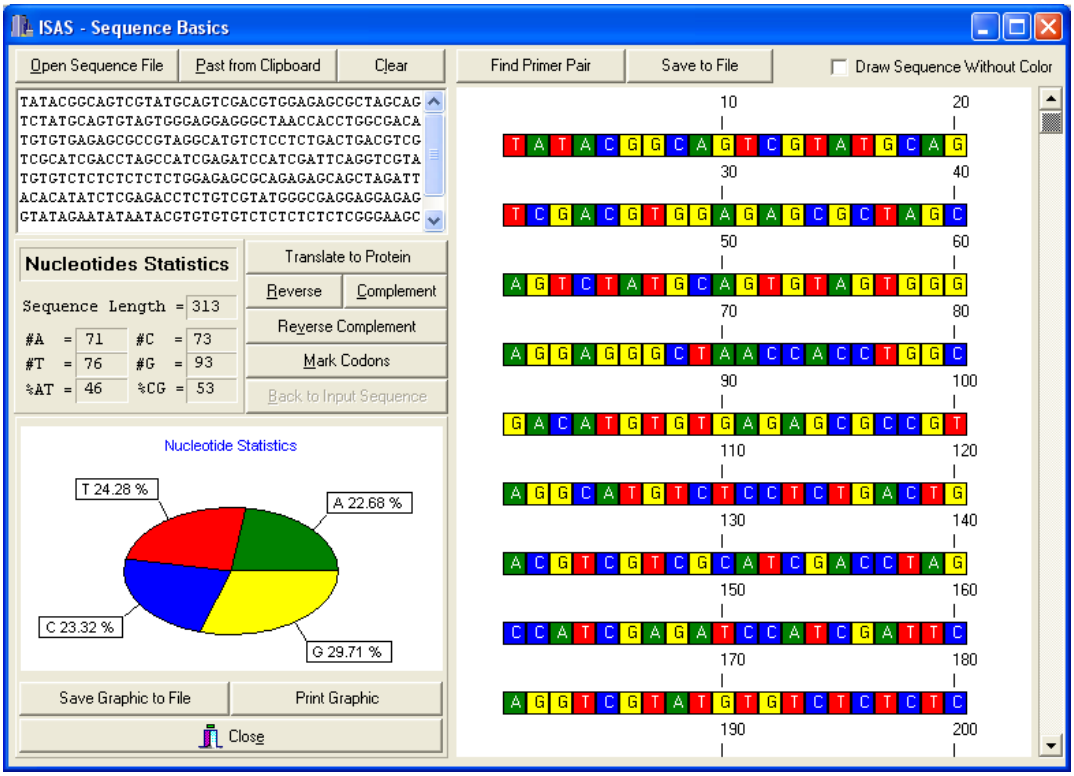


Figura 4.2. Subsistema *Sequence Basics* e suas funcionalidades. A cada uma das quatro bases que compõem a seqüência de DNA é atribuída uma cor característica para auxiliar na visualização da composição da seqüência.

Para uma dada seqüência de DNA, diversas funcionalidades são disponibilizadas, as quais podem ser ser acionadas via os botões:

- *Translate to Protein* → traduz a seqüência de DNA na seqüência de aminoácidos correspondente;
- *Reverse* → calcula o reverso da seqüência de DNA;
- *Complement* → calcula o complemento da seqüência de DNA;
- *Reverse Complement* → calcula o complemento reverso da seqüência de DNA;
- *Mark codons* → destaca cada um dos códons que compõem a seqüência de DNA com uma cor diferente;
- *Back to Input Sequence* → redesenha a seqüência de DNA originalmente informada pelo usuário;
- *Find Primer Pair* → encontra um possível par de *primers* a ser utilizado na amplificação da seqüência de DNA fornecida.

A Figura 4.3 mostra a mesma seqüência de DNA apresentada na Figura 4.2, porém agora os códons que a compõem encontram-se destacados em cores distintas e não mais suas bases. Esta marcação é resultado da ativação do botão *Mark Codons*.

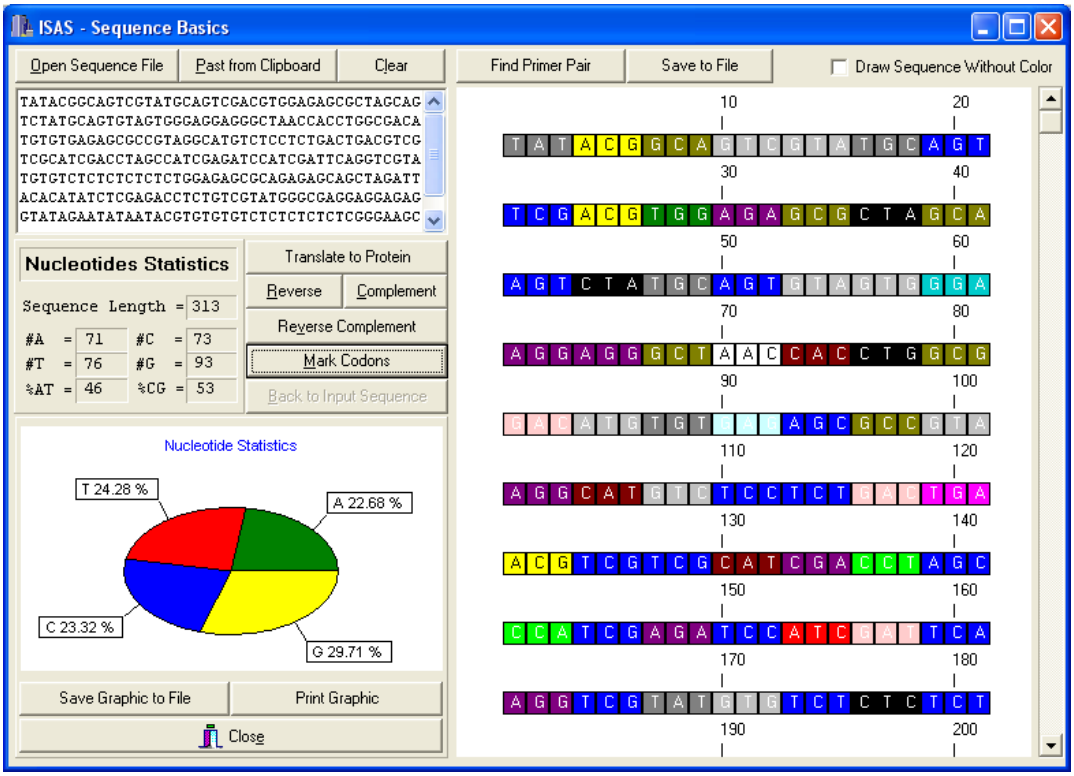


Figura 4.3. Subsistema *Sequence Basics*. Seqüência de DNA com os códons em destaque.

O subsistema *Sequence Basics* implementa também um conjunto de métricas para a análise de seqüências de DNA ou proteína. Para uma dada seqüência de DNA, o sistema automaticamente calcula o tamanho da seqüência, o número e a porcentagem de cada nucleotídeo bem como gera um gráfico de pizza para facilitar a visualização dos valores calculados para essas métricas. Caso a seqüência de DNA tenha sido traduzida em proteína, situação representada na Figura 4.4, tanto a freqüência de cada códon quanto de cada aminoácido na seqüência são apresentadas ao usuário em forma de tabelas. Os dados de cada uma das tabelas podem ser representados graficamente (botão *Draw Graphic*) e tais gráficos podem ser salvos em arquivos (botão *Save Graphic to File*) ou enviados para uma impressora (botão *Print Graphic*).

A apresentação da seqüência pode ser feita também na ausência de cores. Para tal, a opção *Draw Sequence Without Color* deve ser marcada pelo usuário. Porém, quando esta opção é selecionada, o botão *Mark Codons* fica desabilitado.

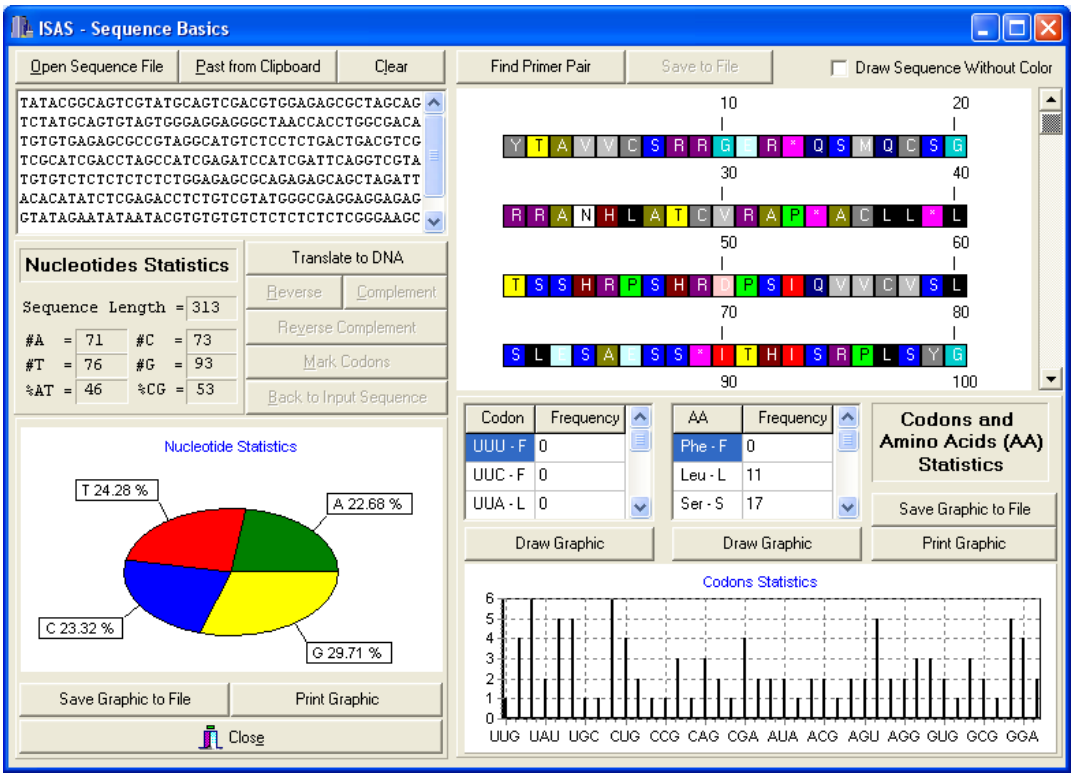


Figura 4.4. Subsistema *Sequence Basics*. Estatíticas dos códons e aminoácidos que compõem a seqüência.

A implementação da função de determinação do par de *primers* para a amplificação da seqüência de DNA (botão *Find Primer Pair*) foi realizada segundo os princípios do algoritmo de *Simulated Annealing* (SA). O algoritmo SA é muito utilizado para resolver problemas de otimização ((MONTERA; NICOLETTI, 2008), (MONTERA et al., 2008a), (TAHERI; ZOMAYA, 2007), (RODRIGUES, ZHANG, 2006)), como é o caso, por exemplo, do problema de determinação de par de *primers* para a amplificação, uma vez que a determinação de pares de *primers* envolve a validação de diversos parâmentros como, tamanho, composição (% de bases C e G), especificidade, entre outros. Detalhes sobre as questões envolvidas na determinação de *primers*, bem como sobre o algoritmo implementado estão descritos em (MONTERA; NICOLETTI, 2008) e resumidos no Apêndice D. A Figura 4.5 mostra a tela de execução resultante da busca por *primers* acionada pelo botão *Find Primer Pair*.

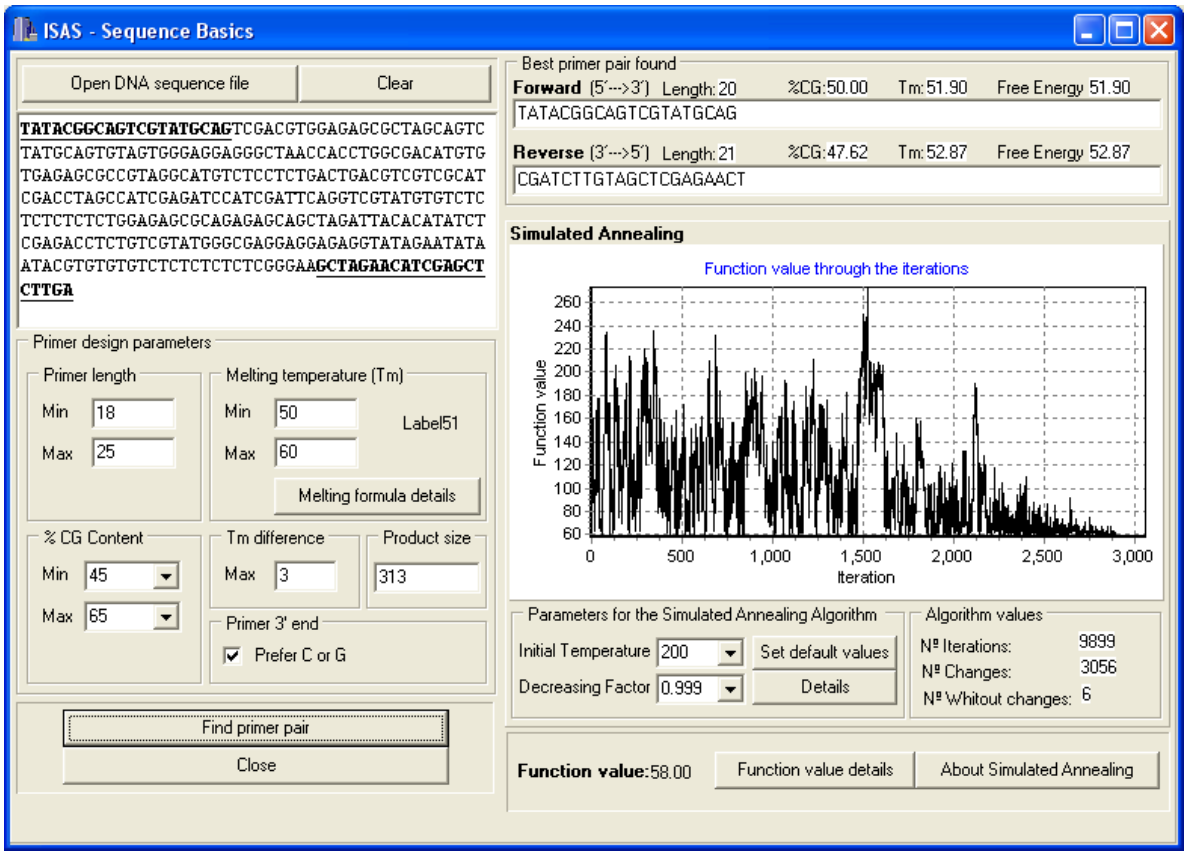


Figura 4.5. Busca por *primers*. Tela de execução da função que, dada uma seqüência de DNA, encontra um par de *primers* para a amplificação da mesma. O gráfico *Function value through the iteration* representa o valor associado à adequabilidade dos pares de *primers* encontrados ao logo da busca.

4.2.2 O subsistema *Pre Shuffling*

O subsistema *Pre Shuffling* agrupa um conjunto de funcionalidades associadas a cadeias de DNA, que precedem o processo de DNA *shuffling*, com o objetivo de coletar informações para inferir o sucesso (ou não) de um experimento *in vitro* de DNA *shuffling*. As funcionalidades disponibilizadas pelo subsistema *Pre Shuffling* estão divididas em dois grupos:

- O grupo das funções relativas à análise de duas seqüências parentais, bem como a análise de mútiplas seqüências, implementadas pelo módulo *Parental Analyses* e ativado via palhetas *Pairwise Parental Analyses* e *Multiple Parental Analyses*, respectivamente (ver Figura 4.7). Nesse módulo, são implementadas funções para avaliar a adequabilidade de seqüências de DNA como candidatas a parentais em experimentos de DNA *shuffling*;
- O grupo das funções relativas às simulações do processo de DNA *shuffling*, implementadas pelo módulo *Shuffling Simulations*, ativadas via palheta de mesmo nome (ver Figura 4.7). Nesse módulo, são disponibilizados modelos computacionais que podem ser utilizados para a predição de resultados de experimentos de DNA *shuffling*.

Os módulos *Parental Analyses* e *Shuffling Simulations* estão descritos em detalhes nas subseções 4.2.2.1 e 4.2.2.2, respectivamente.

4.2.2.1 *Parental Analyses*

Como descrito anteriormente, o objetivo do DNA *shuffling*, bem como de diversas outras técnicas de evolução *in vitro*, é recombinar em uma única seqüência mutações presentes em seqüências distintas, ditas parentais. Tais recombinações podem resultar em seqüências com funcionalidades melhoradas em relação aos parentais ou, ainda, em seqüências com novas funcionalidades.

No processo de DNA *shuffling*, os parentais são fragmentados e esses fragmentos remontados por meio de ciclos de PCR. Uma recombinação acontece quando fragmentos originários de parentais distintos e, que carregam mutações, unem-se com base em trechos complementares e são, em seguida, estendidos pela ação de uma polimerase (ver subseção 2.6.3). Como comentado por Volkov et al. (2000), um importante parâmetro para determinar a utilidade de qualquer método de evolução *in vitro* é o número de eventos de recombinação por gene – isto é, freqüência de cruzamento – que pode ser alcançado. Nesse sentido, uma análise prévia das seqüências parentais disponíveis à realização do *shuffling* pode resultar na escolha daquelas cuja freqüência de recombinação resultante seja a melhor possível, bem como na determinação das condições, como por exemplo, tamanho do fragmento e temperatura de pareamento, que podem viabilizar um maior número de recombinações entre os parentais escolhidos.

O modulo *Parental Analyses* implementa funcionalidades relativas à análise de duas seqüências parentais candidatas ao *shuffling*, bem como implementa uma medida proposta para avaliar a adequabilidade de um conjunto de $N > 2$ seqüências parentais disponíveis à realização de experimentos de DNA *shuffling*. O submódulo que avalia duas seqüências canditadas a parentais recebe o nome de *Pairwise Parental Analyses* enquanto que o submódulo que avalia a adequabilidade de múltiplas seqüências candidatas ao *shuffling* é chamado *Multiple Parental Analyses* e são descritos nas subseções 4.2.2.1.1 e 4.2.2.1.2, respectivamente.

4.2.2.1.1 *Pairwise Sequence Analyses*

Para a apresentação e discussão das funções implementadas pelo submódulo *Pairwise Sequence Analyses* são utilizadas as seqüências correspondentes a dois genes codificadores de cisteínas nomeadas Oryza (*oryzacystatin*, número de acesso no GenBank NM_190953) e SD (*sugarcane cystatin dubbed*, número de acesso no GenBank CA132601). As seqüências de DNA codificadoras dos genes Oryza e SD são apresentadas na Figura 4.6. Como comentado

anteriormente o formato do arquivo de entrada contendo as seqüências parentais deve ser o FASTA, como mostrado na Figura 4.6.

```
>Oryza
ATGTCGAGCGACGGAGGGCCGGTGCTTGGCGGCGTCGAGCCGGTGGGGAACGAGAACGACCTCCACCTCGTCGACCTC
GCCCGCTTCGCCGTCACCGAGCACAACAAGAAGGCCAATTCTCTGCTGGAGTTCGAGAAGCTTGTGAGTGTGAAGCAG
CAAGTTGTCGCTGGCACTTTGTACTATTTCACAATTGAGGTGAAGGAAGGGGATGCCAAGAAGCTCTATGAAGCTAAG
GTCTGGGAGAAACCATGGATGGACTTCAAGGAGCTCCAGGAGTTCAAGCCTGTCGATGCCAGTGCAAATGCCTAA
>SD
ATGGCGTTGGCCGGCGGCATCAAGGACGTGCCGGCGAACGAGAACGACCTCCACCTCCAGGAGCTCGCCCGCTTCGCC
GTCGATGAGCACAACAAAAAGGCCAATGCTCTTCTGGGGTACGAGAAGCTTGTGAAGGCCAAGACACAAGTAGTTGCT
GGCACGATGTACTATCTCACTGTTGAGGTGAAGGATGGCGAAGTCAAAAAGCTCTACGAAGCTAAGGTCTGGGAGAAG
CCATGGGAGAACTTCAAGGAGTTGCAAGAATTCAAGCCTGTTGAAGAGGGTGCTAGCGCCTAA
```

Figura 4.6. Seqüência de nucleotídeos de duas cistatinas, Oryza e SD, armazenadas no formato FASTA.

O primeiro passo para a realização da análise dos parentais é a construção de um alinhamento entre eles. Dois algoritmos clássicos para a construção do alinhamento foram implementados e disponibilizados para a escolha do usuário: o algoritmo de Needleman–Wunsch (NEEDLEMAN; WUNSCH, 1970) para a construção do alinhamento global ótimo[29] e o algoritmo de Smith–Waterman (SMITH; WATERMAN, 1981) para a construção do alinhamento local ótimo. No alinhamento global entre dois parentais, as seqüências inteiras são alinhadas, enquanto que no alinhamento local, apenas o trecho mais similar entre as seqüênciasé alinhado.

A Figura 4.7 mostra a tela referente ao subsistema *Pre Shuffling* na qual a palheta *Pairwise Parental Analyses* está ativada. Na figura é mostrada uma parte do alinhamento global entre Oryza e SD construído pela ferramenta. O arquivo de entrada contendo as duas seqüências parentais foi carregado em memória pela ativação do botão *Open Parental Sequence File* e o alinhamento construído pela ativação do botão *View Alignment*. No alinhamento, as colunas em branco indicam a ocorrência de *match*, as colunas em cinza claro a ocorrência de *mismatch*, enquanto que as colunas em cinza escuro representam a ocorrência um *gaps*. Um *match*, um *mismatch* e um *gap* indicam, respectivamente, o alinhamento de dois caracteres iguais, o alinhamento de dois caracteres distintos e o alinhamento entre um caractere e um espaço (representado pelo caractere '-').

[29] Um alinhamento é dito ótimo se ele possui a maior pontuação dentre todos os alinhamentos possíveis, como mostrado mais adiante no texto.

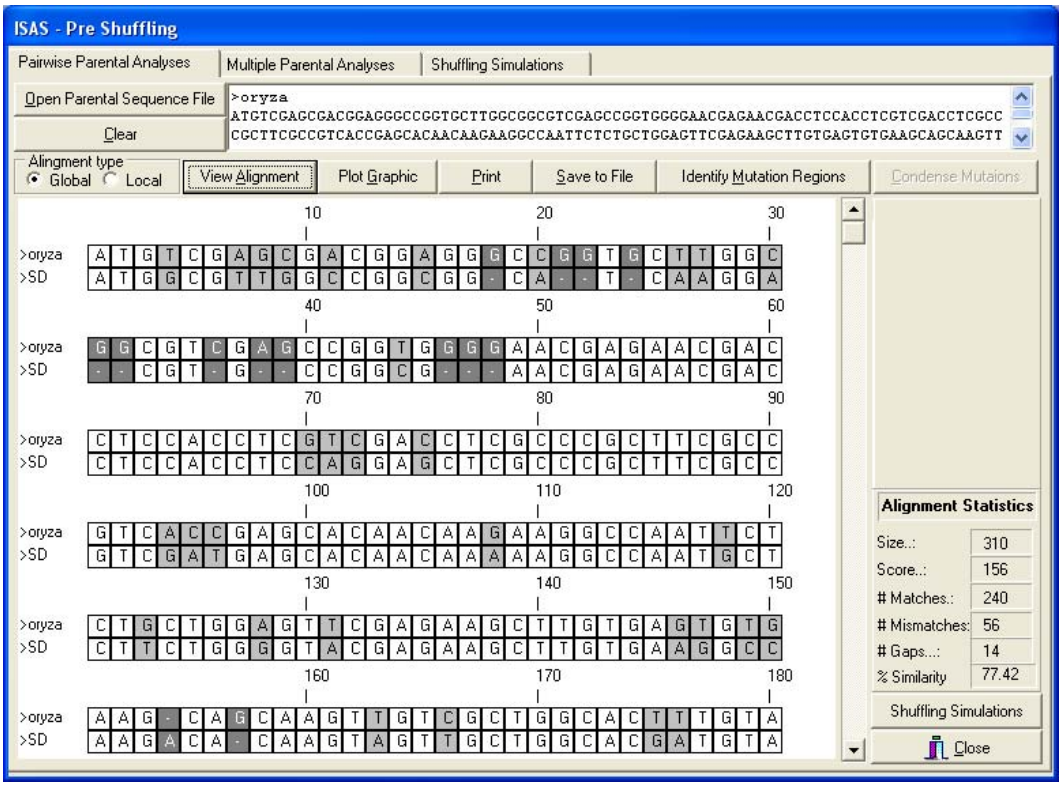


Figura 4.7. *Pairwise Parental Analyses*. Alinhamento entre as seqüências Oryza e SD.

Para o alinhamento construído entre duas seqüências, as seguintes estatísticas são automaticamente apresentadas no painel *Alignment Statistics*:

- *Size* → tamanho do alinhamento encontrado;
- *Score* → valor atribuído ao alinhamento;
- # *Matches* → número de *matches* no alinhamento;
- # *Mismatches* → número de *mismatches* no alinhamento;
- # *Gaps* → número de *gaps* no alinhamento;
- % *Similarity* → porcentagem de similaridade entre as seqüências.

Pela análise das estatísticas de um alinhamento é possível avaliar quão similares são duas seqüências. O *score*, ou pontuação, resume em um único valor estas estatísticas, visto que seu cálculo é dependente do número de *matches*, *mismatches* e *gaps* ocorridos no alinhamento, como estabelece a eq. (4.1):

$$\sum_{i=1}^{Size} \alpha(P1_i, P2_i) \tag{4.1}$$

onde *Size* corresponde ao tamanho do alinhamento, $P1_i$ e $P2_i$ correspondem, respectivamente, ao caractere do Parental 1 na posição i do alinhamento e ao caractere do Parental 2 nesta mesma posição e $\alpha(P1_i, P2_i)$ representa o "custo" de alinhar $P1_i$ e $P2_i$, dado pelo esquema de pontuação estabelecido na eq. (4.2).

$$\alpha(P1_i, P2_i) = \begin{cases} 1, \text{se } P1_i = P2_i \\ -1, \text{se } P1_i \neq P2_i \wedge P1_i \neq '-' \wedge P2_i \neq '-' \\ -2, \text{se } P1_i = '-' \vee P2_i = '-' \end{cases} \quad (4.2)$$

O esquema de pontuação definido pela eq. (4.2) é muito utilizado na prática (SETUBAL; MEIDANIS, 1997). Em se tratando de alinhamentos de seqüências de proteínas, matrizes de substituição de aminoácidos são utilizadas no cálculo do *score* (ex. matrizes da família PAM (DAYHOFF et al. 1978) e matrizes da família BLOSUM (HENIKOFF; HENIKOFF, 1992)). Os esquemas de pontuação utilizados para medir a similaridade entre duas seqüências penalizam a ocorrência de *gaps* e *mismatches* e premiam a ocorrência de *matches* (como o esquema de pontuação apresentado pela eq. (4.2)), de forma que, o alinhamento de maior *score* (alinhamento ótimo) dentre todos os possíveis alinhamentos entre as seqüências parentais é o que se deseja.

Uma vez construído o alinhamento, regiões de similaridades e de diferenças entre as seqüências parentais podem ser identificadas. As diferenças entre os parentais são chamadas de mutações entre eles. Uma mutação pode ser caracterizada como uma mutação única (um único par de bases) ou como um grupo de mutações consecutivas (um grupo de pares de bases consecutivos que diferem entre os parentais no alinhamento). A fim de identificar as mutações presentes entre os parentais alinhados, este trabalho propôs e adotou a seguinte regra: se duas ou mais mutações não estão separadas por um número mínimo de pares de bases iguais no alinhamento (*matches*), indicado pela variável NEBP (*Number of Equal Base Pairs*), elas serão agrupadas em uma única região de mutação. Sabe-se que mutações consecutivas que estão muito próximas umas das outras são difíceis de serem remontadas e, conseqüentemente, foi uma decisão de projeto agrupar essas mutações em uma única mutação. O valor *default* NEBP = 6 (o qual estabelece quão próximas devem estar duas mutações consecutivas a fim de serem consideradas uma única mutação) é adotado pelo ISAS. O usuário, entretanto, pode alterar este valor para melhor atender às condições experimentais (tais como tamanho do fragmento e temperatura de pareamento utilizados) ou mesmo para avaliar a influência deste parâmetro no número de mutações encontradas pelo software.

Ao clicar no botão *Identify Mutation Regions* um painel de informação aparece para explicar ao usuário como as mutações são agrupadas bem como para permitir que o valor de NEBP seja alterado.

Considerando o alinhamento entre Oryza e SD e o valor NEBP = 6, foram encontradas 14 mutações. Apenas três das quatorze mutações são compostas por um único par de bases. O restante das mutações é o resultado do agrupamento de mutações próximas e consecutivas. A Figura 4.8 mostra uma parte do alinhamento no qual as mutações encontradas estão destacadas na cor cinza. Após a identificação das mutações entre as seqüências parentais, um painel informando o número de mutações encontradas bem como a posição relativa de cada uma delas no alinhamento é apresentado ao usuário.

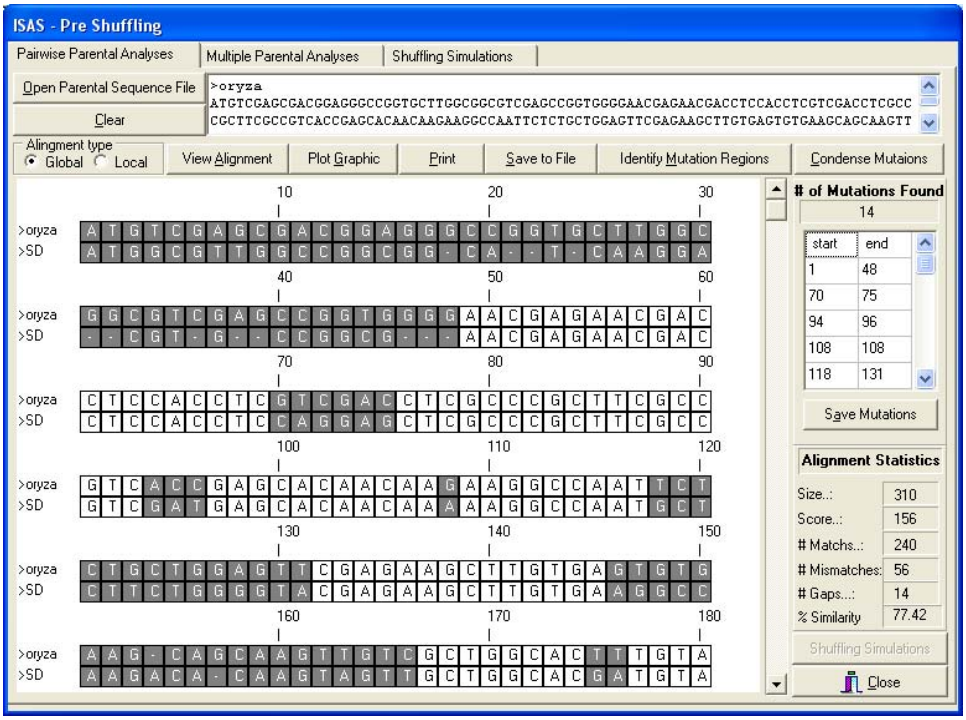


Figura 4.8. *Pairwise Parental Analyses*. Mutações encontradas entre os parentais Oryza e SD (destacadas em cinza).

É possível também visualizar no alinhamento cada uma das mutações encontradas sem levar em conta sua composição, ou seja, suas bases. O botão *Condense Mutations* aciona um processo que produz o alinhamento condensado, no qual cada uma das regiões de mutações encontradas ocupa apenas uma coluna do alinhamento. Com as regiões de mutação condensadas em um único ponto, fica mais fácil para o usuário visualizar as regiões onde cruzamentos entre as seqüências podem ocorrer (regiões de igualdade entre os parentais). A Figura 4.9 apresenta de

forma condensada as mutações encontradas entre os parentais Oryza e SD onde as bases que compõem as mutações são substituídas pelo caractere 'M' e estão assinaladas em cinza.

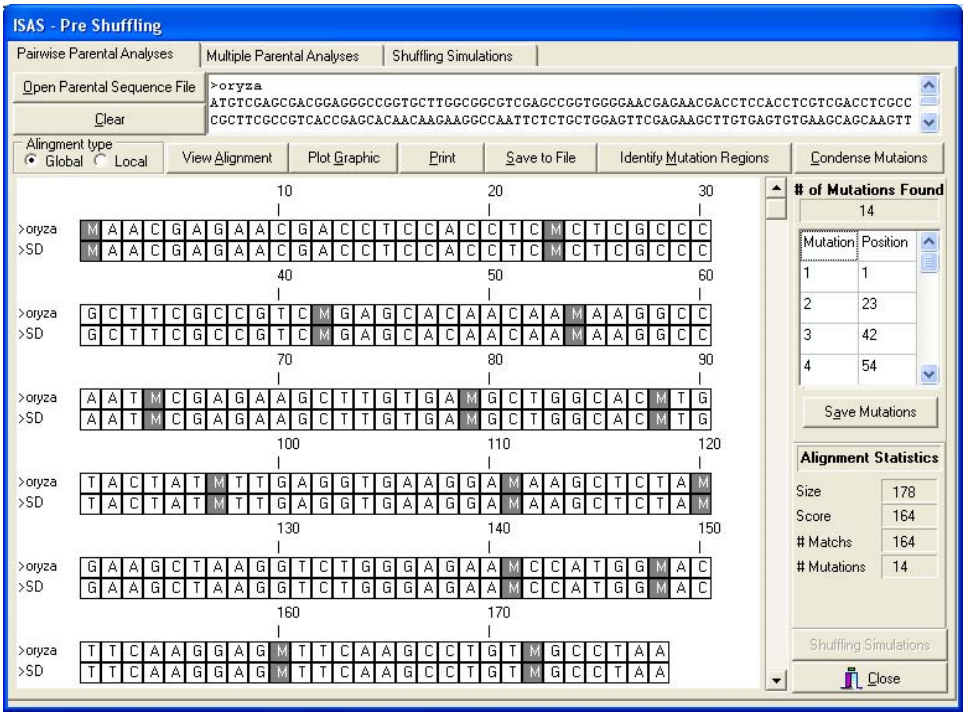


Figura 4.9. *Pairwise Parental Analyses*. Regiões de mutação entre os parentais representadas na forma condensada no alinhamento.

Espera-se que as mutações identificadas entre os parentais sejam encontradas nas seqüências resultantes do *shuffling* entre eles. Seqüências resultantes contendo mutações consecutivas pertencentes, originalmente, a parentais distintos são evidências da "mistura" dos parentais, ou seja, de que o experimento de DNA *shuffling* realizado foi bem sucedido e, nesses casos, diz-se que cruzamentos ocorreram entre os parentais. Assumindo que mutações consecutivas distantes uma das outras por menos do que seis pares de bases idênticos não podem ser eficientemente remontadas durante o *shuffling* entre os parentais Oryza e SD, espera-se que as seqüências resultantes contenham as mutações identificadas na Figura 4.8. Por exemplo, espera-se que toda seqüência resultante do *shuffling* entre Oryza e SD contenha a subcadeia GTCGAC ou CAGGAG, uma vez que estas representam uma mutação entre os parentais (ver o alinhamento mostrado na Figura 4.8, posições de 70 a 75).

É importante mencionar que mutações resultantes de um único par de bases ou da união de poucos pares de bases podem ser encontradas ao acaso nas seqüências resultantes quando a busca por mutações nas seqüências resultantes é executada automaticamente por um programa de computador (como será apresentado na Seção 4.2.3.2). A fim de otimizar a identificação desse

tipo de mutação (mutações pequenas), ISAS utiliza as bases vizinhas às mutações para melhor caracterizá-las e identificá-las. Desta forma, se uma mutação tem tamanho s (em número de pares de bases) menor que um valor mínimo n, então as n − s bases seguintes à mutação serão incorporadas à mutação. ISAS define por *default* n = 9, contudo, este valor pode ser alterado pelo usuário. As bases vizinhas necessárias a uma melhor caracterização das mutações pequenas serão adicionadas apenas no momento em que o usuário solicitar que o conjunto de mutações encontrado seja armazenado em arquivo (botão *Save Mutations*).

Considere o problema de encontrar uma subseqüência S de tamanho n pares de bases, em uma seqüência X, sendo ambas construídas utilizando-se o mesmo alfabeto {A, T, C, G}. Estatisticamente, a chance de S ser encontrada em uma seqüência X é dada pela eq. (4.3).

$$P(S) = \left(\frac{1}{4}\right)^n \tag{4.3}$$

Pela eq. (4.3) tem-se que, quanto maior for o tamanho n da subseqüência S, menor é a chance desta subseqüência ser encontrada ao acaso em uma seqüência qualquer.

As mutações identificadas entre os parentais devem ser salvas em arquivo, pois serão utilizadas na busca de seqüências recombinantes, como será descrito na Seção 4.2.3.2. A Tabela 4.1 descreve todas as mutações encontradas entre os parentais Oryza e SD utilizando NEPB = 6 e n = 9. As bases adicionadas a cada uma das mutações cujo tamanho é menor que n estão destacadas em negrito e sublinhadas.

4.2.2.1.2 *Multiple Sequence Analyses*

Como enfatizado anteriormente em vários pontos dessa tese, é fundamental que as seqüências a serem submetidas a um experimento de DNA *shuffling* sejam previamente avaliadas, com o intuito de verificar quão promissoras são para que o experimento seja bem sucedido. Características tais como similaridade e localização das mutações entre as seqüências parentais são fatores determinísticos nos resultados de experimentos deste tipo. Tais características devem ser utilizadas como guia para a definição e/ou estimativa de vários parâmetros envolvidos no processo (tais como tamanho dos fragmentos e temperatura de pareamento) com vistas ao seu sucesso. O sucesso de um processo de DNA *shuffling*, de certa forma, pode ser medido em função do número de recombinações obtidas a partir das seqüências parentais. Quanto maior for o

número de cruzamentos nas seqüências resultantes, mais bem sucedido pode ser considerado o experimento.

Tabela 4.1. Mutações encontradas entre os parentais Oryza e SD. Bases vizinhas foram adicionadas às mutações com tamanho menor que 9 (destacadas em negrito e sublinhadas).

Mutação	Mutações do parental Oryza	Mutações do parental SD
M_1	ATGTCGAGCGACGGAGGGCCGGTGCTT GGCGGCGTCGAGCCGGTGGGG	ATGGCGTTGGCCGGCGGCATCAAGGAC GTGCCGGCG
M_2	GTCGAC**CTC**	CAGGAG**CTC**
M_3	ACC**GAGCAC**	GAT**GAGCAC**
M_4	G**AAGGCCAA**	A**AAGGCCAA**
M_5	TCTCTGCTGGAGTT	GCTCTTCTGGGGTA
M_6	GTGTGAAGCAGCAAGTTGTC	AGGCCAAGACACAAGTAGTT
M_7	TT**TGTACTA**	GA**TGTACTA**
M_8	TTCACAA**TT**	CTCACTG**TT**
M_9	AGGGGATGCCAAG	TGGCGAAGTCAAA
M_{10}	T**GAAGCTAA**	C**GAAGCTAA**
M_{11}	A**CCATGGAT**	G**CCATGGGA**
M_{12}	ATGG**ACTTC**	GAGA**ACTTC**
M_{13}	CTCCAGGAG	TTGCAAGAA
M_{14}	CGATGCCAGTGCAAAT	TGAAGAGGGTGCTAGC

A dúvida associada à incerteza de quais seqüências escolher, dentre as seqüências parentais disponíveis para a realização de um experimento de DNA *shuffling*, foi a motivação para a investigação descrita nessa seção, que resultou na proposta de uma nova medida de adequabilidade (ao processo de DNA *shuffling*), associada a um par de seqüências.

Sabe-se que uma característica importante das seqüências a serem utilizadas como parentais é o fatos delas compartilhares regiões de igualdade entre seus pares de bases. Contudo, além dessa informação, a localização das regiões não similares (regiões de mutação) é também um fator determinante. No que segue é descrita a proposta de uma nova medida de similaridade, denominada medida baseada em mutações, que leva em consideração as características mencionadas anteriormente e que tem por objetivo avaliar a adequabilidade de seqüências candidatas a parentais, em um processo de DNA *shuffling*. A proposta da medida, bem como avaliações comparativas de seu uso (*versus* duas outras medidas) estão descritas em (MONTERA

et al., 2008b); A comparação da medida proposta com outras duas medidas bem como os resultados apresentados por Montera et al. (2008b) estão sumarizados no Apêndice E. A ferramenta ISAS disponibiliza um ambiente para a avaliação de múltiplas seqüências candidatas ao processo de DNA *shuffling* usando a métrica proposta na palheta de nome *Multiple Parental Analyses* como pode ser visto na Figura 4.10.

A medida baseada em mutações leva em consideração não apenas o número de mutações, mas também a distância entre mutações consecutivas existentes entre os parentais. A medida estabelece que a adequabilidade de duas seqüências candidatas ao *shuffling* é inversamente proporcional à distância média entre as mutações consecutivas identificadas entre os parentais. Dado um alinhamento entre duas seqüências X e Y, é possível identificar as regiões de igualdade e não-igualdade, ou mutações, existentes entre elas. Desta forma, por meio do alinhamento, é possível medir a similaridade entre duas seqüências. A medida de similaridade baseada em mutações estabelece uma relação entre as regiões de igualdade e não-igualdade entre as seqüências.

Inicialmente, o alinhamento entre as seqüências X e Y deve ser construído. O algoritmo utilizado para implementação do alinhamento ótimo foi o proposto por Needleman-Wunsch (1970). Uma vez construído o alinhamento, as mutações entre os parentais são identificadas em duas etapas distintas: inicialmente cada *gap*, bem como cada *mismatch* encontrado no alinhamento é considerado uma mutação. Em seguida, mutações consecutivas que não estejam distantes uma da outra por um número mínimo m de *matches* são agrupadas e tratadas como uma única mutação. A decisão de considerar um par de mutações que estão relativamente perto uma da outra como uma única mutação pode ser justificada pelo fato de que um cruzamento dificilmente ocorrerá em pequenas regiões de *matches*. Após o agrupamento das mutações, as distâncias n_i (medida em pares de bases) que separam duas mutações consecutivas M_i e M_{i+1}, para $1 \leq i \leq$ total de mutações – 1, e que corresponde ao número de *matches* existentes entre as mutações M_i e M_{i+1}, são determinadas.

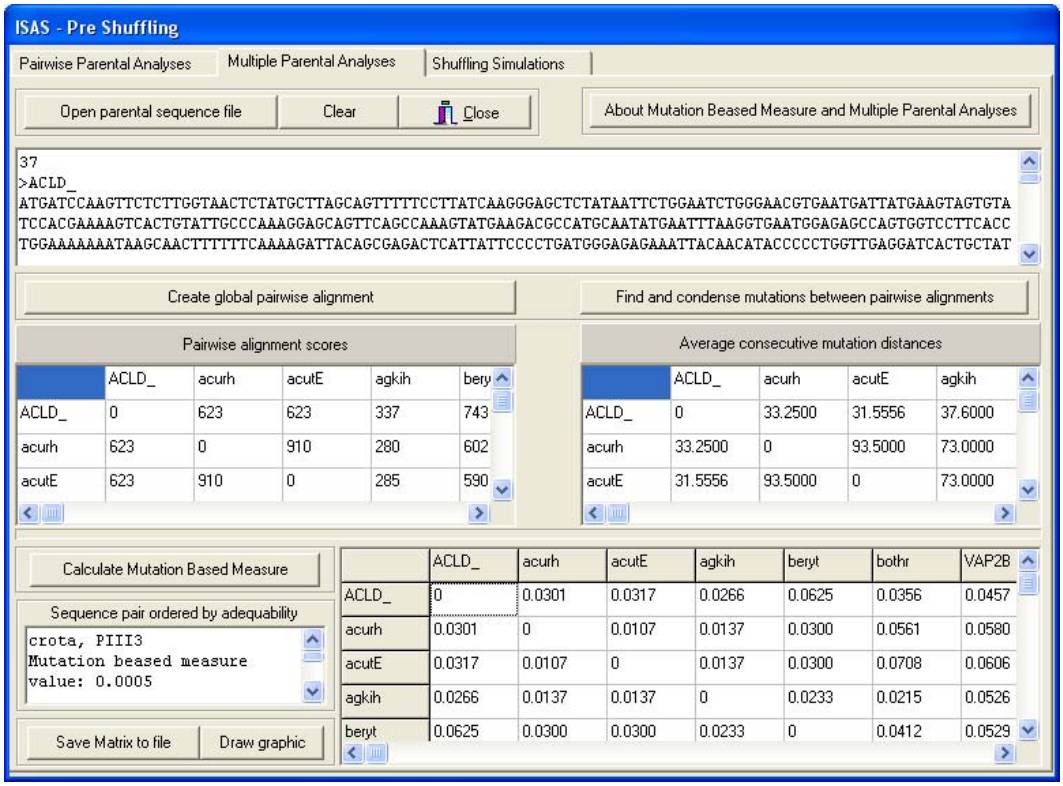
Dado um alinhamento entre as seqüências X e Y e, uma vez determinado o número de mutações entre duas seqüências, bem como a distância que separa cada duas mutações consecutivas, o cálculo da medida baseada em mutações é realizado como mostra a eq. (4.4),

$$S_{mut}(X,Y) = \frac{1}{\frac{\#mut - 1}{\frac{\sum_{i=1} n_i}{\#mut}}} = \frac{\#mut}{\#mut - 1} \sum_{i=1} n_i \tag{4.4}$$

na qual #mut representa o número de mutações existentes entre X e Y, com $0 \leq \#mut \leq |alinhamento|$, e $|alinhamento|$ representa o tamanho do alinhamento entre X e Y. Desta forma, dado um conjunto de p seqüências, a matriz de distância baseada em mutações entre todos os possíveis pares de seqüências é dada pela equação (4.5).

$$M_{mutação}[X][Y] = \begin{cases} S_{mut}(X,Y), \text{ se } X \neq Y \\ 0, \text{ caso contrário} \end{cases} \tag{4.5}$$

A investigação realizada com foco na determinação da adequabilidade de parentais para serem submetidos a experimentos de DNA *shuffling*, além de contribuir para um melhor entendimento do processo, resultou na proposta de uma nova medida. Diferente de outras medidas convencionais que avaliam duas (ou mais) seqüências com base em suas características evolutivas, a medida proposta tem foco apenas em experimentos de DNA *shuffling*. A Figura 4.10 apresenta a tela do subsistema *Pre Shuffling* com a palheta de *Multiple Parental Analyses* ativada. Os dados vistos na figura correspondem à determinação das medidas de distância para um conjunto de 37 seqüências de DNA codificadoras de metalopeptidases. Maiores detalhes sobre esta família de proteínas pode ser encontrado no Apêndice E.1.



	ACLD_	acurh	acutE	agkih	bery
ACLD_	0	623	623	337	743
acurh	623	0	910	280	602
acutE	623	910	0	285	590

	ACLD_	acurh	acutE	agkih
ACLD_	0	33.2500	31.5556	37.6000
acurh	33.2500	0	93.5000	73.0000
acutE	31.5556	93.5000	0	73.0000

	ACLD_	acurh	acutE	agkih	beryt	bothr	VAP2B
ACLD_	0	0.0301	0.0317	0.0266	0.0625	0.0356	0.0457
acurh	0.0301	0	0.0107	0.0137	0.0300	0.0561	0.0580
acutE	0.0317	0.0107	0	0.0137	0.0300	0.0708	0.0606
agkih	0.0266	0.0137	0.0137	0	0.0233	0.0215	0.0526
beryt	0.0625	0.0300	0.0300	0.0233	0	0.0412	0.0529

Figura 4.10. *Multiple Parental Analyses*. Análise de 37 seqüências candidatas a parentais para as quais foram calculadas a matriz de *score* (derivada dos alinhamentos), a matriz de distância média entre as mutações consecutivas e a matriz de distância baseada em mutações, para todos os pares de seqüências.

É possível também visualizar graficamente a lista de pares de seqüências ordenada pelo menor valor da medida de distância baseada em mutação, como mostra a Figura 4.11. A representação gráfica dá uma magnitude geral da diferença entre a adequabilidade dos pares de parentais avaliados. Vale lembrar que quanto maior o valor da medida de distância baseada em mutações para um par de seqüências, menor é sua adequabilidade em relação a experimentos de *shuffling*, uma vez que a medida é inversamente proporcional à distância que separa as mutações e, mutações muito próximas uma das outras são mais ineficientemente remontadas por este tipo de experimento.

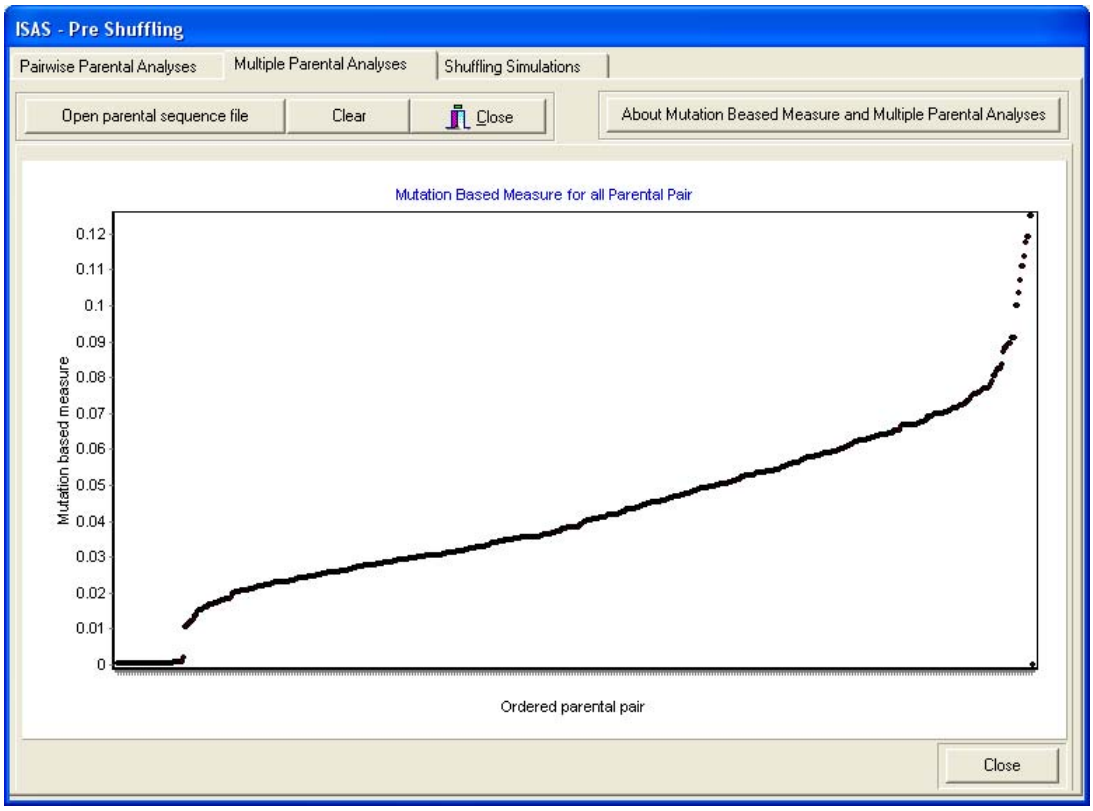


Figura 4.11. Representação gráfica da adequabilidade dos pares de seqüências avaliados segundo a medida baseada em mutações.

4.2.2.2 *Shuffling Simulations*

Embora o *shuffling* de DNA tenha sido utilizado com sucesso em muitos experimentos descritos na literatura, a obtenção de bons resultados, ou seja, a obtenção de uma quantidade satisfatória de seqüências *full-length* resultantes do maior número possível de cruzamentos entre os parentais, pode não ser um processo simples devido ao grande número de parâmetros envolvidos em experimentos deste tipo. A fim de melhorar a eficiência do experimento, otimizações são necessárias. Contudo, realizar tais otimizações é uma tarefa complexa que, segundo Maheshri e Schaffer (2003), depende de uma diversidade de parâmetros, tais como concentração e composição das seqüências de DNA a serem submetidas ao *shuffling*, tamanho dos fragmentos utilizados e condições nas quais os ciclos de PCR ocorrem (número de ciclos, tempo de duração de cada ciclo, temperatura, etc). Esses parâmetros influenciam diretamente nos resultados obtidos, dentre os quais Maheshri e Schaffer (2003) destacam:

- tamanho final dos fragmentos remontados;
- número e localização dos cruzamentos;

– fração dos fragmentos remontados que correspondem a seqüências com o mesmo tamanho dos fragmentos parentais (*full-length*);
– fração de fragmentos que darão origem a uma proteína com função melhorada.

Como discutido no Capítulo 3, existem na literatura algumas modelagens computacionais que objetivam colaborar com o processo de DNA *shuffling* ou, pelo menos, com algumas de suas etapas, a fim de aumentar a sua eficiência, resultando assim em bibliotecas com maior diversidade gênica. Os modelos propostos por Patrick et al. (2003), Maheshri e Schaffer (2003) e Moore et al. (2001) descritos no Capítulo 3, seções 3.2, 3.3 e 3.4 respectivamente, foram implementados pelos autores como os softwares DRIVeR, ShuffIt e eShuffle, respectivamente. Os softwares segundo os modelos de Patrick e colaboradores e Maheshri e Schaffer, foram obtidos dos endereços www.bio.cam.ac.uk/~blackburn/stats.html e http://www.cchem.berkeley.edu/~schaffer/*shuffling*, respectivamente. A implementação do modelo proposto por Moore e colaboradores foi obtida por meio de contado com um dos autores (informação pessoal)[30].

Em contato realizado com Fengzhu Sun (informação pessoal)[31], autor do modelo proposto em (SUN, 1999), ele informou que não dispunha de um software implementando o modelo descrito. A falta de algumas informações essenciais e a maneira vaga como alguns processos foram discutidos/propostos no modelo Sun contribuíram para tornar a implementação computacional do modelo inviável. Ao mesmo tempo em que o autor afirma que não existe uma fórmula explicita para o cálculo da probabilidade de remontagem de uma determinada região compreendida entre duas mutações consecutivas, o autor diz – apesar de não apresentar no trabalho – ter usado uma aproximação para o cálculo desta probabilidade (ver Seção 3.5).

Pelas dificuldades encontradas na execução do ShuffIt, bem como pelos motivos expostos no Apêndice B, a disponibilização do ShuffIt pelo subsistema *Shuffling Simulations* não foi realizada.

ISAS disponibiliza ao usuário uma interface amigável para a realização da simulação e/ou predição de resultados de experimentos de DNA *shuffling* segundo os modelos propostos por Patrick et al. (2003) e Moore et al. (2001) e implementados via software DRIVeR e eShuffle, respectivamente. Para viabilizar ambos os modelos DRIVeR e eShuffle via ISAS, o seguinte procedimento foi adotado:

[30] MOORE, G. About eShuffle. Mensagem recebida por lumontera@gmail.com em 16 de Maio de 2006.
[31] SUN, F. Modeling DNA Shuffling. Mensagem recebida por lumontera@gmail.com em 20 de Abril de 2007.

a) A descrição e análise dos fundamentos teóricos e resultados que subsidiam ambos os modelos foram cuidadosamente pesquisados e estudados, com base nas publicações (PATRICK et al., 2003) e (MOORE et al., 2001), descritas no Capítulo 3;
b) O código fonte de ambos os softwares, originalmente escritos em Fortran, foram analisados e estudados de maneira a abstrair o algoritmo correspondente a cada um deles que efetivamente representa a expressão procedural do modelo no qual cada um dos softwares está baseado;
c) Cada um dos algoritmos foi então programado na linguagem C, incorporando um conjunto de funcionalidades com vistas, principalmente, a facilitar a interação com o usuário e viabilizar uma melhor interpretação dos resultados.

Especificamente para o DRIVeR, a análise dos parentais (como descrita na Seção 4.2.2.1.1) fornece as informações referentes ao tamanho do alinhamento entre os parentais, número de mutações e posições relativas das mutações. O agrupamento em uma única região de mutações consecutivas muito próximas umas das outras observadas no alinhamento entre dois parentais é extremamente importante para a execução do DRIVeR uma vez que, devido à grande quantidade de memória necessária à sua execução, sua implementação limita o número de mutações entre os parentais em no máximo 20. Assim, ao realizar a busca por mutações entre dois parentais com o objetivo de simular os resultados do *shuffling* entre eles utilizando o DRIVeR, o usuário deve escolher um valor de NEPB para o qual o número de mutações identificadas não seja superior a 20 (quanto maior o valor de NEPB, menor será o número de mutações identificadas entre os parentais). Além dessas informações obtidas da análise dos parentais, o usuário deve informar ainda os valores dos parâmetros referentes ao número real de cruzamentos (λ^{true}) e o tamanho da biblioteca de DNA *shuffling* construída. De posse de todas essas informações ISAS gera automaticamente o arquivo de entrada necessário à execução do modelo DRIVeR, no formato apresentado no Apêndice A, Seção A.1.

A Figura 4.12 mostra como o ISAS disponibiliza a execução do DRIVeR. Além de visualizar o arquivo texto contendo os resultados obtidos pelo DRIVeR para o número esperado de variantes distintas (*Expected number of distintic variants*) e para o número observado de cruzamentos por seqüência resultante (*Observed number of crossovers per sequence*), o usuário também pode realizar a simulação para diferentes valores de número real de cruzamentos (*Number of real crossovers*) e diferentes tamanhos de biblioteca (*Library size*) e visualizar graficamente os resultados obtidos. Os resultados apresentados na Figura 4.12 consideram a análise prévia dos parentais

Oryza e SD (como mostrada na Seção 4.2.2.1) e os seguintes valores para o parâmetro tamanho da biblioteca (*Library Size)*: 500, 1.000, 1.500, 2.000 e 2.500 em duas execuções distintas, a primeira para num número real de cruzamentos igual a 2 e a segunda para número real de cruzamentos igual a 3.

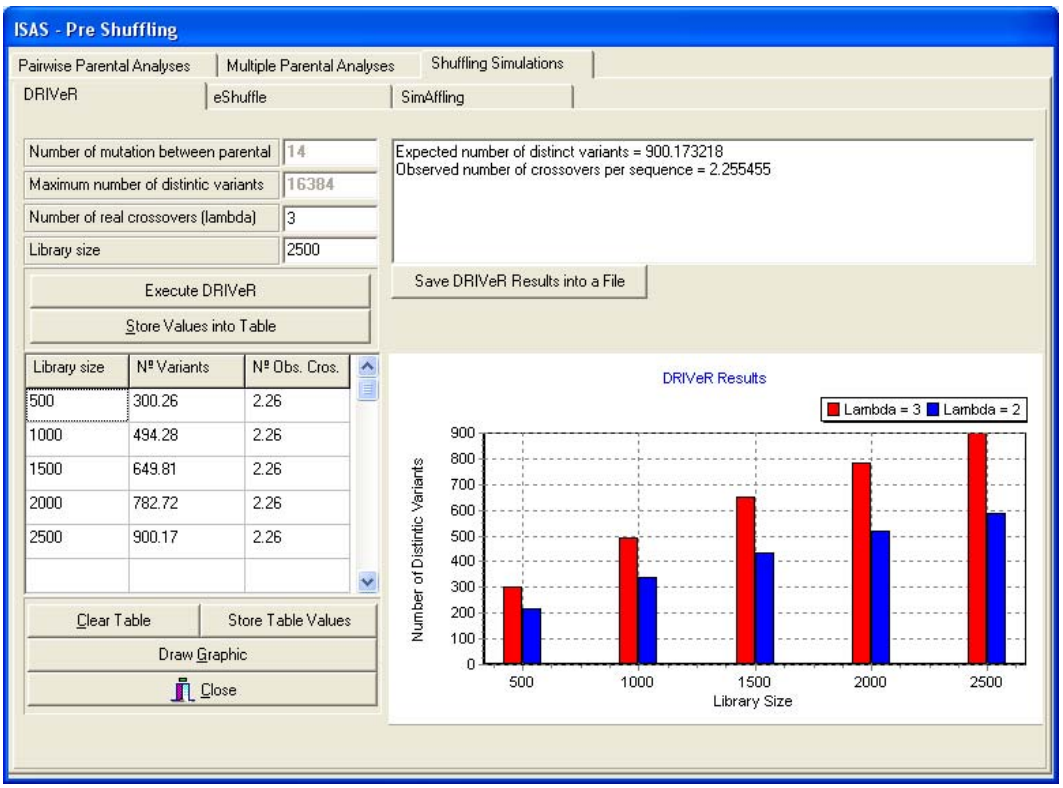


Figura 4.12. *Shuffling Simulations*. Simulações do processo de DNA *shuffling* entre os parentais Oryza e SD utilizando o DRIVeR.

Os valores para os parâmetros *Number of mutations between parental* e *Maximum number of distinct variants* que representam, respectivamente, o número de mutações entre os parentais e o número máximo de variantes distintas que podem ser obtidas ($2^M - 2$), não são informados pelo usuário e sim obtidos automaticamente da análise prévia dos parentais.

Para a execução do eShuffle, como visto anteriormente, é necessário que as seqüências correspondentes aos parentais, derivadas do alinhamento entre elas (isto para que as seqüências parentais tenham sempre o mesmo tamanho), sejam armazenadas em um arquivo de entrada, o qual deve respeitar um formato específico (ver Apêndice C, Seção C.1). Em um outro arquivo devem ser armazenadas as informações referentes ao tamanho do fragmento, a temperatura de pareamento, o tamanho do alinhamento entre os parentais, o número de parentais, e o nome do arquivo contendo as seqüências parentais. Com base nas informações fornecidas pelo usuário

(ver Figura 4.13) e na análise prévia dos parentais, ambos os arquivos necessários à execução do eShuffle são gerados automaticamente pelo ISAS.

Além das modificações realizadas no eShuffle para facilitar sua execução no que diz respeito aos parâmetros de entrada, o software foi implementado de maneira a permitir ao usuário escolher um dentre quatro conjuntos distintos de valores de ΔS e ΔH – para pares de bases sem *mismatches* – necessários ao cálculo da energia livre associada aos eventos de pareamentos. Além do conjunto de valores para ΔS e ΔH utilizado na implementação original do eShuffle (ALLAWI; SANTALUCIA, 1997), o usuário pode optar também pela utilização dos valores sugeridos por Sugimoto (1996), SantaLucia (1998) e Breslauer (1986). Independentemente da escolha do usuário, que diz respeito apenas aos valores de ΔS e ΔH para pares de bases onde não ocorrem *mismatches*, os valores de ΔS e ΔH para os pares de bases onde *mismatches* ocorrem são os valores apresentados na Tabela C.2 do Apêndice C.

A Figura 4.13, mostra o resultado da simulação do eShuffle considerando os parentais Oryza e SD, cujo alinhamento entre suas seqüências tem tamanho 310 pb, para uma temperatura de pareamento igual a 50ºC, fragmentos de tamanho 45 pb, utilizando os valores para ΔS e ΔH sugeridos por Allawi e SantaLucia (1997).

Adicionalmente à disponibilização/viabilização dos softwares eShuffle e DRIVeR pela ferramenta ISAS, um outro simulador para o processo de DNA *shuffling* foi implementado como proposta desta pesquisa. Esta implementação recebeu o nome de SimAffling e será detalhado no Capítulo 5.

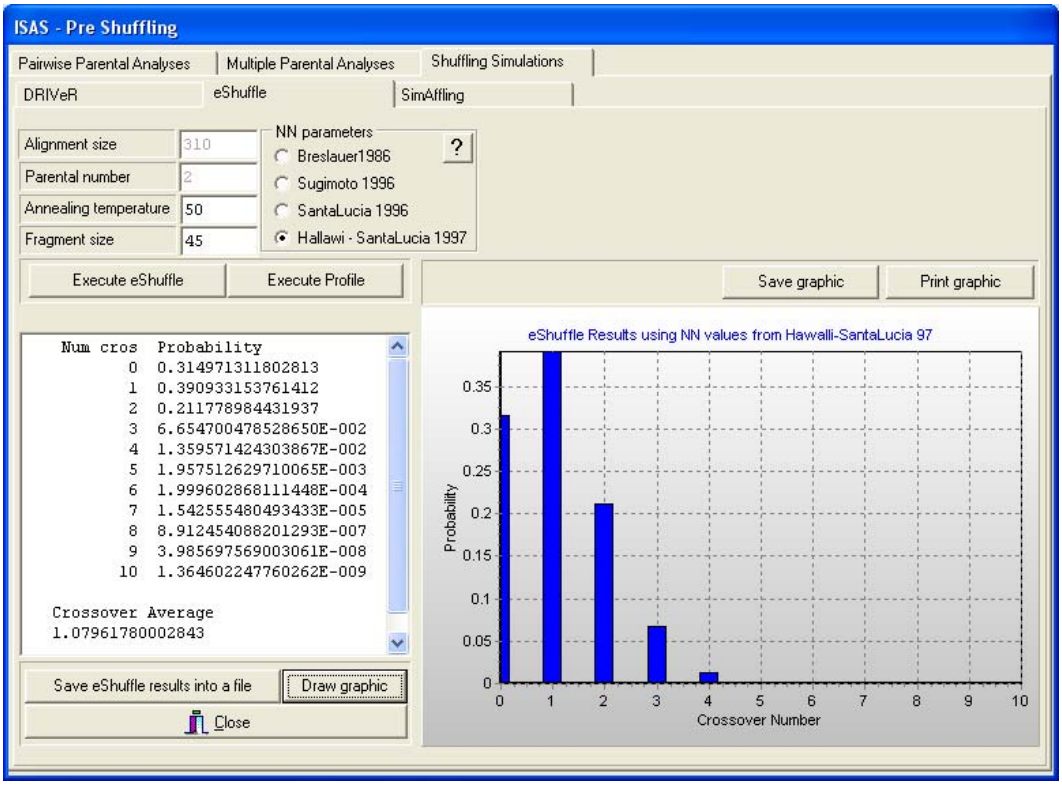


Figura 4.13. *Shuffling Simulations*. Simulação do processo de DNA *shuffling* entre os parentais Oryza e SD utilizando o eShuffle.

4.2.3 O subsistema *Post Shuffling*

O subsistema *Post Shuffling* tem o seu foco principal na análise das seqüências resultantes de um experimento de DNA *shuffling* com o objetivo de encontrar seqüências remontadas (seqüências *shuffled*) que são uma "mistura" de fragmentos originários de parentais distintos. Para experimentos de DNA *shuffling* que não possuem método de *screening* disponível, o seqüenciamento dos clones resultantes é o primeiro passo na busca por recombinantes. A fim de encontrar tais seqüências, dois módulos distintos são disponibilizados; um definido como *Library Housekeeping*, o qual realiza tarefas relacionadas à "limpeza" das seqüências *shuffled*, como por exemplo, a remoção de subseqüências contaminantes (originárias dos vetores de clonagem e dos *primers* utilizados na amplificação, por exemplo), além da verificação da orientação das mesmas. O outro módulo, identificado como *Search for Recombinants*, é responsável por realizar a busca por seqüências recombinantes entre as seqüências resultantes do *shuffling* – faz uma busca por seqüências resultantes de cruzamentos entre os parentais. Além desses dois módulos, o subsistema *Post Shuffling* disponibiliza a visualização do alinhamento entre as seqüências parentais e qualquer uma das seqüências *shuffled*. O alinhamento é construído considerando-se a seqüência de aminoácidos ou nucleotídeos dos parentais e da seqüência *shuffled* escolhida.

As seções 4.2.3.1, 4.2.3.2 e 4.2.3.3 descrevem, respectivamente, os três agrupamentos de funcionalidades implementadas pelo módulo *Post Shuffling* identificadas por: *Library Housekeeping*, *Search for Recombinants* e *Parental and Shuffled Sequence Alignment*.

4.2.3.1 *Library Housekeeping*

Como visto anteriormente, em experimentos de DNA *shuffling*, uma vez obtidas as seqüências *full-length*, elas devem ser clonadas em vetores de expressão com o objetivo de amplificá-las e expressá-las em suas proteínas correspondentes. De uma maneira simples, o processo de clonagem envolve basicamente quatro etapas:

- Escolha do fragmento de DNA a ser clonado;
- Inserção do fragmento selecionado em um vetor de expressão, criando assim uma molécula recombinante;
- Tranformação da célula hospedeira (geralmente bactérias) para que esta receba a molécula recombinante;
- Criação de uma biblioteca de DNA por meio do crescimento das bactérias transformadas dando origem a diversas colônias.

Em experimentos de DNA *shuffling*, o fragmento de DNA a ser clonado (o qual será referenciado aqui como DNA alvo) corresponde às seqüências resultantes do passo 4 do *shuffling*, ou seja, aos fragmentos amplificados cujo tamanho correspondem ao tamanho dos parentais. Os vetores de expressão são moléculas de DNA que, usualmente, possuem a capacidade de auto-replicação, e são utilizadas como um veículo para carregar o DNA alvo para dentro de uma célula hospedeira. Desta forma, ao mesmo tempo em que a célula hospedeira se duplica, o vetor, juntamente como o DNA alvo, também é duplicado. Assim, por meio da auto-replicação do vetor e da replicação da célula hospedeira, obtêm-se inúmeras cópias do DNA alvo resultando em uma biblioteca de DNA. Maiores detalhes sobre o processo de clonagem foram descritos no Capítulo 1, Seção 1.5.

Após o procedimento de clonagem, os fragmentos de DNA alvo podem ser isolados por meio de fragmentação enzimática e, em seguida, seqüenciados para que a seqüência de nucleotídeos que os compõem seja determinada. Apesar do processo de isolamento do fragmento do DNA alvo, alguns contaminantes podem ainda restar na seqüência alvo. Tais contaminantes podem corresponder a seqüências de *primers* utilizados na etapa de amplificação (passo 4 do

processo de DNA *shuffling*) bem como nucleotídeos originários do vetor de clonagem que não foram excluídos pela fragmentação enzimática durante o processo de isolamento do DNA alvo.

Desta forma, após os fragmentos de interesse terem sido isolados e seqüenciados, eles ainda devem passar por uma "limpeza" para que as seqüências contaminantes sejam removidas. Após essa limpeza, a busca por recombinantes, como descrita mais adiante na Seção 4.2.3.2, torna-se um processo mais simples. A ferramenta ISAS disponibiliza ao usuário as seguintes funções relacionadas à "limpeza" da biblioteca resultante:

– Identificação e remoção de fragmentos contaminantes, denominados *tags*, das seqüências *shuffled*;
– Identificação da orientação (5'→ 3' ou 3'→ 5') das seqüências *shuffled* e cálculo do complemento, reverso e complemento reverso, quando necessário.

A Figura 4.14 mostra a tela do subsistema *Post Shuffling* com a palheta relativa ao módulo *Library Housekeeping* ativada. Primeiramente, o usuário deve abrir o arquivo de entrada contendo as seqüências resultantes do *shuffling* (botão *Open Shuffled Sequence File*). Depois de aberto o arquivo, as *tags* a serem buscadas nas seqüências devem ser informadas, uma a uma, e o botão *Find Tag in the Sequences* ativado para dar início à busca. O usuário pode escolher, por meio do botão *Choose Color*, com que cor a *tag*, quando encontrada na seqüência, deve ser destacada. Na Figura 4.14, as *tags* identificadas nas cores azul e vermelho correspondem, respectivamente, às seqüências dos:

primer forward (CCCATATGGCGTTGGCCGGCGGCATC); e
primer reverse (GATGCCAGTGCAAATGCCTAAGAATTCGG)

utilizados durante a amplificação das seqüências remontadas após os ciclos de PCR sem *primers* no processo de DNA *shuffling*.

Considere uma determinada *tag* ou contaminante, encontrada em uma seqüência S. O usuário pode escolher entre remover a subseqüência anterior à *tag* (botão *Clear Sequence Before Tag*) ou a subseqüência posterior à *tag* (botão *Clear Sequence After Tag*) em S. A remoção das subseqüências anterior ou posterior à *tag* podem ou não incluí-la e, para tal, o usuário deve escolher entre as opções *Tag sequence included* ou *Just sequence before/after Tag*, respectivamente. O usuário pode ainda realizar a remoção dos contaminantes um a um nas seqüências onde a *tag* foi encontrada, ou realizar a remoção de uma única vez em todas elas.

Para uma determinada *tag*, o módulo permite também que se busque, dentre as seqüências *shuffled*, pelo seu complemento, reverso ou complemento reverso (botões *Complement*, *Reverse* e *Compl. Rev.*, respectivamente). A busca do complemento reverso do *primers forward* e *reverse* anteriormente apresentados, foi bem sucedida uma vez que esses foram encontrados na seqüência C03, como mostra a Figura 4.15. O complemento reverso do *primer forward* está assinalado em C03 na cor azul e sublinhado e o complemento reverso do *primer reverse* na cor vermelho e sublinhado. Note na figura que as subseqüências contaminantes presentes na seqüência A01 já foram removidas.

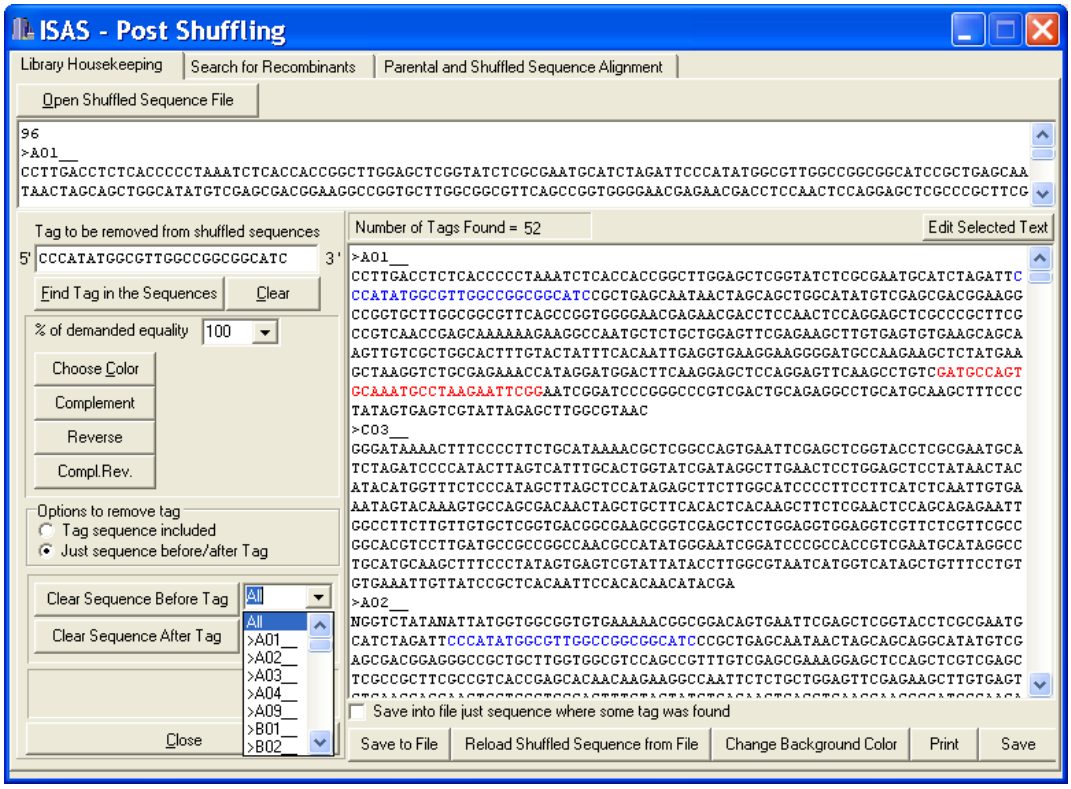


Figura 4.14. *Post Shuffling*. Busca por contaminantes nas seqüências *shuffled* durante o processo de *Library Housekeeping*.

Como na seqüência C03 foram encontrados apenas os complementos reversos das *tags* buscadas, essa seqüência deve ser substituída pelo seu complemento reverso (botão *Calcule Compl. Rev.* Figura 4.15) antes que a busca por recombinantes seja executada.

Além de realizar a busca por subseqüências idênticas à *tag* nas seqüências *shuffled*, o ISAS permite que buscas inexatas sejam executadas. Uma busca é exata quando, dado uma *tag*, busca-se por subseqüências que sejam 100% iguais à *tag*; é inexata quando se busca por subseqüências que sejam pelo menos P% iguais à *tag*. A porcentagem de igualdade exigida na busca tem valor 100% por *default*; o usuário pode informar a porcentagem desejada na lista de opções *% of demanded*

equality. A necessidade de implementar a busca inexata justifica-se por dois motivos: erros ocorridos durante o processo de seqüenciamento e erros ocorridos durante as reações de PCR.

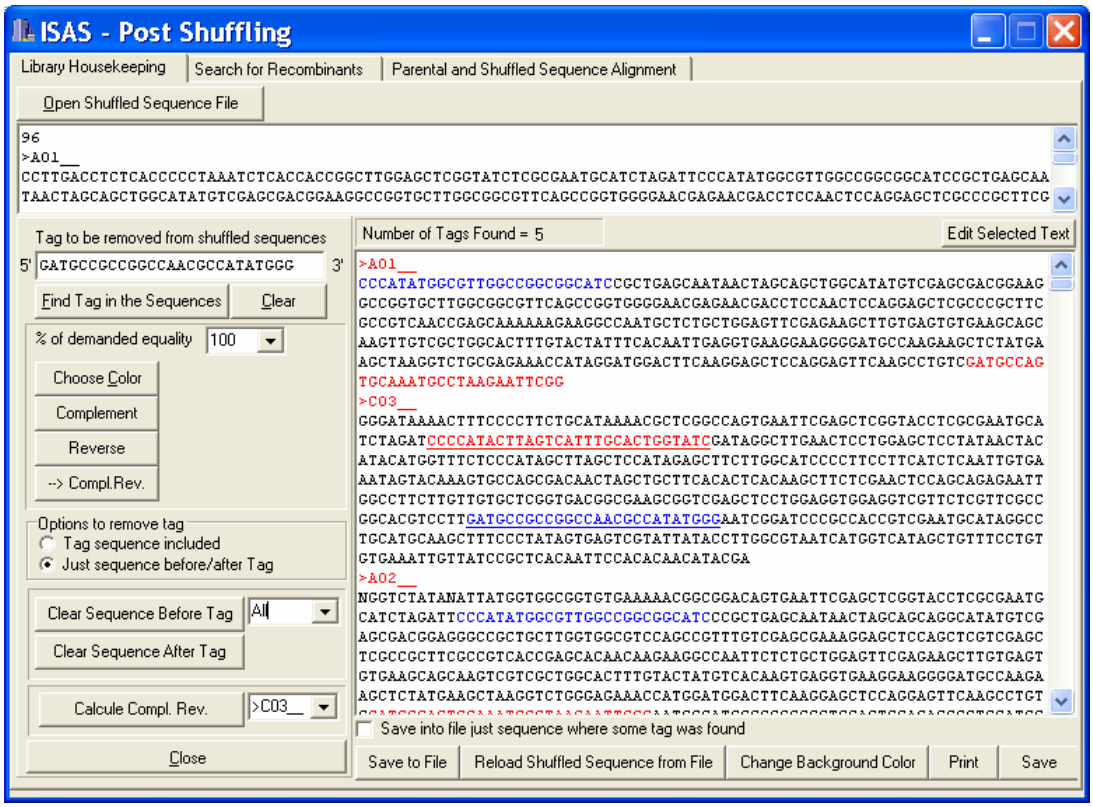


Figura 4.15. *Post Shuffling*. Busca pelo complemento reverso dos *primers* utilizados na amplificação das seqüências *shuffled*.

Quando a máquina seqüenciadora for incapaz de interpretar o sinal lido em um ponto específico do processo de seqüenciamento de um fragmento de DNA, sua incerteza é representada pela inserção do caractere N na seqüência ao invés de um dos caracteres A, T, C ou G. Os caracteres N inseridos nas seqüências podem causar problemas durante a busca por *tags*.

Como visto no Capítulo 1, Seção 1.4, nucleotídeos podem ser removidos, inseridos ou ainda substituídos na nova molécula de DNA sendo sintetizada durante as reações de PCR. Estas remoções, inserções e/ou substituições irão, de maneira semelhante aos erros ocorridos durante o seqüenciamento, impedir que *tag*s em potencial sejam localizadas nas seqüências *shuffled*. Desta forma, a implementação da busca inexata permite que o usuário encontre nas seqüências *shuffled* subseqüências que são no mínimo P% iguais à *tag* e decidir se estas podem ser ou não consideradas como a *tag* buscada.

Como exemplo, considere a busca pelo *primer forward* em um arquivo contendo 94 seqüências *shuffled* resultantes do *shuffling* entre os parentais Oryza e SD. A busca exata (P =

100%) retornou 52 ocorrências desta *tag* nas seqüências enquanto que a busca com P = 96% encontrou 62 ocorrências. A seqüência A05 foi uma das seqüências *shuffled* na qual uma aproximação da *tag* foi encontrada. A Figura 4.16 mostra a seqüência A05 com a aproximação da *tag* sublinhada. Note que essa difere da *tag* (*primer forward*) em apenas uma base (destacada em preto). Nesse caso, o usuário pode decidir entre simplesmente trocar a base G pela base correta A (base do *primer*), realizar novamente o seqüênciamento desta seqüência para confirmar qual a base deve ocupar a posição em questão, ou então, descartar essa seqüência considerando que esta aproximação encontrada não corresponde à *tag* buscada.

```
>A05__
NGGTGTAGTAACGGANNNTTGGGGNNTGATCATCGNCATTTGAATTTGAGCTCGGTGACTCGCGAATGCATCTAGAT
TCCCATATGGCGTTGGCCGGCGGCAGCTGGGATGCCAGTGCAAATGCCTAATAATTCGGATGCCAGTGCAAATGCCT
AAGAATTCGGAATCGGATCCCGGGCCCGTCGACTGCAGAGGCCTGCATGCAAGCTTTCCCTATAGTGAGTCGTATTA
GAGCTTGGCGTAATCATGGTCATAGCTGTTTCCTGTGTGAAATTGTTATCCGCTCACAATTCCACACAACATACGAG
CCGGAAGCATAAAGTGTAAAGCCTGGGGTGCCTAATGAGTGAGCTAACTCACATTAATTGCGTTGCGCTCACTGCCC
GCTTTCCAGTCGGGAAACCTGTCGTGCCAGCTGCATTAATGAATCGGCCAACGCGCGGGGAGAGGCGGTTTGCGTAT
TGGGCGCTCTTCCGCTTCCTCGCTCACTGACTTCGCTGCGCTCGGTCGTTCGGCTGCGGCGAGCGGTATCAGC
```

Figura 4.16. Sequencia *shuffled* A05 na qual uma aproximação do *primer forward* foi encontrada (seqüência sublinhada).

Depois de realizado o *housekeeping* das seqüências *shuffled*, é possível identificar seqüências não úteis (seqüência nas quais nenhuma *tag* foi encontrada), seqüências que devem ser seqüenciadas novamente, bem como seqüências potencialmente úteis, ou seja, seqüências que podem ser o resultado de uma "mistura" dos parentais.

4.2.3.2 *Search for Recombinants*

A busca por seqüências recombinantes na biblioteca resultante do *shuffling* é uma etapa crítica em experimentos deste tipo devido ao grande volume de dados produzidos, principalmente quando não se tem um método de *screening* (ou seleção) disponível. Um dos procedimentos mais comuns utilizados para identificar recombinantes é a construção de alinhamentos entre as seqüências parentais e as seqüências da biblioteca. O alinhamento pode ser construído utilizando-se a seqüência de nucleotídeos ou a seqüência de aminoácidos.

O sistema ISAS implementa funções similares para a análise da biblioteca resultante. Contudo, antes de construir o alinhamento, ISAS realiza a busca pelas mutações identificada entre os parentais durante a etapa de *Parental Analyses* (Seção 4.2.2.1) nas seqüências resultantes do *Library HouseKeeping* (Seção 4.2.3.1). Desta forma, o usuário pode economizar no tempo gasto

com a análise da biblioteca visualizando claramente, para cada uma das seqüências *shuffled*, onde as recombinações ocorreram bem como a quantidade de tais recombinações.

Assim como a busca inexata por *tags*, a busca inexata por mutações também é possível. Devido à possibilidade de realização de buscas inexatas, substituições, inserções e/ou remoções de nucleotídeos ocorridos durante as reações de PCR nas regiões identificadas como mutações entre os parentais, bem como erros ocorridos durante o seqüenciamento podem ser identificados e tratados pelo usuário antes que o alinhamento seja construído.

Para dar início à busca pelas mutações nas seqüências resultantes do *shuffling*, dois arquivos devem ser abertos: o arquivo contendo as seqüências *shuffled* resultantes do *Housekeeping* (botão *Open Shuffled Sequence File*, Figura 4.17) e o arquivo contendo as mutações encontradas entre os parentais (botão *Load Mutations from File*, Figura 4.17) durante a fase de *Parental Analyses*. A busca pelas mutações originárias de cada um dos parentais pode ser executada uma a uma selecionando-se a mutação, a cor na qual tal mutação deve ser destacada quando encontrada e clicando no botão *Search for Mutation* (Figura 4.17). Contudo, antes que a busca seja realizada, um painel de informação é mostrado permitindo que o usuário informe com que percentual (P) de igualdade a mutação deve ser buscada nas seqüências *shuffled*. Depois de realizada a busca, o sistema informa o número de vezes que a mutação buscada foi encontrada.

Caso a busca por uma mutação não produza resultados significativos, ou seja, um número muito grande (maior que o número de seqüências do arquivo, por exemplo) dessa mutação tenha sido encontrado, a busca pode ser desfeita clicando-se no botão *Undo*. Resultados não significativos podem ocorrer devido a um baixo valor de P ou ainda como conseqüência de um pequeno valor associado ao parâmetro n, correspondente ao tamanho mínimo das mutações, utilizado durante a etapa de análise dos parentais (Seção 4.2.2.1).

A Figura 4.17 mostra a tela do subsistema *Post Shuffling* na qual foi realizada a busca por algumas das mutações identificadas entre os parentais Oryza e SD em um conjunto de 94 seqüências resultantes do *shuffling* entre esses dois parentais.

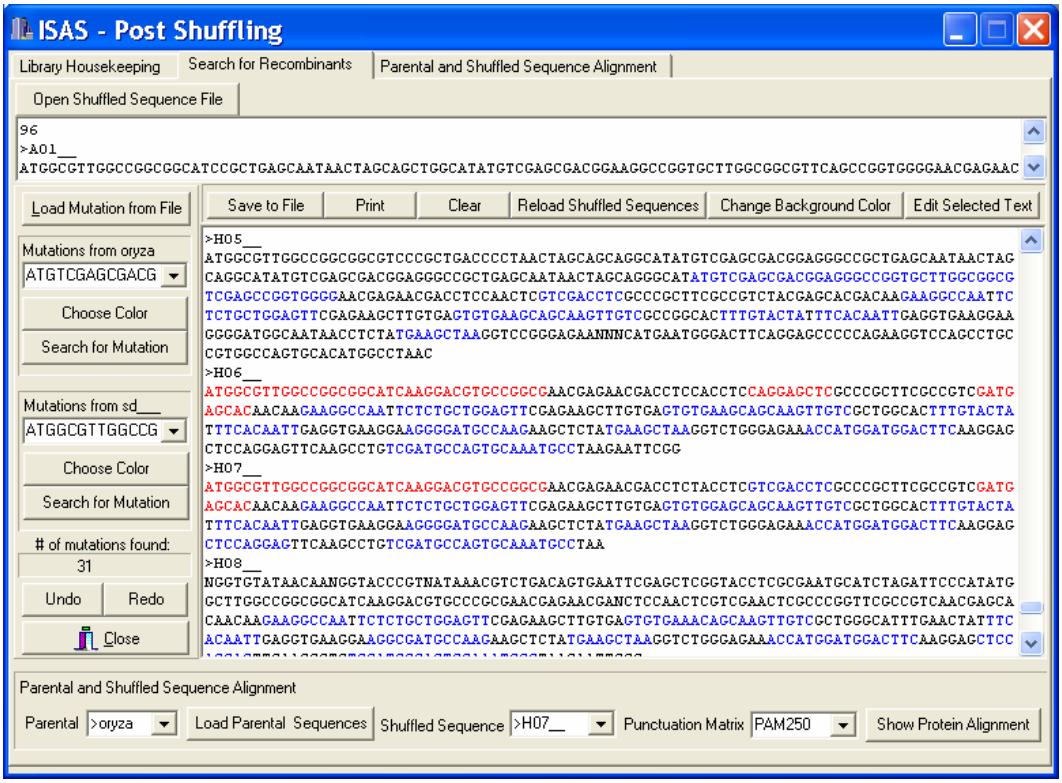


Figura 4.17. *Post Shuffling*. Busca nas seqüências *shuffled* por mutações originárias dos parentais.

A fim de facilitar a identificação visual das regiões de cruzamento, o usuário deve escolher uma única cor para que sejam destacadas todas as mutações originárias de um dos parentais e outra cor para todas as mutações do outro parental, como no exemplo da Figura 4.17, onde todas as mutações originárias do parental Oryza foram coloridas em azul e todas as mutações originárias do parental SD coloridas em vermelho. Observando a seqüência *shuffled* H07, podem ser facilmente identificados três cruzamentos, os quais são evidenciados pela identificação de quatro mutações consecutivas coloridas alternadamente nas cores vermelho e azul. Como indicado pela coloração das mesmas, tais mutações são originárias dos parentais SD e Oryza, e correspondem às mutações M_1, M_2, M_3 e M_4 (ver Tabela 4.1).

4.2.3.3 *Parental and Shuffled Sequence Alignment*

Uma vez que uma seqüência *shuffled* na qual cruzamentos ocorreram foi identificada, ISAS permite ao usuário construir o alinhamento entre os aminoácidos correspondentes a essa seqüência e as seqüências parentais (botão *Show Protein Alignment*). Para cada uma das colunas do alinhamento construído entre as seqüências *shuffled*, Parental 1 e Parental 2, uma dentre quatro situações distintas pode ocorrer:

– os três aminoácidos são iguais, logo não são coloridos;

– o aminoácido da seqüência *shuffled* é igual apenas ao aminoácido do Parental 1. Neste caso, apenas esse aminoácido, tanto na seqüência *shuffled* quanto no Parental 1 é colorido;
– o aminoácido da seqüência *shuffled* é igual apenas ao aminoácido do Parental 2. Neste caso, apenas esse aminoácido, tanto na seqüência *shuffled* quanto no Parental 2 é colorido;
– o aminoácido da seqüência *shuffled* é diferente de ambos os amonoácidos correspondentes no Parental 1 e Parental 2. Neste caso, apenas o aminoácido da seqüência *shuffled* é colorido.

A Figura 4.18 mostra o alinhamento protéico entre a seqüência H07 e os parentais Oryza e SD. Observando a coloração das colunas desse alinhamento, é possível, assim como na Figura 4.17, descobrir como os parentais foram "misturados" para formar a seqüência H07, além de ser possível também identificar a presença de novos aminoácidos nessa seqüência (veja colunas 26 e 60 do alinhamento). Os novos aminoácidos podem ser resultado de erros ocorridos durante a reação de PCR ou, ainda, erros durante o seqüenciamento do clone H07.

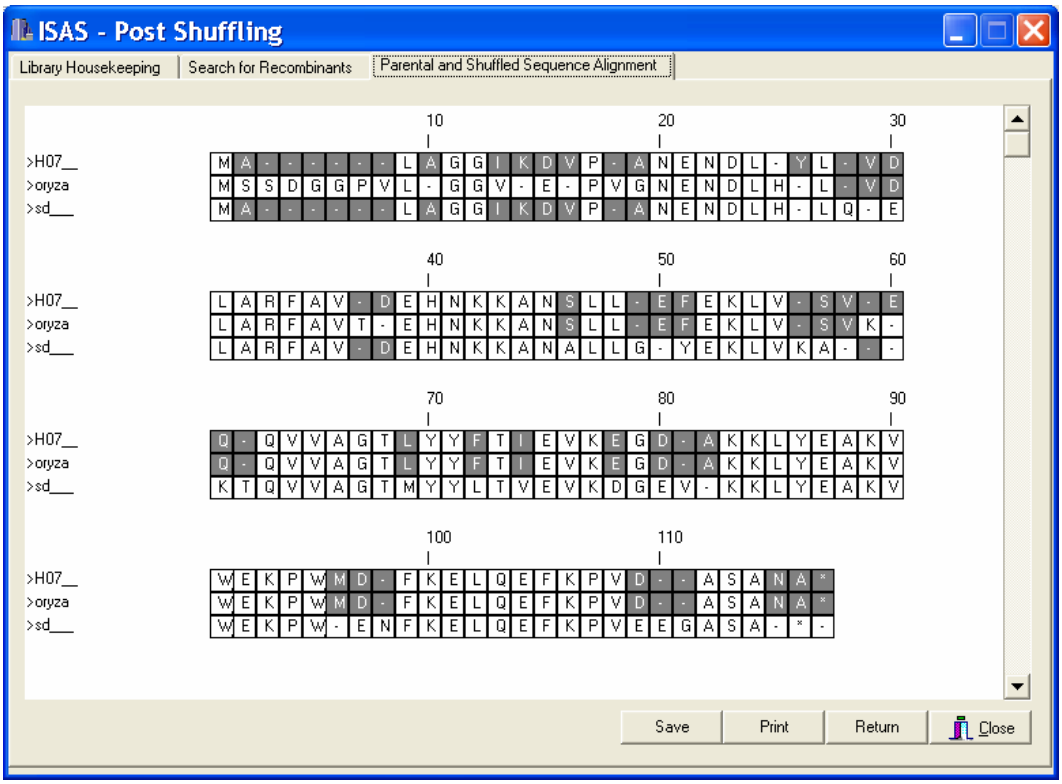


Figura 4.18. *Post Shuffling*. Alinhamento entre a seqüência de aminoácidos correspondentes à seqüência *shuffled* H07 e os parentais Oryza e SD.

4.3 Considerações finais

A ferramenta ISAS (*Interactive Software for Assisting Shuffling Processes*) foi desenvolvida na linguagem C, utilizando o ambiente C++ Builder 5.

A implementação foi feita de forma modular, separando as funcionalidades do ISAS em três diferentes subsistemas os quais agrupam funções relacionadas entre si para:

- análise e manipulação de seqüências (subsistema *Sequence Basics*) como composição, determinação da seqüência complemento, reverso, complemento reverso, estatística dos aminoácidos correspondentes a uma seqüência de DNA, projeto de *primers* necessários à amplificação da seqüência, entre outros;
- análise prévia das seqüências parentais (subsistema *Pre Shuffling*), incluindo construção de alinhamento para que se possam identificar as regiões de similaridade e diferenças (mutações) entre as seqüências e ferramentas para a simulação *in silico* do processo de DNA *shuffling*; e
- análise e manipulação das seqüências resultantes do DNA *shuffling* (subsistema *Post Shuffling*), como remoção de contaminantes e busca por recombinantes.

O módulo de *Pre Shuffling* traz especial contribuição por viabilizar a utilização de dois softwares descritos na literatura para a simulação *in silico* do processo de DNA *shuffling,* eShuffle e DRIVeR, além de implementar o modelo de simulação proposto neste trabalho e denominado SimAffling (descrito no Capítulo 5). Uma interface gráfica, tanto para o fornecimento dos parâmetros necessários à execução dos softwares eShuffle e DRIVeR quanto para a visualização dos resultados foi implementado pelo ISAS a fim de facilitar a utilização dos mesmos, uma vez que suas implmentações originais não dispõem de tal recurso.

Para o software eShuffle, além do valores propostos por Allawi e SantaLucia (1997) para variação de entalpia (ΔS) e variação de entropia (ΔH) dos pares de bases vizinhos que compõem uma determinada região de pareamento implementado originalmente pelo eShuffle, outros três modelos (SUGIMOTO et al., 1996), (SANTALUCIA et al., 1996) e (BRESLAUER, 1986) foram disponibilizados pelo ISAS. Originalmente, o DRIVeR permite apenas que, como parentais, sejam utilizadas seqüências que diferem em no máximo 20 pares de bases. Para possibilitar a utilização do DRIVeR para pares de seqüências que diferem entre si em mais de 20 pares de

bases, foi proposta uma metodologia para agrupar mutações próximas e consecutivas em uma única mutação.

ISAS apresenta-se como uma ferramenta de suporte a usuários que pretendem ou já realizaram experimentos de DNA *shuffling* disponibilizando ferramentas para as diversas etapas envolvidas neste tipo de experimento. A análise das seqüências candidatas a parentais permite que características relevantes, tais como, similaridade, número e localização das mutações entre as seqüências sejam avaliadas no contexto de experimentos de DNA *shuffling*. Por meio da simulação *in silico* do processo de DNA *shuffling*, é possível estimar a influência de parâmetros como, tamanho dos fragmentos e temperatura de pareamento sobre os resultados esperados, e, desta forma, realizar otimizações antes da realização *in vitro* do experimento.

A forma modular como o sistema foi projetado e implementado permite a fácil incrementabilidade do ISAS, de forma que novos subsistemas e funcionalidades podem ser adicionados sem acarretar modificações nas implementações já realizadas.

5 Capítulo

SimAffling: Simulador de DNA *shuffling*

"Todas as flores do futuro estão nas sementes de hoje."
Provérbio Chinês

5.1 Introdução – Motivação para a proposta SimAffling

Após o estudo e investigação dos modelos de Sun, Moore, Patrick e David, bem como dos conceitos e formalismos que subsidiaram tais modelos, foi possível comparar as funcionalidades, contribuições e deficiências de cada um deles. Os quatro modelos se utilizaram de formalismos diferentes para, de alguma forma, avaliar e/ou predizer os resultados de experimentos de DNA *shuffling*. Diferentemente dos modelos eShuffle e ShuffIt, o modelo DRIVeR, proposto por Patrick, e o modelo proposto por Sun são modelos puramente estatísticos, que não utilizam de conceitos de termodinâmica para modelar os eventos de pareamento além de não realizarem a fragmentação dos parentais e tampouco considerarem a composição das seqüências como parâmetros de entrada para os seus modelos. Esses modelos consideram, basicamente, a distância entre as mutações encontradas entre os parentais como fator determinante nos resultados dos DNA *shuffling*.

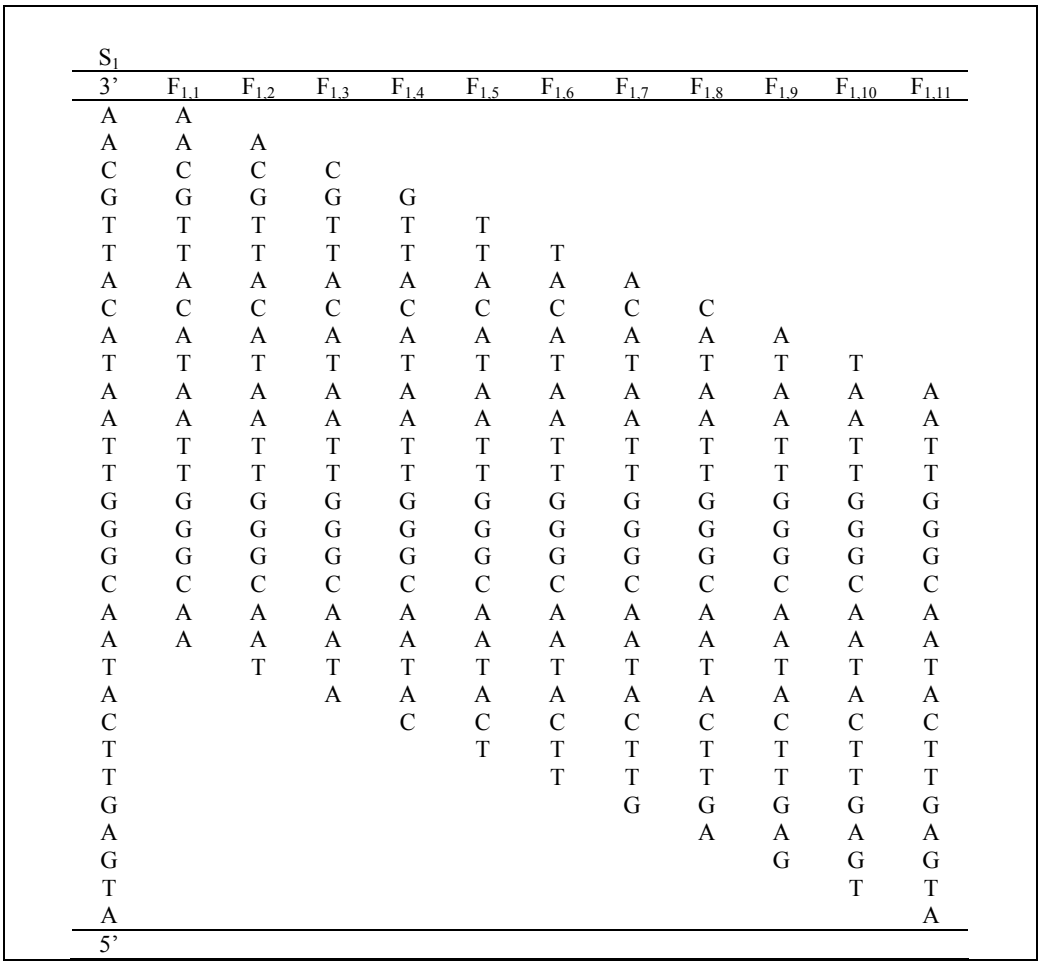
Uma inconveniência para a utilização do DRIVeR é a necessidade de saber, de antemão, o número médio de cruzamentos observado em uma amostra da biblioteca resultante (λ^{true}), bem como o tamanho da biblioteca gerada, uma vez que o modelo utiliza essas informações para estimar o número médio de cruzamentos nas seqüências resultantes do *shuffling*. Para usuários que

ainda não tenham realizado experimentos de DNA *shuffling* ou não tenham feito a análise de uma amostra das seqüências resultantes para a determinação do valor de λ^{obs}, os resultados preditos podem não ser significativos, caso o valor de λ^{obs} utilizado nas simulações não seja um valor passível de ser obtido experimentalmente ou, ainda, se o valor de λ^{obs} tenha sido calculado sobre uma pequena amostra das seqüências resultantes do *shuffling* de forma que os resultados estimados pelo DRIVeR podem não ser significativos.

O número de ciclos de PCR simulados pelo ShuffIt não é um parâmetro fixo do modelo, de forma que o usuário deve testar/validar qual o número de ciclos mais adequado para a simulação que deseja realizar. A determinação do valor desse parâmetro, contudo, não é uma tarefa simples e requer um grande dispêndio de tempo, além do que, tentativas de execução do ShuffIt utilizando valores inadequados podem resultar em erros de execução, como nos testes realizados e descritos no Apêndice B. Outro fator agravante na utilização do ShuffIt é que o uso de valores não apropriados para o número de ciclos de PCR (caso não produzam erros de execução) provavelmente resultarão em estimativas não representativas do experimento de DNA *shuffling* sendo simulado. O autor não apresenta um estudo que direciona a escolha do parâmetro correspondente ao número de ciclos de PCR, de forma que não se pode dizer o que seria um valor "apropriado" para este parâmetro. Em contato com o autor (comunicação pessoal)[32], Maheshri confirmou que o número de ciclos de PCR deve ser determinado empiricamente (por tentativa e erro) com base no tamanho dos parentais e no tamanho dos fragmentos utilizados na simulação.

Apesar de o modelo eShuffle se apresentar como o mais consistente dos quatro, se considerarmos a abordagem dada ao evento de pareamento por utilizar-se dos conceitos de Termodinâmica que o regem e por considerar que a seletividade de um fragmento depende a sua concentração na reação, a fragmentação dos parentais implementada pelo eShuffle, possivelmente, não é representativa do processo. Enquanto que em experimentos de DNA *shuffling*, após a fragmentação dos parentais, os fragmentos resultantes são purificados em gel de agarose para que aqueles cujos tamanhos estejam compreendidos em um intervalo de interesse sejam selecionados, o modelo eShuffle considera apenas fragmentos de tamanho fixo L. A Figura 5.1 mostra um esquema de como a fragmentação de um parental S_1 de tamanho $t_1 = 30$ é abordada no eShuffle considerando L = 20.

[32] MAHESHRI, N. Questions about ShuffIt. Mensagem recebida por lumontera@gmail.com em 25 de Agosto de 2008.



S_1											
3’	$F_{1,1}$	$F_{1,2}$	$F_{1,3}$	$F_{1,4}$	$F_{1,5}$	$F_{1,6}$	$F_{1,7}$	$F_{1,8}$	$F_{1,9}$	$F_{1,10}$	$F_{1,11}$
A	A										
A	A	A									
C	C	C	C								
G	G	G	G	G							
T	T	T	T	T	T						
T	T	T	T	T	T	T					
A	A	A	A	A	A	A	A				
C	C	C	C	C	C	C	C	C			
A	A	A	A	A	A	A	A	A	A		
T	T	T	T	T	T	T	T	T	T	T	
A	A	A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A	A	A
T	T	T	T	T	T	T	T	T	T	T	T
T	T	T	T	T	T	T	T	T	T	T	T
G	G	G	G	G	G	G	G	G	G	G	G
G	G	G	G	G	G	G	G	G	G	G	G
G	G	G	G	G	G	G	G	G	G	G	G
C	C	C	C	C	C	C	C	C	C	C	C
A	A	A	A	A	A	A	A	A	A	A	A
A	A	A	A	A	A	A	A	A	A	A	A
T		T	T	T	T	T	T	T	T	T	T
A			A	A	A	A	A	A	A	A	A
C				C	C	C	C	C	C	C	C
T					T	T	T	T	T	T	T
T						T	T	T	T	T	T
G							G	G	G	G	G
A								A	A	A	A
G									G	G	G
T										T	T
A											A
5’											

Figura 5.1. Esquema representativo da fragmentação dos parentais pelo eShuffle.

A possibilidade de pareamento para cada fragmento $F_{1,i}$ é verificada apenas para os fragmentos $F_{j,i}$, para $1 \leq j \leq N$, onde N é o número de parentais envolvidos na simulação. Com essa restrição, apenas os fragmentos que estão na mesma posição do alinhamento são considerados como candidatos ao pareamento. A suposição que apenas este tipo de pareamento pode ocorrer é uma simplificação drástica para esse tipo de evento, uma vez que em experimentos *in vitro*, quaisquer fragmentos, independente de sua localização, podem se parear e produzir um fragmento recombinante desde que compartilhem complementaridade entre as bases que os compõem. Adicionalmente, o eShuffle considera a ocorrência de pareamentos apenas entre fragmentos de fitas opostas (veja o pseudo-código do algoritmo no Apêndice C, Seção C.2) enquanto que pareamentos entre fragmentos de mesma fita, bem como fragmentos de um mesmo parental, também podem ocorrer. Informações sobre o número, ou a porcentagem de seqüências *full-length* após a simulação também não é determinada pelo eShuffle. Essa informação é essencial, pois as seqüências *full-length* são aquelas que serão posteriormente amplificadas na etapa de PCR com *primers*. Os cruzamentos ocorridos entre fragmentos cujos tamanhos, após os

ciclos de remontagem, não se aproximam do tamanho dos parentais não devem influenciar no número médio de cruzamentos estimado, uma vez que essa estimativa só é relevante para fragmentos do tipo *full-length*.

Para a utilização do ShuffIt, além de conhecimentos básicos da linguagem de programação C para que os parâmetros de entrada sejam modificados dentro do código do programa, é preciso que o usuário tenha também um compilador para a linguagem C. Para a execução do eShuffle, as seqüências parentais devem estar armazenadas em um arquivo de entrada seguindo um formato particular como mostrado no Apêndice C, Seção C.1. O DRIVeR limita o número de mutações entre os parentais em 20. Para que o *shuffling* entre seqüências parentais com mais do que 20 mutações entre si seja simulado pelo DRIVeR, esse trabalho sugere que o usuário realize primeiramente a identificação e o agrupamento das mutações como implementado pelo ISAS (ver Capítulo 4, Seção 4.2.2.1).

Deve-se considerar ainda que todas as três implementações não apresentam interface gráfica, nem mesmo um manual de usuário que esclareça os detalhes de utilização dos softwares, como apresentado nesta tese, fatos esses que podem ser os responsáveis por não terem sido encontrados na literatura outros trabalhos descrevendo experimentos de DNA *shuffling* nos quais qualquer desses modelos tenham sido utilizados.

Frente às dificuldades e limitações encontradas nos modelos estudados e implementados, e para um melhor entendimento de cada uma das etapas do processo de DNA *shuffling*, este trabalho de pesquisa propõe e implementa um simulador de DNA *shuffling* chamado SimAffling, descrito nas próximas seções.

5.2 Simulador de DNA *shuffling*

A simulação é uma forma numérica de realizar experimentos em computador com base em modelos lógicos ou modelos matemáticos, de modo a descrever o comportamento de sistemas ao longo de um determinado período de tempo (RUBINSTEIN, 1981). O recurso de simulação tem sido utilizado na simulação de processos de diversas áreas como manufatura, aplicações militares, logística e transporte. Em alguns casos, simulações podem ser realizadas por modelos matemáticos que descrevem o comportamento do sistema. Entretanto, em muitos casos, o processo a ser simulado é tão complexo que é impossível resolve-lo matematicamente. Nesses

casos, simulações numéricas podem ser utilizadas para imitar o comportamento do processo ou sistema ao longo do tempo (BANKS et al., 2001).

Segundo descrito em Banks et al. (2001), de forma geral, um modelo é construído com base em uma série de suposições a respeito do sistema que se deseja modelar. Essas suposições são expressas por meio de relações matemáticas, lógicas e simbólicas entre os objetos que compõem o sistema. Uma vez desenvolvido e validado, o modelo poderá ser utilizado para investigar uma série de questões referentes ao sistema. Mudanças no sistema podem, por exemplo, ser primeiramente simuladas a fim de predizer o impacto de tais mudanças no seu comportamento. A simulação pode também ser utilizada como uma ferramenta para análise de sistemas que ainda não existem fisicamente.

O experimento de DNA *shuffling* pode ser descrito como um sistema complexo onde diversas reações ocorrem ao longo do tempo. Com base nos estudos realizados, um modelo lógico foi proposto para que um simulador de DNA *shuffling* fosse implementado. A implementação do modelo proposto recebeu o nome de SimAffling e é descrita a seguir.

De forma geral, quatro módulos distintos podem ser identificados no software SimAffling: Geração de números aleatórios segundo uma distribuição de probabilidade; Fragmentação dos parentais; Pareamento; e Extensão dos fragmentos pareados. O módulo de geração de números aleatórios permitirá a fragmentação das seqüências parentais pelo módulo de Fragmentação. Os módulos de Pareamento e Extensão correspondem aos ciclos de PCR e, por essa razão, são repetidos um determinado número de vezes a fim de promover a remontagem dos fragmentos. Por se tratar de uma simulação, o processo deve ser repetido um número suficiente de vezes até sua convergência. A Figura 5.2 descreve o pseudo-código do simulador proposto e implementado.

```
1.  procedure SimAffling(P1, P2, L, Min, Max, n_simulações, tipo_frag, ciclos_PCR,
                         mínimo, Fator)
2.    {Entrada: P1 → seqüência de DNA do parental 1
3.              P2 → seqüência de DNA do parental 2
4.              L → tamanho médio em torno do qual os fragmentos devem ser gerados
5.              Min → tamanho mínimo permitido para um fragmento
6.              Max → tamanho máximo permitido para um fragmento
7.              n_simulações → número de simulações a serem executadas
8.              tipo_frag → indica o tipo de fragmentação a ser realizada, se segundo Poisson (tipo_frag = 0)
```

Figura 5.2. Pseudo-código do SimAffling (continua).

```
9.                          ou se a fragmentação é aleatória (tipo_frag = 1)
10.              ciclos_PCR → número de ciclos de PCR a serem executados
11.              mínimo → tamanho de sobreposição mínima exigida para que um pareamento ocorra
12.              Fator → valor inteiro entre 0 e 10 utilizado para o cálculo do valor Limite
13.   }
14.
15.  {Saída: Número médio de cruzamentos nas seqüências remontadas que, possivelmente,
16.          representam seqüências full-length bem como a porcentagem dessas seqüências que são o
17.          resultado da ocorrência de 0, 1, 2, 3, ..., 5 cruzamentos entre os parentais P1 e P2}
18.  begin
19.    for k = 0, 1, 2, ..., n_simulações do
20.      begin
21.          {gera os números que servirão de base para a fragmentação dos parentais}
22.          if(tipo_frag == 1) then
23.              N = gera_números_poisson(L, Min, Max)
24.          else
25.              N = gera_números_aleatórios(Min, Max)
26.
27.          {determina o número de cópias das seqüências de forma a garantir
28.            um conjunto de fragmentos com tamanho em torno de 1.200}
29.          C = numero_copias(|P1|, |P2|, Min)
30.
31.          {cria C cópias de cada um dos parentais bem como de seus complementos}
32.          S = cria_copia_parental(P1, P2, C)
33.          {realiza a fragmentação das seqüências do conjunto S}
34.          Single_DNA = fragmenta(S, N)
35.          Double_DNA = Ø
36.          Limite = |Single_DNA| / Fator
37.          for i = 0, 1, 2, ..., ciclos_PCR do
38.            begin
39.                n_não_colisões = 0
40.                do
41.                    {seleciona aleatoriamento dois fragmentos do conjunto Single_DNA}
42.                    f1 = seleciona_aleatoriamente(Single_DNA)
43.                    f2 = seleciona_aleatoriamente(Single_DNA)
44.
45.                    {verifica qual o maior pareamento possível entre f1 e f2}
```

Figura 5.2. Pseudo-código do SimAffling (continuação).

```
46.                     tamanho = verifica_pareamento (f1 , f2)
47.
48.                   if (tamanho ≥ mínimo) then
49.                    begin
50.                       {como o pareamento ocorreu, realize a extensão (quanto possível) e armazene o
51.                        número de cruzamentos ocorridos até o momento nestes fragmentos}
52.                       f1' = estende(f1)
53.                       f2' = estende(f2)
54.
55.                       {insere os fragmentos estendidos (ou não) no conjunto Double_DNA}
56.                       Double_DNA = Double_DNA ∪ f1' ∪ f2'
57.
58.                       {remove do conjunto Single_DNA os fragmentos
59.                        inseridos no conjunto Double_DNA}
60.                       Remove_Single_DNA(f1 , f2)
61.
62.                       n_não_colisões = 0
63.                    end
64.                   else
65.                       n_não_colisões = n_não_colisões + 1
66.                 while{n_não_colisões < Limite or Single_DNA ≠ Ø}
67.
68.                 {transfere todos os fragmentos do conjunto Double_DNA para o
69.                  conjunto Single_DNA para que um novo ciclo de PCR se inicie}
70.                 Single_DNA = Single_DNA ∪ Double_DNA
71.
72.                 Double_DNA = Ø
73.             end
74.       end
75.  end
```

Figura 5.2. Pseudo-código do SimAffling (conclusão).

A seguir, cada uma das instruções e procedimentos do algoritmo descrito na Figura 5.2 são detalhados de maneira a facilitar sua implementação e deixar claro como deve ser sua execução. Para facilitar a descrição, todas as linhas do pseudo-código apresentados foram numeradas. As

considerações e explicações a seguir serão feitas para cada uma das linhas do pseudo-código que se fizerem necessárias.

Linha 1

Descreve o procedimento SimAffling e todas entradas necessárias à sua execução. As linhas de 2 a 12 descrevem, uma a uma, as entradas necessárias ao processo de simulação.

Linha 19

Devido ao fato do SimAffling implementar um modelo estocástico (não determinístico), a simulação do processo de DNA *shuffling* deve ser executada um número suficiente de vezes a fim de garantir a convergência do processo, ou seja, diminuir a variância dos resultados. Se esse passo não existisse, as soluções encontradas poderiam não ser representativas por se tratar de uma abordagem aleatória. Por ser uma simulação baseada em eventos aleatórios, como é o caso da colisão entre fragmentos de DNA possíveis de se parear durante os ciclos de remontagem por PCR, como descrito nas Linhas 42 e 43, o método de simulação implementado é denominado método de simulação Monte Carlo. [ROBERT; CASELLA, 1999].

Testes para validar a convergência do SimAffling foram realizados e os resultados estão descritos na Seção 5.3.

Linha 23 e Linha 25

Como visto no Capítulo 2, a fragmentação das seqüências parentais é feita pela exposição do material genético (DNA) a enzimas de restrição que realizam cortes ao longo da molécula. Mais especificamente, os cortes são realizados entre dois nucleotídeos consecutivos de uma mesma cadeia (ou fita) de DNA (ligações peptídicas). Considerando experimentos desse tipo, para cada ligação peptídica entre dois nucleotídeos, apenas um dentre dois eventos ocorre: a quebra ou a não–quebra pela enzima de restrição da ligação que une os dois nucleotídeos. O SimAffling disponibiliza duas abordagens para modelar o processo de fragmentação dos parentais: fragmentação segundo Poisson em torno de um tamanho médio de fragmentos (Linha 23) e fragmentação aleatória (Linha 25), considerando fragmentos cujo tamanho esteja compreendido num intervalo de tamanho máximo e mínimo. Para que a fragmentação seja possível, valores numéricos inteiros representando o tamanho dos fragmentos são gerados

segundo Poisson ou aleatoriamente. Os conceitos estatísticos necessários à implementação da geração de tais números utilizados durante o processo de fragmentação das seqüências parentais estão descritos no Apêndice F.

Linha 29

Para que as seqüências parentais sejam submetidas a experimentos de DNA *shuffling*, quantidades significativas (ambas as fitas) precisam estar disponíveis. Para tanto, as seqüências são amplificadas por reações de PCR com *primers* (ver Capítulo 1.4). Para fins de simulação, dadas duas seqüências parentais a serem submetidas a um experimento de DNA *shuffling*, C cópias são criadas de cada uma delas, bem como C cópias de cada uma das seqüências complementares.

O valor de C influencia diretamente no número de operações executadas durante os ciclos de PCR uma vez que o número de fragmentos a serem remontados depende deste valor e, consequentemente, influencia no tempo gasto na execução da simulação. Neste trabalho, optou-se pela escolha automática do valor de C. Dados o tamanho das seqüências parentais e o tamanho médio dos fragmentos (L), o número de cópias das seqüências parentais (C) é escolhido de forma a produzir, após a fragmentação, um conjunto com cerca de 1.200 fragmentos. Testes para valores menores e maiores que 1.200 foram realizados e o número médio de cruzamentos estimados, bem como o tempo gasto nas simulações comparados. Os testes realizados estão descritos na Seção 5.3.

Caso a simulação a ser realizada considere dois parentais cujo menor deles tem tamanho m e a fragmentação produza fragmentos cujo tamanho esteja em torno de L pares de bases, o número necessário de cópias (C) de cada seqüência parental (bem como de seu complemento) para garantir um conjunto de fragmentos de tamanho no mínimo 1.200 deve ser:

$$C = \frac{1.200L}{4m}$$

Linha 32

Uma vez determinado o valor de C, o conjunto S contendo C cópias de cada parental, bem como de seus complementos, é criado. Este procedimento corresponde ao processo de amplificação das seqüências de DNA a serem submetidas ao *shuffling*.

Linha 34

Dado o conjunto de números aleatórios gerados pelos procedimentos descritos na Linha 23 ou Linha 25, a fragmentação de todas as S cópias das seqüências de DNA criadas pelo procedimento descrito na Linha 32 é realizada, e um conjunto de fragmentos de fita simples, chamado Single_DNA, é produzido. Considere a seqüência de 10 números aleatórios a_1, a_2, a_3, ..., a_{10}, e duas cópias de uma seqüência parental S_1 de tamanho m pares de bases. Os valores de a_1, a_2, a_3, ..., a_{10} ditam o tamanho dos fragmentos de S_1 a serem produzidos como resultado da fragmentação, como mostra o esquema apresentado na Figura 5.3.

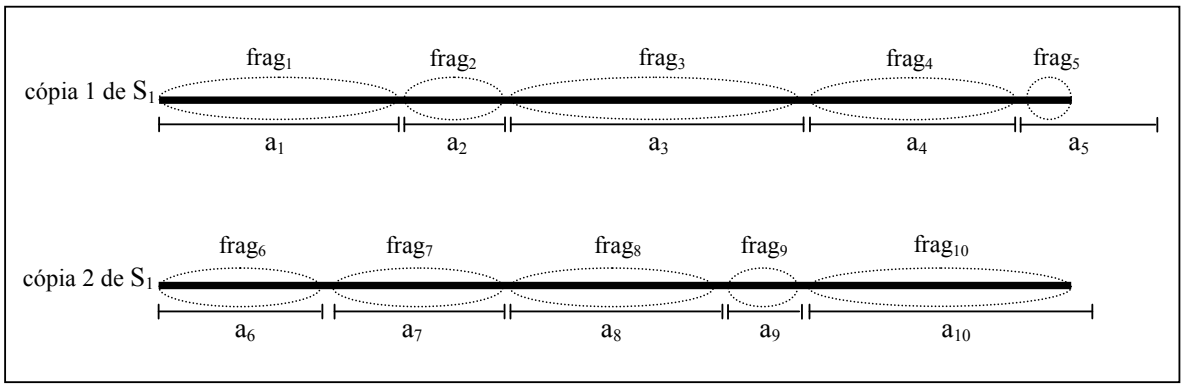


Figura 5.3. Esquema de fragmentação das seqüências parentais.

Note na figura que a fragmentação da extremidade direita da seqüência pode resultar em um fragmento menor do que o valor a_i definido para este fragmento. Feita a fragmentação das N cópias das seqüências parentais bem como de suas seqüências complementares, a purificação destes fragmentos é realizada para que apenas os fragmentos cujo tamanho esteja compreendido no intervalo (Min, Max) definido (pelo usuário) sejam inseridos no conjunto Single_DNA. Note que, após a purificação, o conjunto Single_DNA poderá conter um número ligeiramente menor do que 1.200 fragmentos devido a remoção de fragmentos de tamanho menor que Min, bem como a remoção de fragmentos de tamanho maior do que Max.

Linha 37

Nesse passo são iniciadas as simulações dos ciclos de PCR nos quais a remontagem dos fragmentos acontece. O protocolo original de DNA *shuffling* proposto por Stemmer sugere que sejam executados de 25 a 30 ciclos de PCR. O SimAffling utiliza um valor *default* de 30 ciclos; este valor, entretanto, pode ser modificado pelo usuário antes da simulação. Testes realizados para

valores superiores a 30 ciclos mostraram um elevado número de seqüências com tamanho superior a duas vezes o tamanho das seqüências parentais. Seqüências muito longas dificilmente conterão seqüências *full-length* as quais poderão, ao final do processo de *shuffling*, ser clonadas e expressas, e por este motivo, são removidas do conjunto sempre que encontradas.

Cada um dos ciclos de PCR compreende uma seqüência de eventos que devem acontecer até que uma das duas condições seja atingida: (1) o conjunto de fragmentos de DNA de fita simples (Single_DNA) fique vazio, o que indica que não existem mais fragmentos para que novos eventos de pareamento ocorram; ou (2) o número Limite de tentativas de pareamento sem sucesso tenha sido atingido. As condições de parada (1) e (2) estão descritas na Linha 66. O valor Limite para o número de tentativas de pareamento é dado por:

$$\text{Limite} = \frac{|\text{Single\_DNA}|}{\text{Fator}}$$

onde:

- |Single_DNA| = número de fragmentos contidos no conjunto de fragmentos de fita simples;
- Fator = valor inteiro de 1 a 10. Este valor irá determinar a fração de fragmentos do conjunto Single_DNA que deve ser testada antes que as tentativas de pareamentos sejam interrompidas.

O valor *default* para a variável Fator é 2, o que indica que, se após |Single_DNA|/2 tentativas consecutivas, nenhum pareamento ocorrer, um novo ciclo de PCR deve ser iniciado.

As linhas de 40 a 66 descrevem os procedimentos envolvidos nos ciclos de PCR.

Linha 39 e Linha 40

A cada ciclo de PCR, os fragmentos do conjunto Single_DNA devem colidir[33] aleatoriamente para que a possibilidade de pareamento entre eles seja verificada. A variável n_não_colisões conta o número consecutivo de tentativas de pareamentos sem sucesso durante cada ciclo de PCR e, por isso, é inicializada com zero (Linha 39). A Linha 40 corresponde ao início das colisões entre os fragmentos.

[33] Neste contexto, por colisão entende-se o "encontro" de dois fragmentos.

Linha 42 e Linha 43

Dado o conjunto de fragmentos de fitas simples (Single_DNA), dois fragmentos distintos são aleatoriamente selecionados deste conjunto como fragmentos candidatos ao pareamento. Para tal, dois números aleatórios e distintos, compreendidos entre 1 e |S|, são gerados (ver Apêndice F, Seção F.2). Desta forma, dados dois números aleatórios i e j, para i ≠ j, serão escolhidos como fragmentos candidatos ao pareamento o i-ésimo e o j-ésimo fragmento armazenado no conjunto Single_DNA. Note que o conjunto Single_DNA deve ser implementado como uma lista na qual seus elementos estão ordenados arbitrariamente.

Linha 46

Sejam dois fragmentos f_1 e f_2 de tamanhos arbitrários t_1 e t_2, respectivamente, selecionados aleatoriamente como candidatos ao pareamento. Todas as possíveis configurações de pareamento entre esses dois fragmentos (um total de $(t_1 + t_2 - 1)$ possibilidades) serão consideradas e apenas a melhor delas, caso exista, será considerada como o pareamento que, possivelmente, acontecerá. A Figura 5.4 mostra esquematicamente todas as possíveis configurações de pareamento entre dois fragmentos f_1 e f_2 cujos tamanhos são $t_1 = 10$ e $t_2 = 5$, respectivamente.

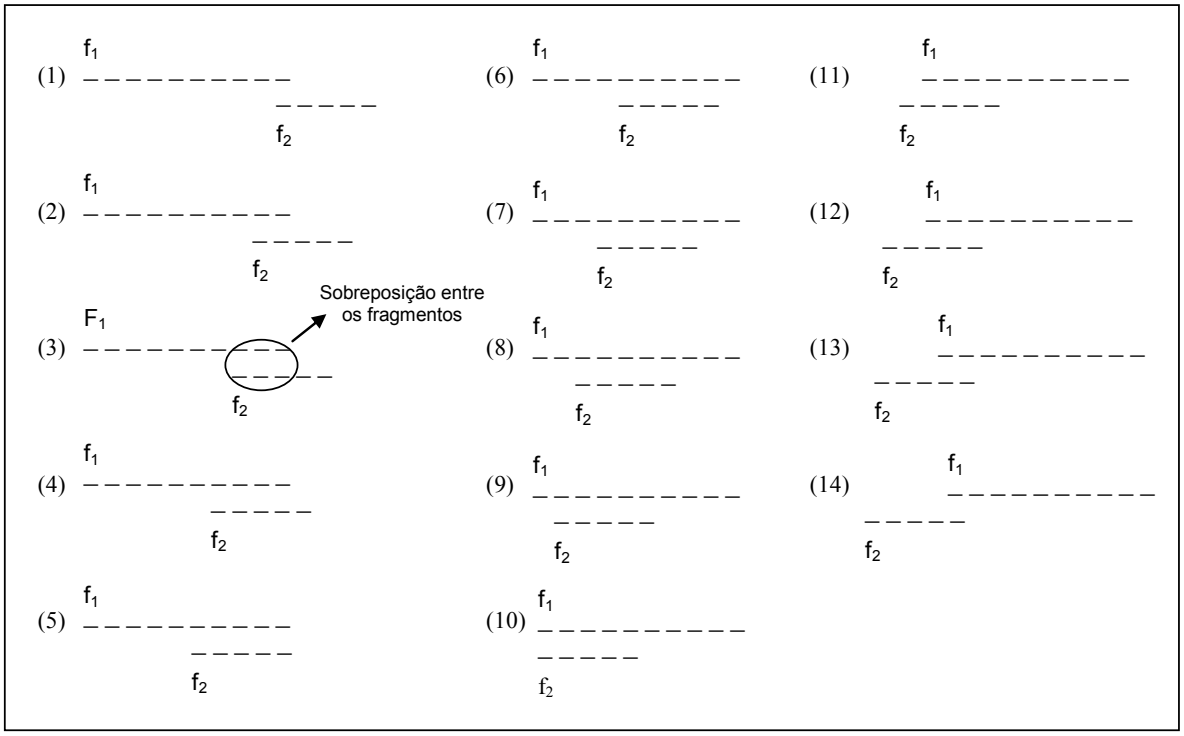


Figura 5.4. Todas as possíveis sobreposições entre os fragmentos f1 e f 2.

Dadas todas as configurações de pareamento ou sobreposição, apenas aquelas que não apresentam *mismatches* são relevantes. Dentre essas últimas, apenas a de maior tamanho é considerada como um pareamento possível de ocorrer entre os fragmentos f_1 e f_2. Contudo, pareamentos envolvendo uma região de sobreposição muito pequena entre os fragmentos, mesmo que essa seja composta apenas por *matches*, são menos estáveis além de menos prováveis de acontecer, exceto se temperaturas de pareamento muito baixas forem utilizadas durante os ciclos de PCR.

Como o SimAffling não contempla a temperatura como um fator determinante nos eventos de pareamentos, optou-se por considerar apenas o pareamento de fragmentos com uma região de sobreposição maior ou igual a um valor *default* de s = 6. Contudo, o valor de s pode ser modificado pelo usuário de forma a melhor representar as condições reais de um experimento de DNA *shuffling*, uma vez que é sabido que temperaturas de pareamento mais baixas (durante os ciclos de PCR) favorecem o pareamento de regiões de sobreposição menores enquanto que temperaturas mais altas favorecem o pareamento de regiões de sobreposição maiores. Sendo assim, quanto menor a temperatura de pareamento utilizada no experimento, menor deve ser o valor de s, e vice-versa.

Linha 48

Enquanto que os modelos de Moore e Maheshri−Schaffer calculam o custo associado a um evento de pareamento com base na variação de energia livre (ΔG) resultante da região de sobreposição, o modelo SimAffling define que o custo associado a um pareamento corresponde ao tamanho da sua região de sobreposição. Regiões de sobreposição onde *mismatches* ocorrem não são consideradas. Esta abordagem mais simples foi adotada no modelo uma vez que se sabe que pareamentos com maior região de sobreposição são mais prováveis de acontecer (por serem mais estáveis) quando comparadas ao pareamento com menores regiões de sobreposição.

Dado um par de fragmentos de DNA para os quais a maior sobreposição entre eles tem tamanho s', $s' \geq s$, onde s é a sobreposição mínima exigida, o modelo considera que um pareamento entre esses fragmentos ocorreu.

Linhas 52, 53, 56, 60 e 62

Dada uma determinada configuração de pareamento entre dois fragmentos, cujo tamanho da sobreposição é no mínimo s, esses devem ser estendidos se possível (Linha 52 e Linha 53) e

inseridos no conjunto de moléculas de fita dupla Double_DNA (Linha 56) até que um novo ciclo de PCR se inicie.

Para decidir se a extensão dos fragmentos pareados é possível ou não, a orientação dos fragmentos deve ser avaliada, uma vez que apenas fragmentos com orientação 5'→ 3' podem ser estendidos. Dados dois fragmentos pareados f_1 e f_2 as seguintes combinações podem acontecer:

1) f_1 e f_2 possuem orientação 5'→ 3';
2) f_1 e f_2 possuem orientação 3'→ 5';
3) f_1 possui orientação 5'→ 3' e f_2 possui orientação 3'→ 5';
4) f_1 possui orientação 3'→ 5' e f_2 possui orientação 5'→ 3'.

Para cada um dos casos anteriores, a localização da região de sobreposição entre os fragmentos irá determinar se a extensão acontecerá ou não, bem como se ambos ou apenas um dos fragmentos será estendido. A Figura 5.5 (a), (b), (c) e (d), mostra todos os possíveis pareamentos que os fragmentos f_1 e f_2 podem assumir em relação um ao outro, em cada um dos casos 1), 2), 3) e 4) enumerados anteriormente. Na figura, considerou-se uma sobreposição de tamanho 4 e que os fragmentos f_1 e f_2 possuem tamanhos distintos $t_1 = 13$ e $t_2 = 8$, respectivamente, sem perda de generalidade para os casos em que $t_1 \leq t_2$ e uma sobreposição de tamanho qualquer. É importante lembrar que, para a extensão ocorrer, a enzima responsável necessita de um fragmento f de orientação 5'→ 3' para dar início à cópia e de um fragmento de DNA molde p, o qual servirá de base para a incorporação de nucleotídeos ao fragmento f, como mostrado no Capítulo 1 Seção 1.4.

Uma vez ocorrido o pareamento e, mesmo que nenhum dos fragmentos tenha sido estendido, ambos os fragmentos são inseridos no conjunto Double_DNA (Linha 56) e aí permanecem até o início do próximo ciclo de PCR. Os fragmentos correspondentes do conjunto Single_DNA, que foram inseridos no conjunto Double_DNA, são então removidos do conjunto de fragmentos possíveis de se parear (Single_DNA) neste ciclo de PCR (Linha 60). Como um pareamento ocorreu, a variável que acumula o número de tentativas consecutivas de pareamento sem sucesso (n_não_colisões) recebe o valor zero (Linha 62).

Nesse ponto, é possível determinar se o(s) fragmento(s) estendido(s) é resultado ou não de um cruzamento entre os parentais. Um fragmento estendido é resultado de um cruzamento se os fragmentos pareados são originários de parentais distintos. O número de cruzamentos ocorrido

ao longo da remontagem dos fragmentos (ciclos de PCR) pode, desta forma, ser determinado para cada fragmento ao longo das simulações.

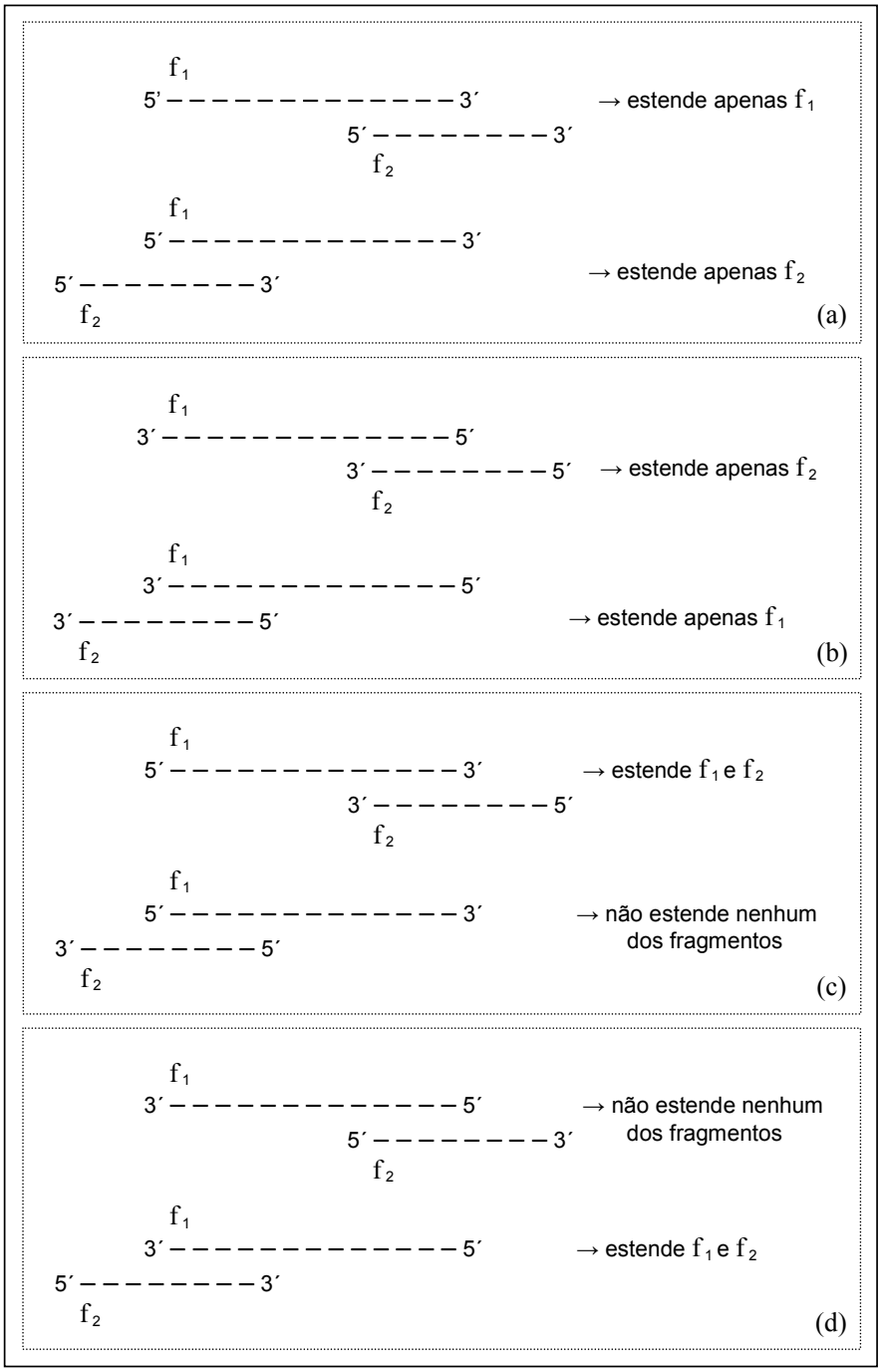


Figura 5.5. Possíveis casos de extensão de fragmentos pareados.

Seja um fragmento f_i de orientação 5'→ 3' que se pareou com o fragmento f_j de orientação 3'→ 5', sendo f_i um fragmento originário do parental P_i e f_j um fragmento originário do parental P_j, e $i \neq j$. Nesse caso, ambos os fragmentos serão estendidos e os fragmentos resultantes são

ditos ser o resultado de um cruzamento entre os parentais P_i e P_j. Após a extensão, o fragmento f_i terá sua extremidade 5' originária do parental P_i e sua extremidade 3' originária do parental P_j . O contrário acontece com o fragmento f_j como pode ser observado na Figura 5.6.

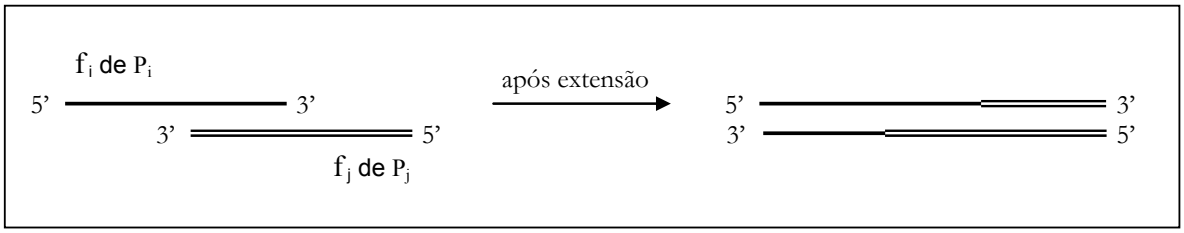


Figura 5.6. Esquema de um evento de cruzamento entre dois fragmentos remontados.

Num próximo momento, após o fragmento de fita dupla mostrado na Figura 5.6 ter se desnaturado e dado origem a dois fragmentos de fita simples, estes últimos podem se parear novamente entre si ou com outros fragmentos distintos, dando continuidade aos eventos de remontagem. A verificação da ocorrência de cruzamentos entre os fragmentos pareados irá sempre depender das extremidades de ambos os parentais onde o pareamento ocorreu, bem como da origem (parental de origem) dessas extremidades.

Linha 65

Caso o pareamento entre os fragmentos selecionados aleatóriamente não tenha ocorrido, a variável que conta o número de tentativas consecutivas de pareamentos sem sucesso é incrementada de um.

Linha 66

Esta linha controla o número de tentativas de pareamentos entre os fragmentos do conjunto Single_DNA dentro de um ciclo de PCR. Caso o número de tentativas de pareamentos sem sucesso seja superior ao Limite ou muitos pareamentos ocorreram de forma que o conjunto Single_DNA tornou-se vazio, as tentativas de pareamentos durante este ciclo de PCR são suspensas e um novo ciclo é iniciado.

Linha 70 e 72

Antes de iniciar um novo ciclo de PCR, os elementos do conjunto Double_DNA são adicionados aos fragmentos restantes do conjunto Single_DNA. Note que o tamanho médio dos fragmentos contidos no conjunto Single_DNA irá aumentar, uma vez que os fragmentos originários do conjunto Double_DNA sofreram algum tipo de extensão.

Neste ponto (Linha 72), todos os elementos do conjunto Double_DNA foram transferidos para o conjunto de fragmentos candidatos à remontagem no próximo ciclo de PCR e por isso este conjunto fica vazio.

5.3 Considerações sobre a convergência do SimAffling e testes preliminares

Esta seção descreve os testes preliminares realizados com o SimAffling com o objetivo de validá-lo *in silico*. São apresentadas execuções do simulador com o propósito de validar sua convergência em termos dos resultados produzidos, ou seja, do número médio de cruzamentos estimado por seqüências *full-length* resultantes do *shuffling* entre dois parentais. No contexto do SimAffling entende-se por seqüências *full-length* todas as seqüências remontadas pelo simulador de DNA *shuffling* cujo tamanho é igual ou maior[34] do que o tamanho do menor dos parentais (caso os parentais tenham tamanhos diferentes).

Como o número de cruzamentos ocorridos durante a remontagem de um fragmento é acumulado ao longo dos ciclos de PCR, é possível calcular o número médio de cruzamento por seqüência *full-length* ao final de cada simulação, bem como ao longo de todas as simulações executadas, como mostram as equações (5.11) e (5.12), respectivamente.

$$\overline{cros}_{sim(i)} = \frac{\sum_{j=1}^{n\_full\_length} n\_cros(seq\_full\_length_j)}{n\_full\_length} \tag{5.11}$$

$$\overline{\overline{cros}} = \frac{\sum_{i=1}^{n\_simulações} \overline{cros}_{sim(i)}}{n\_simulações} \tag{5.12}$$

[34] São consideradas seqüências remontadas cujo tamanho seja, no máximo, duas vezes o tamanho do menor parental.

sendo que n_cros(seq_full_length$_j$) representa o número de cruzamentos na j-ésima seqüência *full-length*; n_full_length é o número de seqüências *full length* obtidas ao final da i-ésima simulação, n_simulações é o número total de simulações executadas.

Seja $E(X) = \mu$ o valor esperado (ou média) para a variável aleatória X que, neste caso, representa o número médio de cruzamentos nas seqüências *full-length*. A variância de X é definida como:

$$\sigma^2 = E((X - \mu)^2) \tag{5.13}$$

A eq.(5.13) define a variância quando o conjunto de observação corresponde a uma população e é chamada de variância populacional. Quando apenas uma mostra de n elementos da população é considerada, tem-se uma variância amostral, que é calculada como mostra a eq. (5.14):

$$\sigma^2 = \frac{1}{n-1}\sum_{i-1}^{n}\left(x_i - \bar{x}\right)^2 \tag{5.14}$$

sendo $\bar{x}$ a média da amostra e x_i o valor da amostra i.

Note que à medida que o número de dados amostrais aumenta, a variância diminui. Para o caso das simulações de DNA *shuffling*, quanto maior o número de simulações (amostras) menor será a variação no número médio de cruzamentos estimado pelo simulador e desta forma, mais próximo o resultado estimado estará do resultado real, considerando que o modelo implementado é representativo. Esta variância irá tender a zero quando o número de simulações tender ao infinito.

Outra medida de dispersão para variáveis aleatórias é o desvio padrão, definido como a raiz quadrada da variância:

$$\sigma = \sqrt{\sigma^2} \tag{5.15}$$

Em simulações é comum estimar, juntamente com o resultado da simulação, um intervalo de confiança associado ao resultado. Um intervalo de confiança (IC) para um parâmetro desconhecido x é definido por:

$$l \le x \le L \tag{5.16}$$

tal que l é o limite inferior do intervalo e L o limite superior do intervalo, ambos dependentes do valor estimado x' para x. Quanto maior o tamanho do intervalo de confiança (L – l) mais garantia tem-se que o intervalo contém o valor verdadeiro de x. Obviamente, quanto maior o intervalo de confiança, menos informação se tem sobre o verdadeiro valor de x e quão próximo o valor estimado x' está de seu valor real.

Seja o valor do intervalo de confiança desejado IC = 95% e assumindo uma amostra aleatória de tamanho n suficientemente grande, a distribuição das médias amostrais ($\bar{x}$) em torno da média populacional é Normal com desvio padrão $\sigma/\sqrt{n}$ e pode ser calculada como mostra a eq. (5.17), considerando a normal padronizada (média nula e desvio padrão unitário N(0,1)):

$$\bar{x} - 1.96 * \frac{\sigma}{\sqrt{n}} \leq \bar{x} \leq \bar{x} + 1.96 * \frac{\sigma}{\sqrt{n}} \qquad (5.17)$$

Observe que o tamanho do IC é inversamente proporcional ao número de simulações (n) e que, quanto maior o número de simulações, menor a variância da amostra (ver eq. (5.14)). Desta forma, quanto maior o número de simulações, menor será o tamanho do intervalo definido na eq. 5.17 e, consequentemente, mais próximo o resultado encontrado estará do resultado real.

Para os testes iniciais com o SimAffling foram utilizadas duas seqüências fictícias seq1 e seq2 de tamanho 90 pb que diferem em quatro pares de bases, distantes uma das outras por 32, 19 e 16 bases, respectivamente, como mostra a Figura 5.7.

```
>seq1
AATAATACGCCATACGACGTACGGGGGGACTACGACGCACATACACGTCAGCAGCATACTACTCTTCTCCGCAACGACGACGTCATCATG
>seq2
AATAATACGCAATACGACGTACGGGGGGACTACGACGCACATAGACGTCAGCAGCATACTACTGTTCTCCGCAACGACGAGGTCATCATG
```

Figura 5.7. Duas seqüências de tamanho 90 pb que diferem entre si em 4 pb, destacadas em preto.

O programa SimAffling foi executado considerando como parentais as seqüências seq1 e seq2 e fragmentos de tamanhos L = 10, 20, 30 e 40 pares de bases, sendo que o tamanho mínimo e máximo dos fragmentos permitidos, para todos os casos, está no intervalo L−10 ≤ L ≤ L+10. Para cada tamanho específico de fragmento, foram realizadas 100, 500 e 1.000 execuções do simulador e exigida uma sobreposição mínima de 6 pb entre dois fragmentos para que um pareamento ocorresse. As simulações permitiram que os resultados fossem comparados, bem como a convergência do simulador avaliado, em cada um dos casos. Para todas as simulações, o

número de cópias das seqüências parentais foi escolhido de forma a resultar, após a fragmentação, em um conjunto Single_DNA com aproximadamente 1.200 fragmentos.

O resultado de cada simulação mostra estimativas da porcentagem de seqüências *full-length* obtidas no processo de DNA *shuffling*, a porcentagem dessas seqüências que foram remontadas como resultado da ocorrência de 0, 1, 2, 3, 4, 5 ou mais cruzamentos e o número médio de cruzamentos esperado nas seqüências remontadas. As tabelas de 5.1 a 5.4 sumarizam os resultados das simulações a traz também o tempo gasto em cada simulação dado em minutos (min). A Figura 5.8 apresenta a tela de execução do software para um dos casos simulados.

Tabela 5.1. Estimativas do SimAffling considerando fragmentos de tamanho 10 pb.

Número de simulações / Tempo gasto na simulação (min)	% seqüência *full-length*	Número médio de cruzamentos por seqüência *full-length*	% seqüências com					
			0 cruz.	1 cruz.	2 cruz.	3 cruz.	4 cruz.	≥ 5 cruz.
100 / 38,56	46,16	1,97	10,65	27,21	31,09	19,67	8,29	3,09
500 / 195,99	45,06	2,00	10,87	26,83	30,55	20,11	8,36	3,26
1.000 / 388,55	45,04	1,98	10,78	27,30	30,58	19,84	8,34	3,16

Tabela 5.2. Estimativas do SimAffling considerando fragmentos de tamanho 20 pb.

Número de simulações / Tempo gasto na simulação (min)	% seqüência *full-length*	Número médio de cruzamentos por seqüência *full-length*	% seqüências com					
			0 cruz.	1 cruz.	2 cruz.	3 cruz.	4 cruz.	≥ 5 cruz.
100 / 44.28	61,66	1,35	19,42	40,00	28,60	10,05	1,77	0,21
500 / 212,02	64,51	1,37	19,55	38,94	29,38	10,02	1,89	0,21
1.000 / 431,10	64,15	1,38	18,98	39,35	29,33	10,19	1,97	0,22

Tabela 5.3. Estimativas do SimAffling considerando fragmentos de tamanho 30 pb.

Número de simulações / Tempo gasto na simulação (min)	% seqüência *full-length*	Número médio de cruzamentos por seqüência *full-length*	% seqüências com					
			0 cruz.	1 cruz.	2 cruz.	3 cruz.	4 cruz.	≥ 5 cruz.
100 / 42,874	82,90	0,81	35,31	50,68	13,45	0,57	0,021	0,00
500 / 225,22	84,98	0,81	35,06	50,84	13,28	0,77	0,048	0,00
1.000 / 441,94	83,97	0,80	35,89	49,28	14,01	0,80	0,008	0,00

Tabela 5.4. Estimativas do SimAffling considerando fragmentos de tamanho 40 pb.

Número de simulações / Tempo gasto na simulação (min)	% seqüência *full-length*	Número médio de cruzamentos por seqüência *full-length*	% seqüências com					
			0 cruz.	1 cruz.	2 cruz.	3 cruz.	4 cruz.	≥ 5 cruz.
100 / 38,27	98,46	0,12	92,70	7,17	0,14	0,00	0,00	0,00
500 / 206,57	98,30	0,10	93,55	6,9	0,16	0,00	0,00	0,00
1.000 / 378,40	98,05	0,09	93,48	6,36	0,16	0,00	0,00	0,00

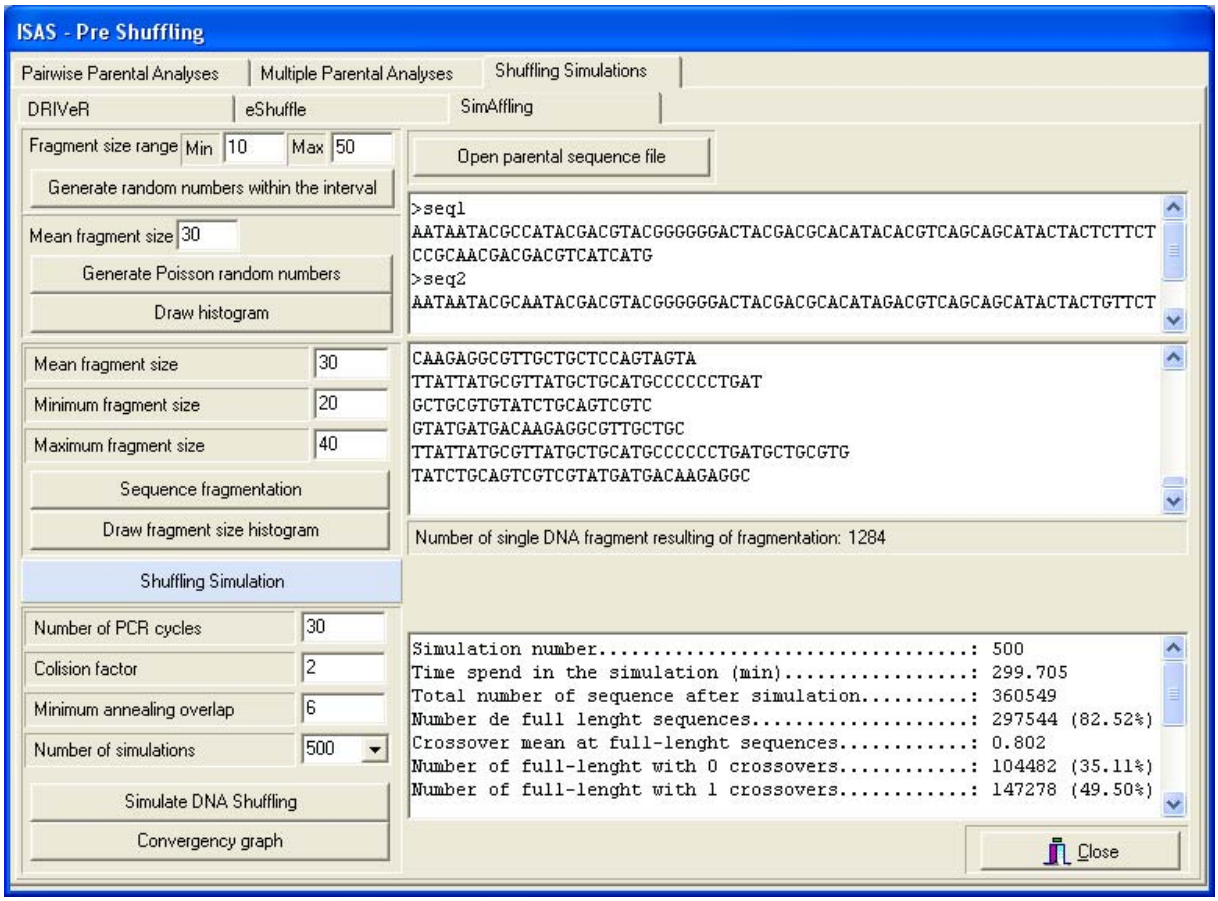


Figura 5.8. Tela de execução do Simulador de DNA shuffling.

É possível observar nos resultados das simulações apresentados nas tabelas de 5.1 a 5.4 que, quanto menor o tamanho dos fragmentos submetidos ao *shuffling*, maior é o número médio de cruzamentos. Contudo, as simulações mostram também que a remontagem de fragmentos menores resulta em uma menor porcentagem de seqüências *full-length* obtidas ao final do processo de DNA *shuffling*. Essa tendência mostrada pelo simulador e representada pelo Gráfico 5.1, está de acordo com o comentário de Volkov e Arnold (2000) de que o número médio de cruzamentos

é inversamente proporcional ao tamanho dos fragmentos, e que fragmentos menores são mais ineficientemente remontados.

A eficiência de remontagem em cada um dos casos simulados pode ser melhor visualizada nos histogramas apresentados nos gráficos de 5.2 a 5.5, os quais apresentam os histogramas dos tamanhos dos fragmentos remontados ao final de 1.000 simulações considerando fragmentos de tamanhos médio 10, 20, 30 e 40 pb, respectivamente. Nos gráficos pode-se observar que a tendência de remontagem segundo o tamanho dos fragmentos, como afirma Volkov, foi capturada pelo modelo proposto e implementado pelo SimAffling. A estimativa da porcentagem de seqüências *full-length* obtida em um experimento de DNA *shuffling* é de extrema importância para que se possa determinar o tamanho da amostra de clones resultantes na qual a busca por recombinantes acontecerá. Experimentos que resultam em uma baixa eficiência de remontagem de *full-length* exigem que uma amostra maior seja avaliada para que possíveis recombinantes sejam encontrados.

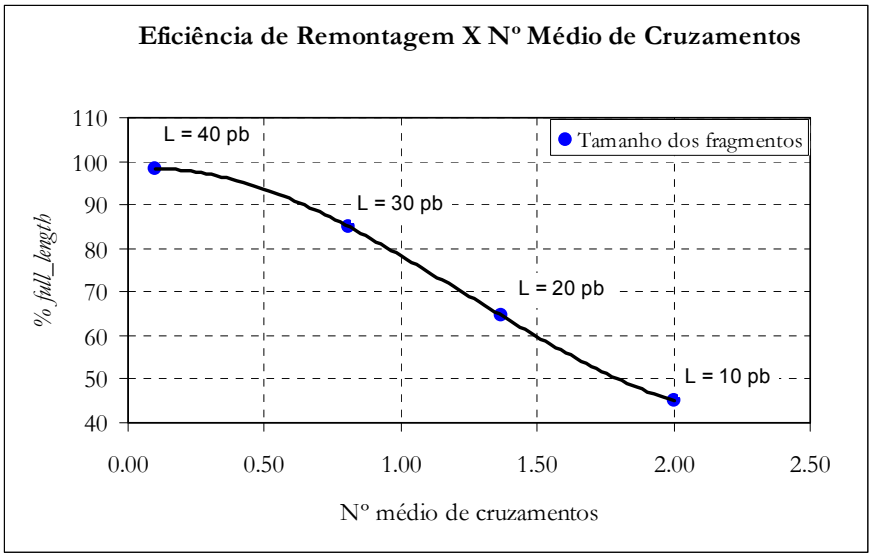


Gráfico 5.1. Número médio de cruzamentos e % de seqüências *full-length* obtidas na simulação do *shuffling* entre os parentais seq1 e seq2, para 500 simulações com diferentes tamanhos iniciais de fragmentos (L), usando o SimAffling.

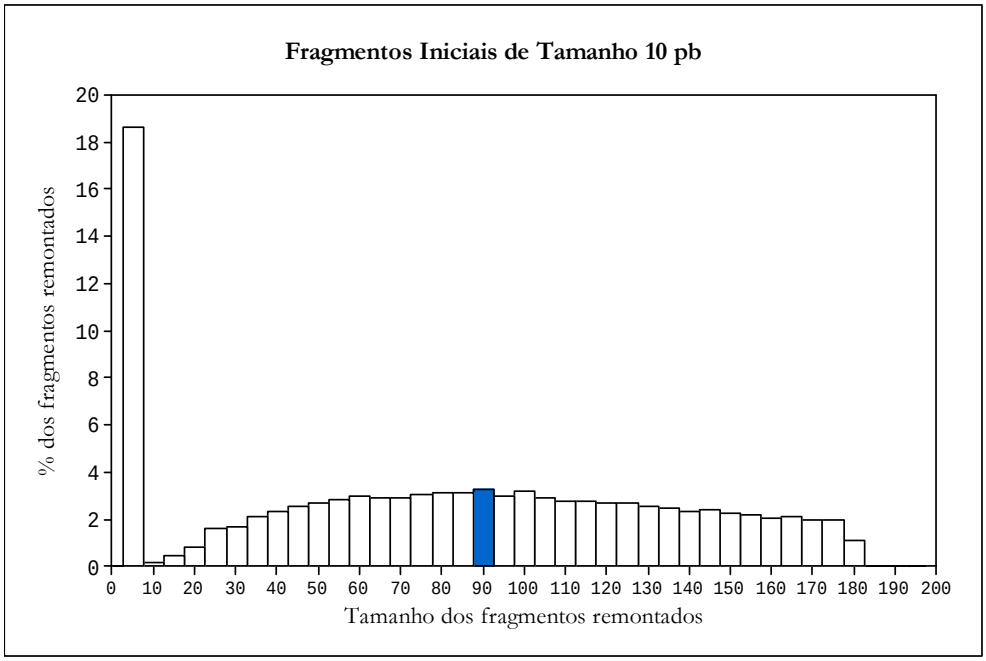


Gráfico 5.2. Histograma do tamanho dos fragmentos remontados ao final de 1.000 simulações partindo-se de fragmentos de tamanho 10 pb. A barra do histograma destacada em azul representa a porcentagem de fragmentos remontados cujo tamanho está entre 87,5 e 92,5 pb.

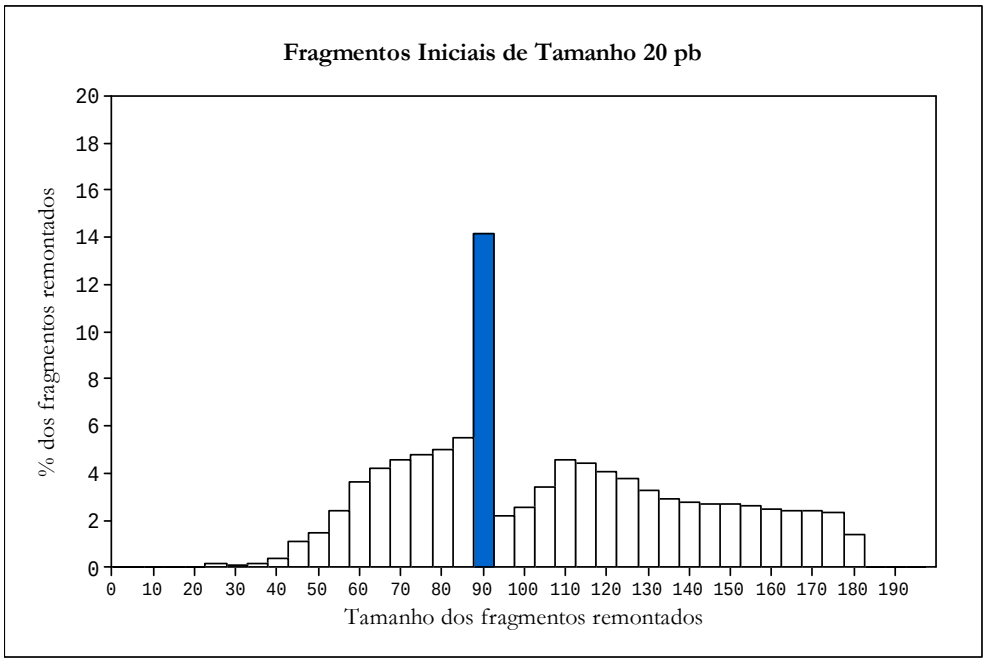


Gráfico 5.3. Histograma do tamanho dos fragmentos remontados ao final de 1.000 simulações partindo-se de fragmentos de tamanho 20 pb. A barra do histograma destacada em azul representa a porcentagem de fragmentos remontados cujo tamanho está entre 87,5 e 92,5 pb.

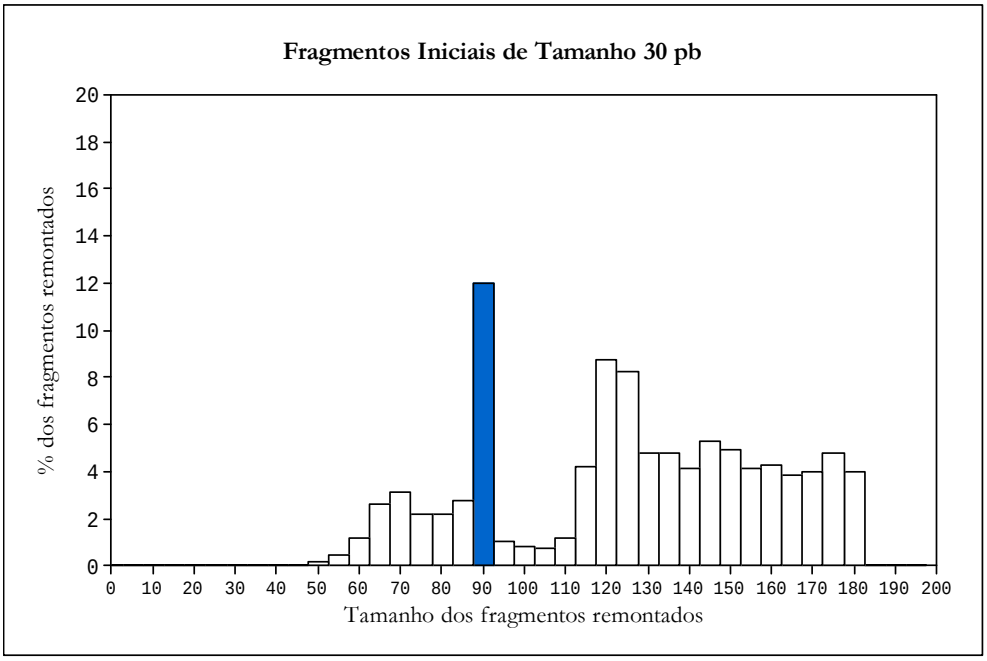


Gráfico 5.4. Histograma do tamanho dos fragmentos remontados ao final de 1.000 simulações partindo-se de fragmentos de tamanho 30 pb. A barra do histograma destacada em azul representa a porcentagem de fragmentos remontados cujo tamanho está entre 87,5 e 92,5 pb.

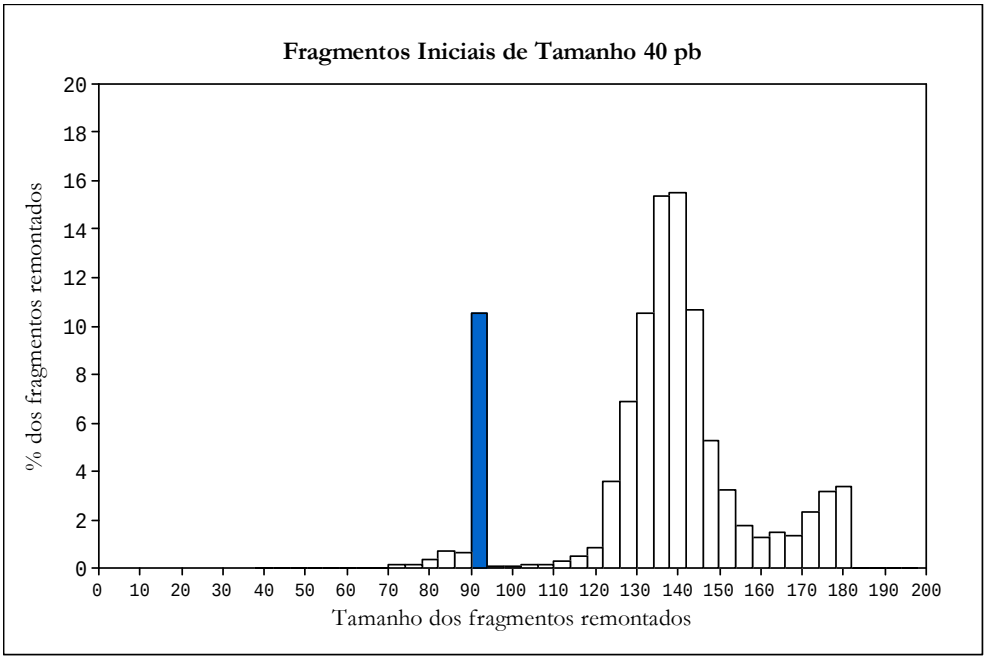


Gráfico 5.5. Histograma do tamanho dos fragmentos remontados ao final de 1.000 simulações partindo-se de fragmentos de tamanho 40 pb. A barra do histograma destacada em azul representa a porcentagem de fragmentos remontados cujo tamanho está entre 90,0 e 94,0.

Considerando o número médio de cruzamentos nas seqüências *full-length*, para um mesmo tamanho de fragmento inicial, observou-se uma pequena diferença nos resultados encontrados ao

final de 100, 500 e 1.000 simulações. Contudo, essas diferenças não são significativas e podem ser atribuídas a não convergência do simulador quando da utilização de um número insuficiente de simulações. Os gráficos de 5.6 a 5.8 apresentam a convergência em relação ao número médio de cruzamentos nas seqüências *full-length* obtidas nas simulações considerando fragmentos de tamanho 30 pb (Tabela 5.3) e o intervalo de confiança (95%). Nos gráficos, a linha azul representa a média acumulada dos cruzamentos observados ao longo das simulações e as linhas pretas, abaixo e acima da linha azul representam, respectivamente, o limite inferior e superior do intervalo de confiança. Observe nos gráficos de convergência que a confiança de que o resultado encontrado pelo simulador está próximo do resultado real do processo aumenta com o número de simulações, uma vez que o intervalo de soluções 95% confiáveis fica cada vez menor. Este intervalo tende a zero no infinito, ou seja, para um modelo de simulação sem erros (perfeito) o seu resultado será igual ao resultado real do processo sendo simulado.

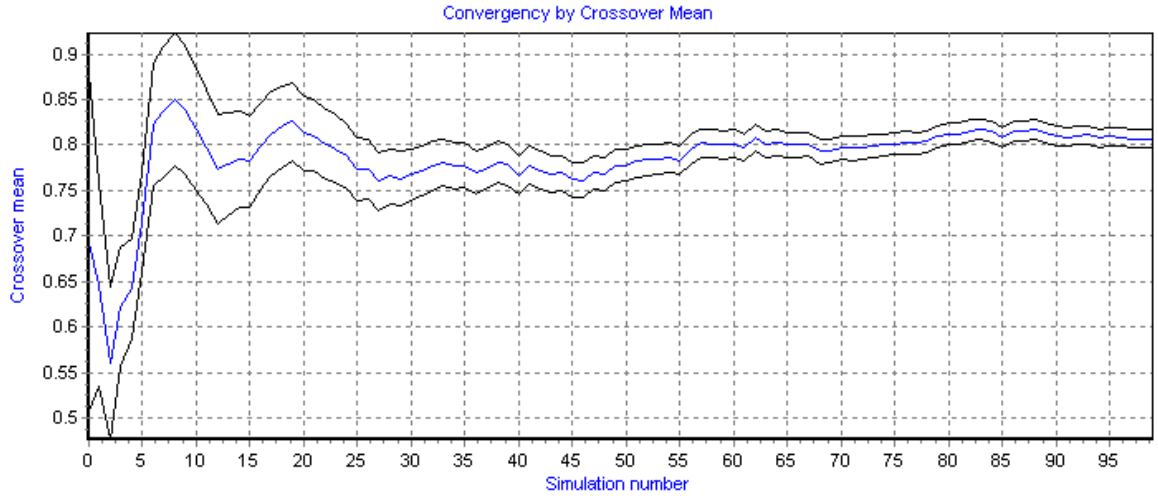


Gráfico 5.6. Convergência do simulador ao longo de 100 simulações.

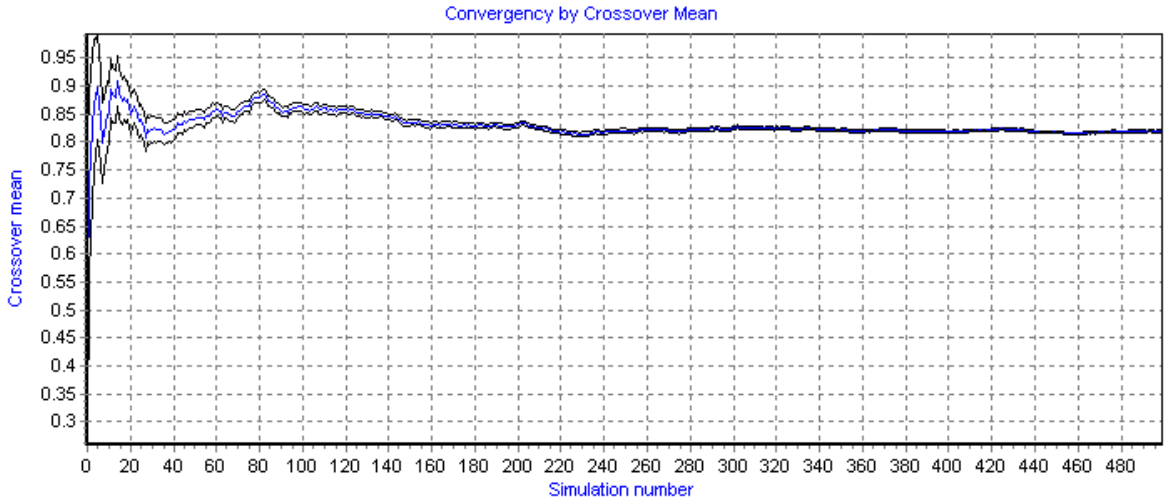


Gráfico 5.7. Convergência do simulador ao longo de 500 simulações.

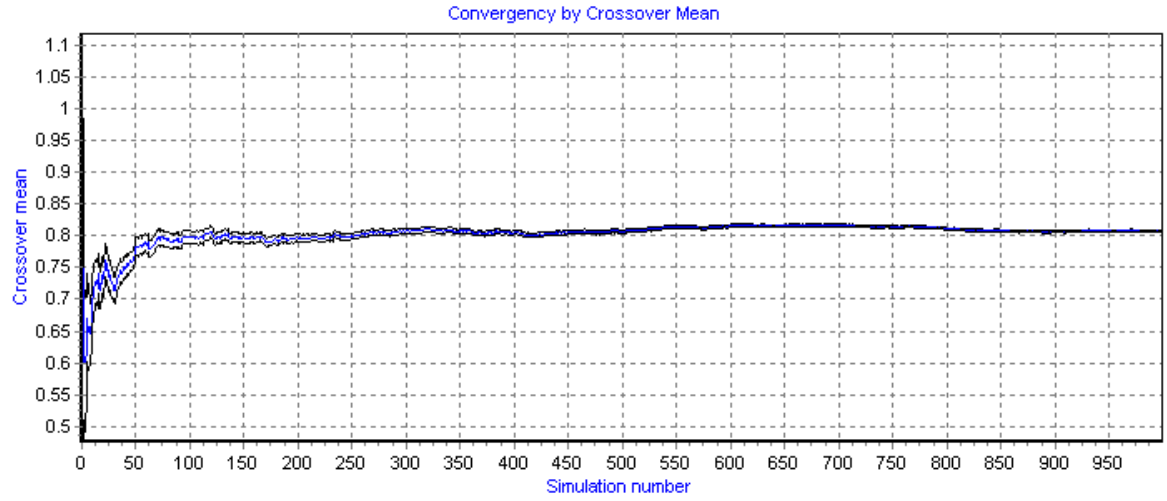


Gráfico 5.8. Convergência do simulador ao longo de 1.000 simulações.

Apesar do número médio de cruzamentos ainda sofrer certa variação após 100 simulações (ver Gráfico 5.6) quando comparada com a variação para os casos de 500 e 1.000 simulações, estas flutuações são pequenas, cerca de ± 0,02, e o resultado do simulador, também para este caso, pode ser aceito como estimativas para o *shuffling* de DNA sendo simulado. Observe na Tabela 5.3 que o tempo médio gasto em 100 simulações é cerca de 80% menor que o tempo gasto em 500 simulações e 90% menor que o tempo gasto em 1.000 simulações.

Diferenças pequenas também foram observadas no número estimado de seqüências *full-length* com 0, 1, 2, 3, 4, ou mais do que 5 cruzamentos em todos os casos simulados. É preciso lembrar, entretanto, que os resultados estimados pelo simulador devem servir apenas como direções para o profissional que pretende realizar um experimento de DNA *shuffling* e não como

valores determinísticos, por isso devem ser interpretados de forma crítica. Através das simulações espera-se obter informações relevantes como, por exemplo, qual tamanho médio de fragmentos são mais facilmente remontados ou ainda, qual tamanho dos fragmentos resulta em um maior número de cruzamentos por seqüências *full-length*.

Nas simulações descritas até o momento, o número de cópias das seqüências parentais utilizadas foi tal que um conjunto de aproximadamente 1.200 fragmentos foi produzido após a fragmentação dos parentais. Para verificar a influência do número de cópias dos parentais no número médio de cruzamento estimado pelo simulador, bem como no tempo gasto na simulação, foram realizados testes para a remontagem de fragmentos de tamanho 30 pb e 500 simulações, sendo o conjunto de fragmentos a serem remontados de tamanhos no intervalo entre 200 e 2.500 fragmentos. Os resultados dessas simulações estão sumarizados no Gráfico 5.9. Os pontos em azul representam o número médio de cruzamentos estimados e os pontos em vermelho, o tempo gasto nas simulações. Uma regressão não linear foi realizada em ambos os resultados das simulações (número médio de cruzamentos e tempo de simulação) para que as linhas de tendência em cada um dos casos fossem também traçadas no gráfico.

Observe no Gráfico 5.9 que a linha de tendência para o número médio de cruzamentos mostra que maiores variações ocorrem quando o conjunto de fragmentos a serem remontados tem tamanho entre 100 e 1.000 e que, para valores acima de 1.000, esta flutuação tende a desaparecer. Pelas simulações realizadas pode-se dizer que o tempo gasto é uma potência do número de fragmentos. Apesar de produzirem resultados similares, o tempo gasto nas simulações utilizando-se mais do que 1.200 fragmentos é consideravelmente maior do que o tempo gasto na simulação com 1.200 fragmentos e, por este motivo, optou-se por realizar a fragmentação das seqüências parentais de tal forma que o tamanho do conjunto de fragmentos a serem submetidos aos ciclos de remontagem (ciclos de PCR) seja em torno de 1.200.

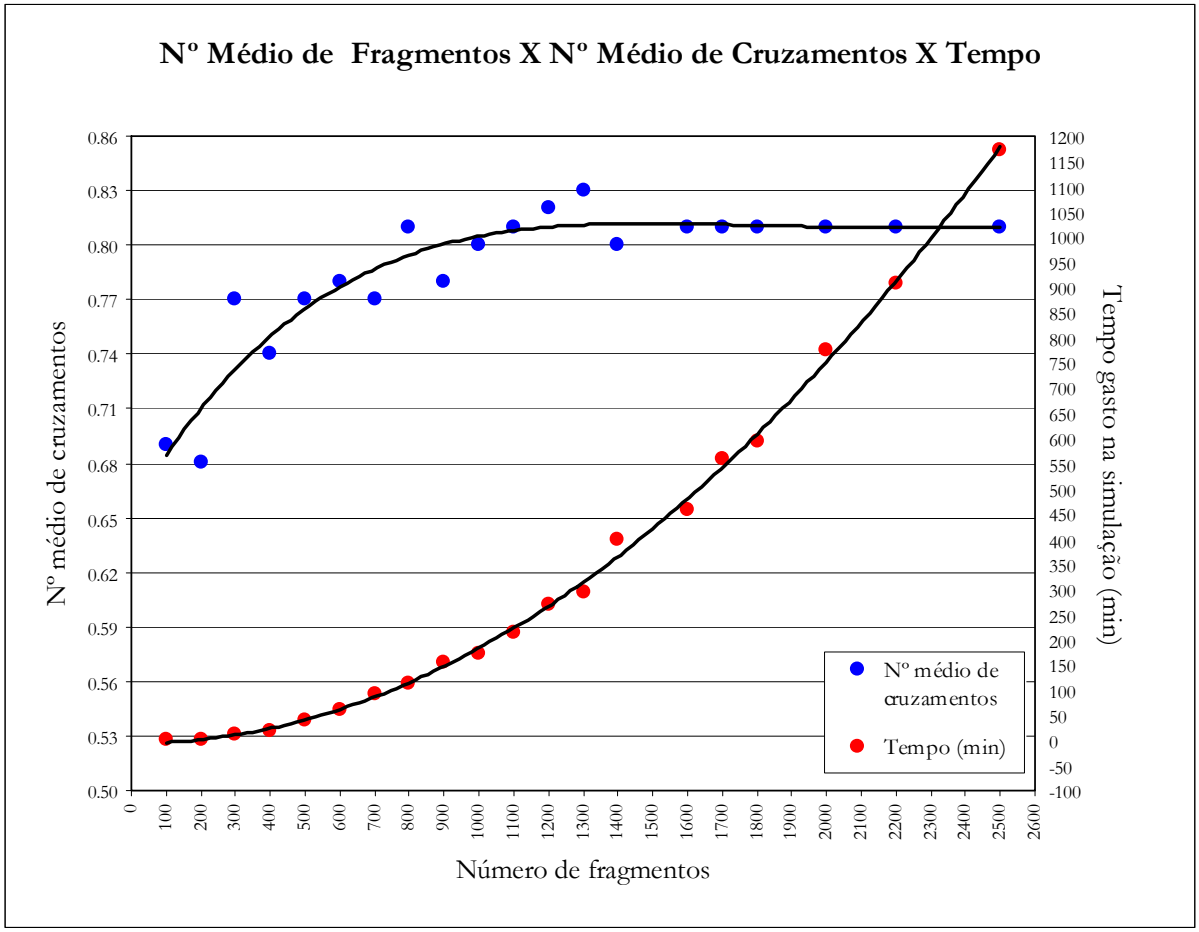


Gráfico 5.9. Relação entre o número médio de cruzamentos, o número de fragmentos a serem remontados e o tempo gasto na simulação.

5.4 Comparação entre resultados simulados e experimentais

A validação do modelo computacional proposto para a simulação e predição de resultados de experimentos de DNA *shuffling*, assim como a avaliação de qualquer outro modelo de simulação, deve ser feita através da comparação entre resultados experimentais e resultados teóricos produzidos pelo modelo. Contudo, como comentado em Joern et al. (2002), poucos dados referentes à composição das seqüências resultantes de experimentos de *shuffling* estão disponíveis, o que pode comprometer a avaliação de modelos para este fim. Como resultado de experimentos de DNA *shuffling*, bem como de outros experimentos de Evolução Molecular Direta, tem-se, na maioria das vezes, a descrição e caracterização apenas do 'melhor' clone recombinante (ou de um pequeno número de melhores clones) em relação à propriedade de interesse (atividade, estabilidade ou resistência à determinada droga, por exemplo). Entretanto, para a validação apropriada de modelos computacionais, seriam necessários dados referentes a uma amostra

significativa de seqüências resultantes do *shuffling* (recombinantes ou não) bem como a descrição das condições nas quais o experimento foi realizado.

A apresentação dos modelos implementados pelos softwares DRIVeR e eShuffle, como feita em suas publicações originais ((PATRICK et al., 2003) e (MOORE et al., 2001)), respectivamente, dá-se pela apresentação dos formalismos/técnicas utilizadas pelo modelo para predizer/simular o DNA *shuffling* seguido pela sua utilização para predição de resultados de experimentos de DNA *shuffling* reportados na literatura, de forma que os resultados experimentais e teóricos possam ser comparados e o modelo validado. Essa mesma metodologia foi utilizada para validar o SimAffling. Os parentais utilizados nesta fase de validação estão descritos na Tabela 5.5 e são os mesmos utilizados na validação dos modelos DRIVeR, eShuffle e ShuffIt.

Tabela 5.5. Seqüências parentais utilizadas para validação de modelos para o DNA *shuffling*.

	Parentais	Descrição
[1]	KF707 e LB400	Genes cujas seqüências compartilham 96% de identidade (KUMAMARU et al., 1998).
[2]	PurN e GarT	Genes cujas seqüências compartilham 50% de identidade (OSTERMEIER et al., 1999).
[3]	AtzA e TriA	Genes parentais que diferem em apenas 9 pares de bases (RAILLARD et al., 2001).
[4]	Human e Murine 1L-1β gene	Genes cujas seqüências compartilham 75% de identidade (STEMMER, 1994b).

No experimento de DNA *shuffling* descrito por Kumamaru et al. (1998), foram utilizados fragmentos de DNA de tamanhos no intervalo de 10 a 50 pb. Seis clones resultantes foram selecionados e seqüenciados e uma média de 3,8 cruzamentos por seqüência foi observado. O software eShuffle estimou um número médio de 2,8 cruzamentos nas seqüências resultantes. O SimAffling, considerando fragmentos de tamanho aleatório entre 10 e 50 pb, estimou uma média de 2,7 cruzamentos por seqüência. É importante, entretanto, observar que a amostra de seqüências resultantes apresentada por Kamamaru, assim como pela maioria dos experimentos de DNA *shuffling* reportados na literatura, é uma amostra pequena e, além disso, clones seqüenciados para os quais cruzamentos não foram observados podem ter sidos omitidos dos resultados apresentados. Esse fato pode colaborar para a diferença observada entre os resultados experimentais e *in silico*. Esse experimento de DNA *shuffling* também foi simulado utilizando o

software DRIVeR para λ^{true} = 6, uma biblioteca resultante de tamanho L = 2.000, e as mutações entre os parentais localizadas nas posições 1, 701, 730, 748, 763, 777, 803, 819, 833, 860, 875, 894, 910, 924, 937, 953, 967 e 1.005 (as mutações foram identificadas como descrito no Capítulo 4 Seção 4.2.2.1 sendo NEBP = 9). DRIVeR estima que, em média, 1,7 cruzamentos ocorrem nas seqüências *full-length*. Para que o valor estimado pelo DRIVeR se iguale ao número médio de cruzamentos nos clones seqüenciados e reportados por Kamamaru, o valor de λ^{true} deveria ser 20. Como visto no Capítulo 3, valores de λ^{true} muito altos são dificilmente alcançados na prática, por isso, simulações com valores muito altos de λ^{true} não devem ser consideradas.

No experimento de DNA *shuffling* descrito por Ostermeier et al. (1999), a fragmentação das seqüências parentais (PurN e GarT) foi realizada da mesma forma como na metodologia ITCHY, descrita no Capítulo 2, Seção 2.10, de forma que, teoricamente, todos os possíveis tamanhos de fragmentos variando de 1 até o tamanho da seqüência parental foram utilizados para remontagem. Ostermeier relata que, das 10 seqüências randomicamente selecionadas e seqüenciadas, todas elas continham apenas 1 cruzamento. Moore (2001) afirma que a utilização do eShuffle considerando como parentais os genes PurN e GarT resultou numa relação menor do que 10^{-9} entre o número de seqüências remontadas com mais do que 3 cruzamentos e apenas 1 cruzamento. O tamanho dos fragmentos e a temperatura de pareamento utilizados na execução do eShuffle não foi relatado para esta simulação, bem como para as demais simulações apresentadas nesta seção. Para realizar a simulação do experimento com o SimAffling, foram utilizados fragmentos de tamanho médio 35 pb, sendo o número médio de cruzamentos nas seqüências *full-length* de 0,094 e 90% destas contendo 0 cruzamentos e 8,6% contendo apenas 1 cruzamento. Utilizando-se fragmentos randômicos entre 10 e 50 pb, o número médio de cruzamentos estimado pelo SimAffling foi de 0,1. Como descrito no Capítulo 3, Seção 3.2.2, o número de mutações entre os parentais não pode ser superior a 20 para que a execução do DRIVeR seja possível. Desta forma, a execução do DRIVeR para esse tipo de seqüências parentais (pouco similares) não é recomendada, uma vez que o número de mutações entre elas será grande. Mesmo agrupando mutações consecutivas próximas umas das outras, os resultados podem não ser significativos, como no caso desses parentais (PurN e GarT) e λ^{true} = 3, para os quais o número médio de cruzamentos estimado foi de 2,3.

Raillard et al. (2001) descreve o *shuffling* de dois genes (Atz e TriA) que diferem apenas em 9 pares de bases distantes uma das outras por 8, 33, 92, 2, 34, 2, 79 e 3 pb. O tamanho dos fragmentos utilizados no experimento não foi relatado. Assumiu-se que fragmentos de tamanhos entre 10 e 50 pb foram utilizados. Vinte e cinco clones foram selecionados e seqüenciados.

Nestas 25 seqüências, uma média de 2,3 cruzamentos ocorreu. A execução do SimAffling para fragmentos de tamanho 35 pb resultou num número médio de cruzamentos de 3,2, enquanto que a simulação com fragmentos de tamanho 45 pb resultou em 3,0. Foi realizada também a simulação considerando a fragmentação aleatória e fragmentos de tamanho mínimo 10 e máximo 50 pb e o número médio de cruzamentos estimado foi também de 3,2. Patrick utilizou-se também dessas seqüências na validação do DRIVeR, sendo 2 o número médio de cruzamentos estimado pelo modelo. Executamos o eShuffle para essas seqüências considerando o tamanho dos fragmentos de 50 pb e a temperatura de pareamento de 55ºC e o número médio de cruzamentos estimado foi de 3,2. Para fragmentos de tamanho 35 pb, eShuffle estimou 3,5 cruzamentos. A superestimação de resultados observada na simulação *in silico* desse experimento pelo eShuffle e pelo SimAffling pode ter sido o resultado da ocorrência de cruzamentos silenciosos nas seqüências *full-length*, não detectados experimentalmente mas contabilizados por estes dois softwares. Outro fator a ser considerado é que as condições experimentais referentes ao tamanho dos fragmentos e temperatura de pareamento não puderam ser reproduzidas pelos softwares, uma vez que não foram informadas.

Stemmer (1994b) relatou que o *shuffling* entre os genes 1L-1β de humano e macaco (*murine*) resultou em seqüências remontadas com, em média, 1,9 cruzamentos. Devido à baixa similaridade entre as seqüências[35], uma baixa temperatura de pareamento (25ºC) foi utilizada durante o experimento. A execução do DRIVeR para esses genes utilizando $\lambda^{true} = 3$, uma biblioteca de tamanho 2.000 e sendo de 7 o número de bases idênticas exigidas entre mutações consecutivas no alinhamento (o que resultou em 20 mutações) resultou num número médio estimado de 2,5 cruzamentos por seqüência. Para $\lambda^{true} = 2$, o número estimado de cruzamentos é de 1,8. O valor reportado na literatura para o número médio de cruzamentos estimado pelo eShuffle para estes genes é de 1,5. A fim de representar a baixa temperatura de pareamento utilizada neste experimento, a exigência mínima de sobreposição entre os fragmentos possíveis de pareamento durante a execução do SimAffling foi reduzida para 4 pb, enquanto que o valor 6 pb foi utilizado em todos os outros experimentos descritos nesta seção. O resultado estimado pelo SimAffling para este caso, considerando fragmentos de tamanho médio entre 10 e 50 pb foi de 1,2 cruzamentos em média por seqüência *full-length*.

Os resultados descritos nos parágrafos anteriores estão sumarizados na Tabela 5.6.

[35] As mutações estão separadas em média por 4,1 pb.

Tabela 5.6. Comparação entre estimativas e valores experimentais para casos de estudo de DNA *shuffling*.

	Experimento	DRIVeR	eShuffle	SimAffling			
				Frag. de 10 a 50 pb	Frag. de tamanho médio 25 pb	Frag. de tamanho médio 35 pb	Frag. de tamanho médio 45 pb
[1]	3,8	1,7	2,8	2,7	2,7	2,9	2,7
[2]	1,0	2,3	$<10^{-9}$	0,1	0,2	0,1	0,05
[3]	2,3	2,0	3,2	3,2	3,4	3,2	3,0
[4]	1,9	1,8	1,5	1,2	1,5	1,0	0,73

Para todos os casos de simulação apresentados na Tabela 5.6, os resultados do SimAffling foram praticamente os mesmos dos resultados retornados pelo eShuffle, sobretudo quando fragmentos de tamanho entre 10 e 50 pb e tamanho médio 35 pb foram utilizados. Os resultados de todos os três softwares apresentaram variações em comparação com os resultados experimentais. Contudo, deve-se ressaltar que as estimativas do DRIVeR são o resultado de diversas execuções do software na tentativa de fornecer parâmetros de entrada (especialmente λ^{true}) adequados à produção de resultados similares aos experimentais, enquanto que as estimativas do eShuffle e do SimAffling, são o resultado da reprodução das condições nas quais o experimento de fato ocorreu, ao menos para os casos nos quais tais condições foram descritas.

Dois fatores principais podem contribuir para as divergências observadas entre os resultados teóricos e experimentais: a não reprodução exata, no momento da execução dos softwares, das condições nas quais o experimento foi realizado, uma vez que a descrição de tais condições não foram completamente relatadas para alguns dos experimentos utilizados; e o pequeno número de amostras coletadas/seqüenciadas da biblioteca de *shuffling* resultante. Adicionalmente, deve-se considerar que os experimentos de DNA *shuffling* encontrados na literatura descrevem, na sua maioria, apenas as seqüências encontradas nas quais cruzamentos ocorreram, não reportando os clones encontrados nos quais não foi observado nenhum cruzamento, o que contribui para o aumento do número médio de cruzamentos observado na amostra relatada e, conseqüentemente, para a subestimação do número médio de cruzamentos pelos softwares, como ocorreu, possivelmente, na comparação dos resultados experimentais e simulados dos experimentos [1], [2] e [4], descritos na Tabela 5.6. É preciso lembrar ainda que os softwares não fazem distinção entre cruzamentos silenciosos e cruzamentos não-silenciosos, de forma que a ocorrência de ambos é considerada no cálculo do número médio de cruzamentos

estimados e que, em contrapartida, os cruzamentos silenciosos não são detectados durante a análise da amostra. Desta forma, em casos como esse, o software pode superestimar o número médio de cruzamentos nas seqüências resultantes, como pode ter ocorrido, possivelmente, na comparação entre os resultados experimentais e simulados do experimento [3].

5.5 Considerações Finais

O modelo proposto e implementado pelo software SimAffling para a predição de resultados de experimentos de DNA *shuffling* modela os principais eventos envolvidos neste tipo de experimentos: fragmentação das seqüências parentais; colisão entre os fragmentos de fita simples de DNA; possibilidade de ocorrência de pareamento entre fragmentos que colidiram; e extensão de fragmentos pareados. As condições reais (ou desejadas) de temperatura de pareamento e tamanho dos fragmentos submetidos aos ciclos de PCR são parâmetros de entrada do modelo, de forma que, diferentes condições podem ser simuladas e os resultados avaliados a fim de determinar qual condição, possivelmente, é a melhor condição para a realização do experimento.

A temperatura de pareamento do experimento a ser simulado é representada pelo tamanho mínimo de sobreposição exigida entre dois fragmentos para que um pareamento ocorra, uma vez que se sabe que sob temperaturas mais baixas, o pareamento entre menores regiões de sobreposição é favorecido, enquanto que temperaturas mais altas favorecem o pareamento entre regiões de maior sobreposição.

Como os fragmentos candidatos ao pareamento são aleatoriamente selecionados de um conjunto de fragmentos possíveis de se parearem, é possível que a eficiência da remontagem dos fragmentos resultantes em *full-length*, bem como dos fragmentos remontados cujo tamanho é menor ou maior que o tamanho dos parentais sejam avaliados ao final da simulação. A eficiência de remontagem é uma estimativa tão importante quanto o número médio de cruzamentos observados nas seqüências *full-length*, uma vez que o tamanho da amostra a ser avaliada na busca por recombinantes deve estar relacionado com a eficiência de remontagem do experimento realizado.

6 Capítulo

Conclusões

"Se é para buscar abrigo, que seja sob uma árvore grande."
Provérbio Japonês

Técnicas de Evolução Molecular Direta permitem que genes de interesse sejam modificados em laboratório a fim de produzir novos genes com propriedades melhoradas ou ainda com novas propriedades. Dentre as diversas técnicas de Evolução Molecular Direta, ou simplesmente, evolução *in vitro*, a técnica de DNA *shuffling* tem sido aplicada com sucesso em inúmeros experimentos descritos na literatura com os mais diferentes objetivos como, por exemplo, aumento da especificidade, termoestabilidade e atividade de enzimas, aumento de resistência a determinadas drogas e/ou agentes patológicos e aumento da eficiência de vacinas. Dados um ou mais genes de interesse (ou parentais), esses genes são fragmentados e, em seguida, remontados por meio de ciclos de PCR. Dentre as seqüências resultantes, espera-se encontrar seqüências remontadas com fragmentos originários de parentais distintos. Quando fragmentos originários de parentais distintos são remontados em um único fragmento, é dito que um cruzamento entre os parentais ocorreu. O número de cruzamentos ocorridos nas seqüências remontadas cujo tamanho é, aproximadamente, o mesmo tamanho dos parentais (ditas *full-length*), pode ser utilizado como uma estimativa do sucesso ou não do experimento de *shuffling*.

A determinação das condições ideais à realização de um experimento de DNA *shuffling* envolve a avaliação de diversos parâmentros tais como tamanho dos fragmentos, temperaturas de pareamento e desnaturação utilizadas durante os ciclos de PCR, concentração das seqüências parentais, entre outros. A simulação *in silico* de um experimento de DNA *shuffling*, segundo modelos específicos, permite que otimizações sejam implementadas antes que o experimento seja

realizado. Esse trabalho apresentou um estudo sobre quatro modelos para o processo de *shuffling*. O estudo e descrição dos modelos existentes colaboraram para que um novo modelo fosse proposto e implementado como um software, que recebeu o nome de SimAffling.

Além da ferramenta de simulação SimAffling, um ambiente de apoio e suporte a três etapas relacionadas a experimentos de DNA *shuffling* foi desenvolvido. A ferramenta foi denominada ISAS (*Interactive Software for Assisting DNA Shuffling Processes*) e agrupa diversas funcionalidades em três módulos distintos: módulo para análise das seqüências parentais; módulo para simulação de experimentos de DNA *shuffling*; e módulo para análise das seqüências resultantes de experimentos de DNA *shuffling*. A análise dos parentais é uma etapa essencial que realiza a identificação das regiões de similaridade e diferenças entre as seqüências e, desta forma, permite uma avaliação prévia da adequabilidade das seqüências como parentais. Uma medida específica para a avaliação da adequabilidade de pares de seqüências candidatas a parentais foi proposta e sua utilização apresentou resultados promissores. O módulo de simulação de experimentos de DNA *shuffling* permite que experimentos deste tipo sejam simulados utilizando o SimAffling, software que implementa o modelo proposto neste trabalho de pesquisa, e por dois outros softwares chamados DRIVeR e eShuffle, que implementam dois dos quatro modelos encontrados na literatura e descritos neste trabalho. Por fim, o módulo de análise das seqüências parentais viabiliza a busca por seqüências recombinantes dentre todas as seqüências resultantes do *shuffling*.

Os resultados obtidos com o SimAffling durante os testes de validação apresentaram concordância aceitável com os resultados obtidos experimentalmente, bem como com os resultados obtidos pelos softwares DRIVeR e eShuffle. Além da estimativa do número médio de cruzamentos entre as sequências *full-length*, o SimAffling estima ainda a porcentagem desse tipo de seqüência na biblioteca resultante, sendo assim possível um estudo da eficiência de remontagem, que é dependente da composição das seqüências parentais e do tamanho dos fragmentos utilizados.

O principal objetivo da incorporação do DRIVeR e do eShuffle no ambiente do ISAS foi o de desenvolver uma interface gráfica para a utilização desses softwares e, assim, facilitar sua utilização, uma vez que as implementações originais não dispunham de interface com o usuário. Adicionalmente, devido às diferentes abordagens ao processo de DNA *shuffling* adotadas em cada um dos modelos, a utilização paralela destes permite um estudo mais detalhado sobre o experimento a ser realizado e, possívelmente, que otimizações sejam implementadas com base nos resultados desse estudo.

Apesar das inúmeras aplicações do DNA *shuffling*, poucos trabalhos teóricos direcionados ao estudo e modelagem do processo de DNA *shuffling* são encontrados na literatura. Por esse motivo, espera-se que a implementação da ferramenta ISAS, a caracterização e detalhamento dos quatro modelos para simulação de experimentos de DNA *shuffling* existentes e a proposta e implementação de um novo modelo de simulação descritos neste trabalho de pesquisa possa auxilizar/direcionar pesquisadores que trabalham com experimentos de DNA *shuffling*, bem como servir de apoio para outros trabalhos de modelagem e simulação *in silico* de DNA *shuffling*.

Como continuidade do trabalho de pesquisa realizado, pretende-se direcionar esforços no sentido de realizar algumas mudanças na implementação do SimAffling, principalmente no que diz respeito ao não tratamento da ocorrência de cruzamentos silenciosos e à implementação da fase de PCR com *primers*, a fim de verificar a influência desta fase nos resultados produzidos pelo modelo.

BIBLIOGRAFIA

ALBERTS, B.; JOHNSON, A.; LEWIS, J.; RAFF, M.; ROBERTS, K.; WALTER, P. **Molecular biology of the cell**. 4ª ed, Garland Science (Taylor & Francis Group), 2002, 1616p.

ALLAWI, H. T.; SANTALUCIA, J. Jr. **Thermodynamics and NMR of internal G·T mismatches in DNA**. Biochemistry, vol. 36, 1997, p. 10581–10594.

ALLAWI, H. T.; SANTALUCIA, J. Jr. **Thermodynamics of internal C·T mismatches in DNA**. Nucleic Acids Research, vol. 26(11), 1998, p. 2694–2701.

ANG, A. H-S.; TANG, W. H. **Probabilistic Concepts in Engineering Planning and Design**. Vol. 1, Basic Principles, J. Wiley, 1975, 406p.

BALDWIN, A. J.; BUSSE, K.; SIMM, A. M.; JONES, D.D. **Expanded molecular diversity generation during directed evolution by trinucleotide exchange (TriNEx)**. Nucleic Acids Research, 2008, vol. 36(13) e77.

BANKS, J.; CARSON, J.S.; NELSON, II B.L.; NICOL, D.M. **Discrete-Event System Simulation**. 5ª ed, Prentice Hall, 2001, 594p.

BECK, A. T. **Curso de Confiabilidade Estrutural**. Notas de aula Universidade de São Paulo, Escola de Engenharia de São Carlos, Departamento de Engenharia de Estruturas, 2006, 210 p.

BRANNINGAN, J.A.; WILKINSON A.J. **Protein engineering 20 years on**. Nature Reviews Molecular Cell Biology, vol. 3(12), 2002, p. 964–970.

BRESLAUER, K. J.; FRANK, R.; BLOCKER, H.; MARKY, L. A. **Predicting DNA duplex stability from the base sequence**. Proc Natl Acad Sci USA, vol. 83, 1986, p. 3746–3750.

BROWN, T. A. **Genomes**. Jhon Wiley & Sons, 1999, 472p.

CADWELL, R.C.; JOYCE, G.F. **Randomization of Genes by PCR**. PCR Methods and Applications, vol. 2, 1992, p. 28–33.

CAMPBELL, N. A., REECE, J. B.; MITCHELL, L.G. **Biology.** 5ª ed, Benjamin & Cummings, 1999, 1251p.

CASTLE, L.A.; SIEHL, D.L.; GORTON, R.; PATTEN, P.A.; CHEN, Y.H.; BERTAIN, S.; CHO, H-J.; DUCK, N.; WONG, J.; LIU, D.; LASSNER, M.W. **Discovery and directed evolution of a glyphosate tolerance gene.** Scince, vol. 304, 2004, p. 1151–1154.

CHANG, C. C.; CHEN, T. T.; COX, B. W.; DAWES, G. N.; STEMMER, W. P. C.; PUNNONEN, J.; PATTEN, P. A. **Evolution of a cytokine using DNA family shuffling**. Nature Biotchnology, vol. 17(8), 1999, p. 793–797.

CHELLISERRYKATTIL, J.; ELLINGTON, A.D. **Evolution of a T7 RNA polymerase variant that transcribes 2'-O-methyl RNA** Nature Biotechnology, vol. 22, 2004, p. 1155–1160.

CHEN, R. Q.; JIN, Y.; WU, J. B.; ZHOU, X. D.; LI, D. S.; LU, Q. M.; WANG, W. Y.; XIONG, Y. L.

A novel high molecular weight metalloproteinase cleaves fragment F1 of activated human prothrombin. Toxicon vol. 44, 2004, p. 281–287.

CILIBRASI, R.; VITANYI, P. M. B. **Clustering by compression**. IEEE Transactions on Information Theory vol. 51(4), 2005, p. 1523–1545.

COCO, W. M.; LEVINSON, W. E.; CRIST, M. J.; HEKTOR, H. J.; DARZINS A.; PIENKOS, P. T.; SQUIRES, C. H.; MONTICELLO, D. J. **DNA shuffling method for generating highly recombined genes and evolved enzymes**. Nature Biotechonolgy, vol. 19, 2001, p. 354–359.

COSTA, A. S. **Expressão heteróloga, purificação e estudos de atividade de uma proteína inibidora de Cisteíno Protease da cana-de-açúcar e posterior evolução *in vitro* pela técnica de DNA *Shuffling***. Tese de Doutoramento, Universidade Federal de São Carlos, SP, 2004, 119p.

CRAMERI, A.; RAILLARD, S. A.; BERMUDEZ, E.; STEMMER, W. P. C. **DNA shuffling of a family of genes from diverse species accelerates directed evolution**. Nature, vol. 391, 1998, p. 288–291.

CRICK, F. H. C.; WATSON, J. D. **The complementary structure of deoxyribonucleic acid. Medical Research Council Unit for the Study of the Molecular Structure of Biological Systems.** Cavendish Laboratory, University of Cambridge, 1953, p. 80–96.

CROTHERS, D.M.; ZIMM, B.H. **Theory of the Melting Transition of Synthetic Polynucleotides: Evaluation of the Stacking Free Energy**. Journal of Molecular Biology, vol. 9, 1964, p. 1–9.

DAYHOFF, M.O.; SCHWARTZ, R.M.; ORCUTT, B.C. **A model of evolutionary change in proteins**. In Dayhoff MO. (Ed.), Atlas of Protein Sequence and Structure, vol. 5. National Biomedical Research Foundation: Washington DC, 1978, p. 345–352.

FELLER, W. **An Introduction to Probability Theory and Its Applications**. Vol. I, 2 nd ed. J. Willey and Sons, New York, 1957, 461p.

FELSENSTEIN, J.; CHURCHILL, G. A. **A hidden Markov model approach to variation among sites in rate of evolution**. Molecular Biology and Evolution. 1996, p. 13:93–104.

FRED, C.; CHRISTIANS, F. C.; SCAPOZZA, L.; CRAMERI, A.; FOLKERS, G.; STEMMER, W. P. C. **Directed evolution of thymidine kinase for AZT phosphorylation using DNA family shuffling**. Nature Biotechnology, vol. 17, 1999, p. 259–264.

FREIER, S. M.; KIERZEK, R.; JAEGER, J. A.; SUGIMOTO, N.; CARUTHERS, M. H.; NEILSON, T.; TURNER, D. H. **Improved free-energy parameters for predictions of RNA duplex stability**. Proc. Natl. Acad. Sci. USA, vol. 83, 1986, p. 9373–9377.

GAFVELIN, G.; PARMLEY, S.; NEIMERT-ANDERSSON, T.; BLANK, U.; ERIKSSON, T.J.; HAGE, M.V.; PUNNONEN, J. **Hypoallergens for allergen-specific immunotherapy by directed molecular evolution of mite group 2 allergens.** The Journal of Biological Chemistry, vol. 282(6), 2007, p. 3778–3787.

HE, L.; KIERZEK, R.; SANTALUCIA, J. Jr.; WALTER, A. E.; TUNER, D. H. **Nearest-Neighbor parameters for G·U mismatches**: $^{5'GU3'}_{3'UG5'}$ **is destabilizing in the contexts** $^{CGUG}_{CUGC}$, $^{UGUA}_{AUGU}$ **and** $^{AGUU}_{UUGA}$ **but s**$^{GGUC}_{CUGG}$**tabilizing in** . Biochemistry, vol. 30, 1991, p. 11124–11132.

HENIKOFF, S.; HENIKOFF, J.G. **Amino acid substitution matrices from protein blocks**. Proc Natl Acad Sci USA vol. 89, 1992, p.10915–10919.

HERMES, J. D., PAREKH, S. M., BLACKLOW, S. C., KOSTER, H., KNOWLES, J. R. **A reliable method for random mutagenesis**: the generation of mutant libraries using spiked oligodeoxyribonucleotide primers. Gene, vol. 84, 1989, p. 153–151.

HORTA, A.C.L.; ZANGIROLAMI, T.C.; NICOLETTI, M. C.; MONTERA, L.; CARMO, T.S.; GONÇALVES, V.M. **An empirical investigation of the use of a neural networks committee for identifying the streptococcus pneumoniae growth phases in batch cultivations**. IEA/AIE 2008, Lecture Notes in Computer Science, vol. 5027, 2008, p. 215–224.

HOWLEY, P.M.; ISRAEL, M.F.; LAW, M-F.; MARTIN, M.A. **A rapid method for detecting and mapping homology between heterologous DNAs**. Journal of Biological Chemistry, vol. 254, 1979, p. 4876–4883.

JOERN, J.M. **Engineering dioxygenases by laboratory evolution: A comparison of evolutionary search strategies**. Ph. D. Thesis, 233 pages, 2003.

JONES, D.D. **Triplet nucleotide removal at random positions in a target gene: the tolerance of TEM-1 b-lactamase to an amino acid deletion**. Nucleic Acids Research, vol. 33, 2005, e80.

JUKES, T. H.; CANTOR, C. R. **Evolution of protein molecules.**. In H. N. Munro (ed.), Mammalian Protein Metabolism. Academic Press, New York, 1969, p. 21–123.

KAMEL, A.; ABD-ELSALAM. **Bioinformatic tools and guideline for PCR primer design**. African Journal of Biotechnology, vol. 2, 2003, p. 91–95.

KEENAN, R.J.; SIEHL, D.L.; GORTON, R.; CASTLE, L.A. **DNA shuffling as a tool for protein crystallization.** PNAS, vol. 102(25), 2005, p. 8887–8892.

KIKUCHI, M. OHNISHI, K., HARAYAMA, S. **A Novel family shuffling methods for *in vitro* evolution enzymes**. Gene, vol. 236, 1999, p. 159–167.

KIKUCHI, M. OHNISHI, K., HARAYAMA, S. **An effective family shuffling method using single stranded DNA**. Gene, Amsterdan, vol. 243, 2000, p. 133–137.

KIMURA, M. **A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences**. Journal of Molecular Evolution vol. 16, 1980, p. 111–120.

KISHINO, H.; HASEGAWA, M. **Evaluation of the maximum likelihood estimate of the evolutionary tree topologies from DNA sequence data, and the branching order in Hominoidea**. Journal of Molecular Evolution vol. 29, 1989, p. 170–179.

KOCSOR, A. KERTESZ-FARKAS, A. KAJAN, L. PONGOR, S. **Application of compression-based distance measures to protein sequence classification: a methodological study**. Bioinformatics vol. 22 (4), 2006, p. 407–412.

KUIPERS, O. P.; BOOT, H.J.; VOS, W.M. **Improved site-directed mutagenesis method using PCR**. Nucleic Acids Research, vol. 19(16), 1991, p. 4558.

KUMAMARU, T.; SUENAGA, H.; MITSUOKA, M.; WATABANE, T.; FURUKAWA, K. **Enhanced degradation of polychlorinated biphenyls by directed evolution of biphenyl dioxygenase**. Nature biotechnology, vol. 16, 1998, p. 663–666.

LANDER, E. S. and WATERMAN, M. S. **Genomic mapping by fingerprinting random clones**: a mathematical analysis. Genomics, vol. 2(3), 1988, p. 231–239.

LANDT, O., GRUNERT, H. P., HAHN, U. **A general method for rapid site-directed mutagenesis**

using the polymerase chain reaction. Gene, vol. 96, 1990, p. 125–128.

LEUNG, D. W., CHEN, E. GOEDDE, D. U. **A method for random mutagenesis of a defined DNA segment using a modified polymerase chain reaction**. Technique, vol. 1, 1989, p. 11–15.

LI, M.; BADGER, JH.; CHEN, X.; KWONG, S.; KEARNEY, P.; ZHANG, H. **An information-based sequence distance and its application to whole mitochondrial genome phylogeny**. Bioinformatics vol. 17(2), 2001, p. 149–54.

LITTAUER, U. Z.; KORNBERG, A. **Reversible synthesis of polyribonucleotides with an enzyme from Escherichia coli**. Journal of Biological Chemistry, vol. 226, 1957, p. 1077–1092.

LOCHER, C.P.; PAIDHUNGAT, M.; WHALEN, R.G.; PUNNONEN, J. **DNA shuffling and screening strategies for improving vaccine efficacy**. DNA and Cell Biology, vol. 24(4), 2005, p. 256–263.

LODISH, H., BERK, A., ZIPURSKY, S. L., MATSUDAIRA, P., BALTIMORE, D.; DARNELL, J. **Molecular Cell Biology**. Freeman, 2001, 973p.

LUTZ, S., OSTERMEIER, M., MOORE, G. L., MARANAS, C. D., BENKOVIC, S. J. **Creating multiple-crossover DNA libraries independent of sequence identity**. Proc. Natl. Acad. Sci., vol. 98, 2001, p. 11248–11253.

LUTZ, S.; PATRICK, W.M. **Novel methods for directed evolution of enzymes: quality, not quantity**. Current Opinion in Biotechnology, vol. 15, 2004, p. 291–297.

MAHESHRI, N.; SCHAFFER, D. V. **Computational and experimental analysis of DNA Shuffling**. Proc. Natl. Acad. Sci., vol. 100, 2003, p. 3071–3076.

METZKER, M. L.; CASKEY, T. C. **Polymerase Chain Reaction** (PCR). Encyclopedia of Life Sciences. Nature Publishing Group, 2001, p. 1–9.

MONTERA, L.; HORTA, A.C.L.; ZANGIROLAMI, T.C.; NICOLETTI, M.C; CARMO, T.S.; GONÇALVES, V.M. **A heuristic search for optimal parameter values of three biokinetic growth models for describing batch cultivations of Streptococcus pneumoniae in bioreactors**. IEA/AIE 2008, Lecture Notes in Computer Science, vol. 5027, 2008a, p. 359–368.

MONTERA, L.; NICOLETTI, M.C. **The PCR primer design as a metaheuristic search process**. ICAISC 2008, Lecture Notes in Artificial Intelligence, vol. 5097, 2008, p. 963–973.

MONTERA, L.; NICOLETTI, M.C.; SILVA, F.H.; MOSCATO, P. **An effective mutation-based measure for evaluating the suitability of parental sequences to undergo DNA shuffling experiments**. IEEE Congress on Evolutionary Computation (CEC 2008) in World Congress on Computational Intelligence – Hong Kong, China 1-6 June 2008b, p. 765–772.

MONTERA, L.; NICOLETTI, M.C.; SILVA, F.H.; DELLAMANO, M. ISAS: **ISAS: An Interactive Software for Assisting Shuffling Process.** Neural Network World, vol 18, 2008c, p. 499–514.

MONTERA. L.; NICOLETTI, M.C; SILVA, F.H. **Computer assisted parental sequence analysis as a previous step to DNA shuffling process**. Conference on Evolutionary Computation (CEC 2006) in IEEE Congress on Computational Intelligence – Vancouver, Canada 16-21, July 2006, p. 8079–8086.

MOORE, G. L.; MARANAS, C. **Computational challenges in combinatorial library design for protein engineering**. AIChE Journal, vol. 50, 2004, p. 262–272.

MOORE, G. L.; MARANAS, C. D.; LUTZ, S.; BENKOVIC, S. L. **Predicting crossover generation**

in DNA shuffling. Proc. Natl. Acad. Sci., vol. 98, 2001, p. 3226–3231.

MÜLLER, K.M.; STEBEL, S.C.; KNALL, S.; ZIPF, GREGOR; BERNAUER, H.S.; ARNDT, K.M. **Nucleotide exchange and excision technology (NexT) DNA shuffling: a robust method for DNA fragmentation and directed evolution.** Nucleic Acids Research, vol. 33(13), 2005, e117.

MURAKAMI, H.; HOHSAKA, T.; SISIDO, M. **Random insertion and deletion of arbitrary number of bases for codon-based random mutation of DNAs**. Nature Biotechnology, 2002, vol. 20(1), p. 76–81.

NEEDLEMAN, S. B.; WUNSCH, C. D. **A general method applicable to the search for similarities in the amino acid sequence of two proteins**. Journal of Molecular Biology, vol. 48, 1970, p. 443–453.

NESS, J. E.; WELCH, M.; GIVER, L.; BUENO, M.; CHERRY, J. R.; BORCHERT, T. V.; STEMMER, W. P. C.; MINSHULL, J. **DNA shuffling of DNA subgenomic sequences of subtilisin**. Nature Biotechnology, vol. 17, 1999, p. 893–896.

NI, J.; TAKERAHA, M.; WATANABE, H. **Heterologous overexpression of a mutant termine Celullase gene in Escherichia coli by DNA shuffling of four orthologous parental cDNAs**. Biosci. Biotechnol. Biochem., vol. 69(9), 2005, p. 1711–1720.

NILL, K. **Glossary of Biotechology Terms**. 3 ª ed, CRC Press, 2002, 288p.

OLIVA, M. L. V.; CARMONA, A. K.; ANDRADE, S. S.; COTRIN. S. S.; COSTA, A. S.; SILVA, F. H. **Inhibitory selectivity of canecystatin: a recombinant cysteine peptidase inhibitor from sugarcane**. Biochemical and Biophysical Research Communications, vol. 320, 2004, p. 1082–1086.

OSTERMEIER, M.; SHIM, J. H., BENKOVIC, S. J. **A combinatorial pproach to hybrid enzymes independent of the DNA homology**. Nature Biotechnoly, vol. 17, 1999, p. 1205–1209.

PATNAIK, R.; LOUIE, S.; GAVRILOVIC, V.; PERRY, K.; STEMMER, W. P. C.; RYAN, C. M.; CARDAYRÉ, S. **Genome shuffling of Lactobacillus for improved acid tolerance**. Nature Biotechnology, vol. 20, 2002, p. 707–712.

PATRICK, W. M.; FIRTH, A. E.; BLACKBURN, J. M. **User-friendly algorithms for estimating completeness and diversity in randomized protein-encoding libraries**. Protein Engineering, vol. 16(6), 2003, p. 451–457.

RAILLARD, S.; KREBBER, A.; CHEN, Y.; NESS, J. E.; BERMUDEZ, E.; TRINIDAD, R.; FULLEM, R.; DAVIS, C.; WELCH, M.; SEFFERNICK, J.; WACKETT, L. P.; STEMMER, W. P. C.; MINSHULL, J. **Novel enzyme activities and functional plasticity revealed by recombining highly homologous enzymes**. Chemistry & Biology, vol 8, 2001, p. 891–898.

ROBERT, C.P.; CASELLA, G. **Monte Carlo Statistical Methods**. Springer – Verlag, 2004, 645p.

ROBERTS, R. J.; VINCZE, T.; POSFAI, J.; MACELIS D. **REBASE** – enzymes and genes for DNA restriction and modification. Nucleic Acids Research, vol. 35, 2007, p. D269–D270.

RODRIGUES, A.L.B.; ZHANG, X. **A simulated annealing and hill-climbing algorithm for the traveling tournament problem**. European Journal of Operational Research, vol. 174(3), 2006, p. 1459–1478.

ROSIC, N.N.; HUANG, W.; JOHNSTON, W.A.; DEVOSS, J.J.; GILLAM, E.M.J. **Extending the diversity of cytochrome P450 enzymes by DNA family shuffling**. Gene, vol. 395, 2007, p. 40–48.

RUBIN-PITEL, S.B.; ZHAO, H. **Recent advances in biocatalysis by directed enzyme evolution**.

Combinatorial Chemistry & High Throughput Screening, vol. 9, 2006, p. 247–257.

RUBINSTEIN, R. Y. **Simulation and Monte Carlo Method**. New York, John Wiley & Sons, 1981, 278p.

RUSSELL, J. B. **Química Geral.** 2ª ed, vol. 2, São Paulo: Makron Books, 2006 1421p.

RYCHLIK, W.; SPENCER, W.J.; RHOADS, R.E. **Optimization of the annealing temperature for DNA amplification in vitro**. Nucleic Acids Research, vol. 18, 1990, p. 6409–6412.

SAITOU, N., NEI, M. **The neighbor-joining method: a new method for reconstructing phylogenetic trees**. Mol Biol Evol., vol. 4(4), 1987, p. 406–25.

SANTALUCIA, J. Jr. **A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics**. Proc. Natl. Acad. Sci., vol 95, 1998, p. 1460–1465.

SANTALUCIA, J. Jr.; ALLAWI, H. T.; SENEVIRATNE, P. A. **Improved Nearest-Neighbor parameters for predicting DNA duplex stability.** Biochemistry, vol. 35, 1996, p. 3555–3562.

SETUBAL, C.; MEIDANIS, L. **Introduction to Computational Molecular Biology**. PWS Publishing Company, 1997, 320p.

SHAO, Z.; ZHAO, H.; GIVER, L.; ARNOLD, F. H. **Random-priming *in vitro* recombination: an effective tool for directed evolution**. Nucleic Acids Research, vol. 26, 1998, p. 681–683.

SILVA, M. C. M.; CRUZ, C. C. M.; TEIXEIRA, F. R.; DEL SARTO, R. P.; BEZERRA, I. C.; COUTINHO, M. V.; GROSSI de SÁ, M. F. G. **Insecticidal activity of shuffed alpha-amylase inhibitors**. In: XXXV Reunião Anual da SBBq, 2006, Águas de Lindóia. Resumos apresentados na XXXV Reunião Anual da SBBq.

SILVA, M. C. M.; DEL SARTO, R. P.; TEIXEIRA, F. R.; CRUZ, C. C. M.; COUTINHO, M. V.; GROSSI de SÁ, M. F. G. **Efficiency of shuffled alpha-amylase inhibitor in inhibit Anthonomus grandis alpha-amylase**. In: XXXVI Reuniáo Anual da Sociedade Brasileira de Bioquimica e Biologia Molecular, 2007, Salvador. Anais da XXXVI SBBq.

SILVA, M. C. M.; FIGUEIRA, E. L. Z.; SÁ, M. F. G. **A metodologia de DNA shuffling na produção de diversidade gênica**. Embrapa Recursos Genéticos e Biotecnologia, 2003.

SMITH, T. F.; WATERMAN, M. S. **Comparison of biosequences**. Adv. Appl. Math., vol. 2, 1981, p. 482–489.

SNUSTAD, D. P.; SIMMONS, M. J. **Fundamentos da Genética** 2 nd ed., 2001, Guanabara Koogan, 778p.

SPEE, J. H.; VOS, W. M.; KUIPERS, O. P. **Efficient random mutagenesis with adjustable mutation frequency by use of PCR dITP**. Nucleic Acids Research, vol. 21, 1993, p. 777–778.

STEMMER, W. P. C. **DNA shuffling by random fragmentation and reassembly**: *In vitro* recombination for molecular evolution. Proc. Natl. Acad. Sci., vol. 91, 1994b, p. 10747–10751.

STEMMER, W. P. C. **Rapid evolution of a protein *in vitro* by DNA shuffling**. Nature, vol. 370, 1994a, p. 389–391.

STEMMER, W.P.; HOLLAND, B. **Survival of the fittest molecule**. Am. Sci. 91, 2003, p.526–533.

STEVENSON, J. D.; BENKOVIC, S. J. **Combinatorial approches to engineering hybrid enzymes**.

Journal of the Chemical Society, Perkin Trans., vol. 2, 2002, p. 1483–1493.

SUGIMOTO, N.; NAKANO, S.; YONEYAMA, M.; HONDA, K. **Improved thermodynamic parameters and helix initiation factor to predict stability of DNA duplexes**. Nucleic Acids Research, vol. 24, 1996, p. 4501-4505.

SUN, F. **Modeling DNA Shuffling**. Journal of Computational. Biology, vol. 6, 1999, p. 77–90.

TAHERI, J.; ZOMAYA, A.Y. **A Simulated Annealing approach for mobile location management.** Computer Communications, vol. 30 (4), 2007, p. 714–730.

URBAN, A.; NEUKIRCHEN, S.; JAEGER, K. E. **A Rapid and Efficient Method for Site-Directed Mutagenesis Using One-Step Overlap Extension PCR**. Nucleic Acids Research, vol. 25, 1997, p. 2227–2228.

VINGA, S.; ALMEIDA, J. **Alignment-free sequence comparison** - a review. Bioinformatics vol. 19(4), 2003, p. 513–523.

VOLKOV, A A.; SHAO, Z.; ARNOLD, F. H. **Recombination and chimeragenesis by *in vitro* heteroduplex formation and in vivo repair**. Nucleic Acids Research, vol. 27, 1999, e18.

VOLKOV, A. A.; ARNOLD, F. H. **Methods for *in Vitro* Recombination and Random Chimeragenesis**. Methods in Enzimology, vol. 328, 2000, p. 447–456.

WALLACE, R.B.; SHAFFER, J.; MURPHY, R.F.; BONNER, J.; HIROSE, T.; ITAKURA, K. **Hybridization of synthetic oligodeoxyribonucleotides to ϕX 174 DNA: the effect of single base pair mismatch**. Nucleic Acids Research, vol. 6, 1979, p. 3543–3557.

WANG, Y.; Li, Y.; PEI, X.; Yu, L.; FENG, Y. **Genome-shuffling improved acid tolerance and l-lactic acid volumetric productivity in Lactobacillus rhamnosus.** Journal of Biotechnology, vol. 129, 2007, p. 510–515.

WATSON, J.; GILMAN, M.; WITKOWSKI, J.; ZOLLER, M. **Recombinant DNA**. Scientific American Books, 2nd ed., 1992, 626p.

WU, W. B.; HUANG, T. F. **Activation of MMP-2, cleavage of matrix proteins, and adherens junctions during a snake venom metalloproteinase-induced endothelial cell apoptosis.** Experimental Cell Research. vol. 288, 2003, p. 143–157.

ZHANG, Y.X.; KIM, P.; VINCI, V.A.;, POWELL. K.; STEMMER, W.P.; DEL CARDAYRE, S.B. **Genome shuffling leads to rapid phenotypic improvement in bacteria**. Nature, vol. 415, 2002, p. 644–646.

ZHAO, H.; ARNOLD, F. H. **Optimization of DNA shuffling for high fidelity recombination**. Nucleic Acid Research, vol. 25, 1997, p. 1307–1308.

ZHAO, H.; GIVER, L.; SHAO, Z.; AFFHOLTER, A.; ARNOLD, F. H. **Molecular evolution by staggered extension process (StEP) *in vitro* recombination**. Nature Biotechnology, vol. 16, 1998, p. 258–261.

ZHENG, L.; BAUMANN, U.; REYMOND, J-L. **An efficient one-step site-directed and site-saturation mutagenesis protocol**. Nucleic Acid Research, vol. 32 e115, 2004.

ZHOU, Y.; ZHANG, X.; EBRIGHT R. H. **Random mutagenesis of gene-sized DNA molecules by use of PCR with Taq DNA polymerase**. Nucleic Acids Research, vol. 19, 1991, p. 6052.

GLOSSÁRIO

As definições contidas neste glossário foram, na sua grande maioria, compiladas de Nill (2002), Campbell et al. (1999) e Brown (1999).

Ácido nucléico – Polímero composto por monômeros de nucleotídeos. Pode ser de dois tipos: ácido desoxirribonucleico (DNA) e ácido ribonucleico (RNA).

Algoritmo – Seqüência de instruções que descrevem a solução de um problema com vistas à codificação em uma linguagem de programação.

Alinhamento entre seqüências biológicas – Justaposição de seqüências com o objetivo de evidenciar regiões de igualdade entre elas.

Aminoácido – Molécula orgânica que possui um grupo carboxil e um grupo amino. São os monômeros formadores das proteínas.

Amplificação – Processo de produção de cópias adicionais de uma dada seqüência cromossômica.

Apoptose – Morte celular programada.

Biblioteca de recombinantes – No contexto de DNA *shuffling*: representa o conjunto de fragmentos de DNA obtidos ao final do processo; os quais já foram clonados.

Célula hospedeira – Célula cujo metabolismo é utilizado para crescimento e reprodução por um vírus. É também uma célula na qual um vetor de clonagem é inserido para que o vetor, utilizando-se do mecanismo de reprodução da célula hospedeira, também seja reproduzido.

Clivagem – Processo por meio do qual são realizados cortes ao longo de uma molécula de DNA/RNA.

Clonagem – Processo de produção de diversas cópias de uma determinada molécula.

Clone – Grupo de indivíduos ou células geneticamente idênticos.

Clones ativos (de uma biblioteca de DNA *shuffling*) – Clones resultantes do processo de DNA

shuffling cuja função foi preservada.

Código genético – Regras que estabelecem a relação entre triplas de nucleotídeos e aminoácidos durante o processo de síntese de proteínas.

Códon – Seqüência de três nucleotídeos do DNA ou mRNA que especifica um determinado aminoácido. É a unidade básica do Código Genético.

Colônia – Grupo de microoganismos resultantes do crescimento de um mesmo organismo original.

Desnaturação (do DNA) – Separação, em duas fitas simples, da fita dupla que forma a estrutura de dupla hélice da molécula.

DNA ligase – Enzima que sintetiza a ligação fosfodiéster entre a extremidade 3' de um fragmento de DNA e a extremidade 5' de outro fragmento de DNA.

DNA molde (ou *template*) – Cadeia de nucleotídeos que serve como base para a criação de uma nova cadeia de nucleotídeos durante o processo de síntese de DNA (ou RNA).

DNA polimerase – Enzima que catalisa o alongamento de uma molécula de DNA durante o processo de replicação por meio da adição de nucleotídeos à extremidade 3' da molécula sendo alongada.

DNA recombinante – Molécula de DNA produzida *in vitro* resultante da união de dois ou mais fragmentos de DNA de origens distintas.

DNase I – Enzima que degenera o DNA. Produz cortes ao longo da molécula de DNA.

Eletroforese – Técnica de separação (purificação) de moléculas com base no movimento diferenciado apresentado por moléculas com diferentes cargas elétricas quando sujeitas a um campo elétrico.

Eletroforese em Gel de agarose – Processo de eletroforese executado em gel de agarose e utilizado para separação de moléculas de DNA segundo o seu tamanho.

Endonucleases – Classe de enzimas capazes de realizar a clivagem (quebra) de ligações fosfodiésteres presentes nas moléculas de DNA e RNA.

Endonucleases de restrição – Enzimas da família das endonucleases. Realizam a clivagem de

ligações fosfodiésteres entre nucleotídeos específicos dentro de uma cadeia de DNA ou RNA. Por realizarem a clivagem de ligações fosfodiésteres apenas entre nucleotídeos especícifos, estas enzimas são também conhecidas como enzimas de restrição sítio-específicas.

Evolução – Processo realizado em laboratório (*in vitro*) que imita o processo de evolução Darwiniana através de diferentes técnicas com o objetivo de criar variantes de proteínas ou ácidos nucléicos que apresentem funcionalidades novas ou melhoradas.

Evolução Molecular – Mudanças graduais que ocorrem naturalmente no genoma ao longo do tempo devido à ocorrência de mutações e recombinações.

Evolução Molecular Direta – Técnicas experimentais que, de alguma forma, imitam o processo de Evolução Molecular com o objetivo de obter, por exemplo, genes com funcionalidades melhoradas ou novas.

Expressão de um gene – Conversão da informação genética contida em um gene na proteína correspondente.

Filogenia – Esquema de classificação que indica a história evolucionária entre organismos.

Fragmento (de DNA) híbrido / fragmento (de DNA) recombinante – Fragmento de DNA resultante da união de dois ou mais fragmentos de DNA de origens distintas.

Gene – Fragmento de DNA que contém informações biológicas que codifica para um RNA ou uma proteína. É a base de transmissão das características hereditárias entre as gerações.

Genoma – O conjunto de todo o material genético de um organismo.

Heteroduplex – Molécula de DNA na qual cada uma das fitas é originária de indivíduos diferentes e que, possivelmente, possuem pares de bases não complementares entre si.

Homoduplex – Molécula de DNA em que ambas as fitas são originárias de um mesmo indivíduo.

Homologia – Seqüência de aminoácidos em duas ou mais proteínas que são idênticas entre si. Em se tratando de ácidos nucléicos, homologia se refere a cadeias complementares que podem se parear umas às outras.

In silico **(biologia)** – Uso de computadores para simular, processar e analisar um experimento

biológico.

In vivo **−** Derivada do Latin "em vivo". Diz respeito a situações como, por exemplo, o teste de novas drogas em organismos vivos (testes *in vivo*).

Mapa de restrição − Representação das localidades dos sítios de restrição encontrados em uma molécula (ex. plasmídeo).

Match **−** Nome dado à coluna de um alinhamento entre seqüências de DNA ou RNA onde estão alinhadas duas bases iguais.

Matriz de substituição − Matriz que armazena, para todos os possíveis pares de aminoácidos do tipo (A, B), o custo de se substituir o aminoácido A pelo aminoácido B em um dado alinhamento.

Método de *screening* (ou seleção) − Método utilizado para selecionar elementos de um conjunto, com base nas características dos elementos buscados.

Mismatch **−** Nome dado à coluna de um alinhamento entre seqüências de DNA ou RNA onde estão alinhadas duas bases diferentes.

Mutação − Alteração na seqüência de nucleotídeo de uma molécula de DNA.

Mutagênese − Ato de criar mutações.

Nucleotídeo − Unidade básica formadora dos ácidos nucléicos, constituída por uma molécula de açúcar, uma base nitrogenada e um grupo fosfato.

Oligonucleotídeos − Pequenas cadeias de nucleotídeos sintéticos construída pela união de nucleotídeos específicos.

Patógenos − Refere-se a vírus, bactérias, protozoários e outros microorganismos que causam doenças infecciosas ao invadir o corpo de um organismo (animal ou planta).

PCR − Acrônimo para *Polymerase Chain Reaction* (Reação de Polimerase em Cadeia). Reações de PCR permitem, entre outros, que inúmeras cópias de uma mesma molécula de DNA sejam criadas.

Plasmídeo − Molécula de DNA circular auto-replicante e independente do cromossomo geralmente encontrada em bactérias. Muitos plasmídeos possuem gene(s) de resistência a antibióticos.

Primer **−** Oligonucleotídeo complementar ao trecho inicial de uma fita simples de DNA (ou RNA)

que se deseja amplificar. Provê o ponto inicial para a síntese da nova fita de nucleotídeos.

Progênie – Geração, descendência ou prole.

Proteínas – Polímero tridimensional formado por um conjunto de 20 diferentes monômeros chamados aminoácidos, unidos por meio de ligações peptídicas.

Purificação de DNA – Processo pelo qual fragmentos de DNA são separados ou isolados.

Recombinação – União de genes ou fragmentos de DNA em novos rearranjos. A recombinação pode ocorrer *in vivo* bem como *in vitro*.

Replicação (de DNA) – Reprodução de uma molécula de DNA dentro da célula.

RNA Polimerase – Enzima que catalisa a reação de síntese de RNA.

Seqüência de DNA recombinante – Seqüência de DNA formada por fragmentos de DNAs provenientes de duas ou mais espécies diferentes.

Seqüência palíndromo – Uma seqüência é dita ser palíndromo se tem a mesma leitura de trás para frente e de frente para trás.

Seqüência quimérica – Ver **Seqüência de DNA recombinante.**

Seqüências homólogas – O termo seqüências homólogas ou homologia é utilizado para referenciar genes que compartilham um ancestral evolutivo em comum que pode ser revelado pela identidade de suas seqüências.

Similaridade (entre seqüências) – Igualdade entre as bases (ou aminoácidos) que compõem duas ou mais seqüências.

Sítios de restrição – Seqüência de nucleotídeos reconhecida por enzimas do tipo endonucleases de restrição.

Técnicas de mutagênese – Técnicas capazes de produzir mutações em moléculas de DNA.

Tradução – Processo de síntese de uma cadeia de aminoácidos determinada pelas informações contidas na seqüência de nucleotídeos de um mRNA (RNA mensageiro) de acordo com as regras estabelecidas pelo Código Genético.

Transcrição – Processo pelo qual as informações genéticas contidas em uma molécula de DNA e correspondentes aos genes são copiadas para as moléculas de mRNA (RNA mensageiro). Esse processo poder ser visto como a reescrita de informações do DNA para o RNA.

Transformação de bactéria – Processo pelo qual uma bactéria recebe DNA originário de outro organismo.

Vetor de clonagem – Molécula de DNA capaz de se auto-replicar dentro de uma célula hospedeira e, desta forma, pode ser utilizada para clonar outros fragmentos de DNA.

Vetores de expressão – Organismos, geralmente plasmídeos, utilizados para introduzir e expressar (na proteína correspondente) um determinado gene dentro de uma célula hospedeira.

APÊNDICE A – DRIVeR

A.1. O software DRIVeR

O modelo descrito por Patrick e colaboradores, implementado por um software escrito na linguagem Fortran 77, foi obtido no endereço www.bio.cam.ac.uk/~blackburn/stats.html. Neste apêndice é descrito como o DRIVeR, em sua forma original, deve ser executado, bem como quais dados de entrada são necessários à sua execução e as saídas produzidas.

As informações necessárias à execução do DRIVeR são: tamanho do alinhamento entre as seqüências parentais, número real de cruzamentos, tamanho da biblioteca de DNA *shuffling* construída e número e localização das mutações existentes entre os parentais (identificadas por meio da construção do alinhamento entre eles). Tais informações devem estar armazenadas em um arquivo chamado "setup.dat", que deve ter o seguinte formato:

N → tamanho do alinhamento entre os parentais

λ → número real de cruzamentos

L → tamanho da biblioteca

M → número de mutações entre os parentais

M_1 → posição no alinhamento da primeira mutação entre os dois parentais

M_2 → posição no alinhamento da segunda mutação entre os dois parentais

.

.

.

M_M → posição no alinhamento da última mutação entre os dois parentais

A implementação do DRIVeR consta de um único arquivo chamado "rdriver.f". Considerando o compilador DIGITAL Visual Fortran 6.0, para que o arquivo rdriver.f seja compilado e executado, ele deve ser aberto, em seguida o botão *Build* → *Compile rdriver.f* pressionado, seguido do botão *Execute* → *rdriver.exe*. A execução do rdriver.exe resulta na exibição das seguintes estimativas na tela de execução:

– Número de seqüências variantes distintas esperadas na biblioteca de DNA *shuffling* construída (*expected number of distinct sequences*); e

– Número médio de cruzamentos observados por seqüências (*mean number of observable crossovers per sequence*).

Além dos resultados exibidos na tela, o software gera um arquivo chamado "driver.dat" para armazenar a probabilidade de ocorrência de cada uma das $2^M/2$ variantes distintas. Cada uma das variantes está armazenada no arquivo segundo a sua representação binária, como apresentado no Capítulo 3, Seção 3.2.4.

A.2. Pseudo-código do algoritmo que viabiliza o DRIVeR

O código do programa DRIVeR, originalmente em Fortran, foi transcrito para a linguagem C. Essa transcrição possibilitou um melhor entendimento do modelo bem como auxiliou na sua descrição, como apresentado na seção 3.2. O pseuco-código gerado a partir da transcrição realizada é apresentado na Figura A.1.

procedure DRIVeR(T, λ^{true}, L, n_mut, mutação[ ])

{**Entrada:** T → tamanho do alinhamento entre os parentais
λ^{true} → número real de cruzamentos verificado na amostra de DNA *shuffling*
L → tamanho da biblioteca de DNA *shuffling* construída
n_mut → número de mutações entre os parentais
mutação[ ] → lista com a localização no alinhamento das n_mut mutações existentes entre os parentais
}
{**Saída:** C → número esperado de variantes distintas na biblioteca de DNA *shuffling*
λ^{obs} → número médio de cruzamentos verificado nas seqüências da biblioteca

begin
for i = 1, 2, 3, ..., n_mut
begin
{número de pontos onde cruzamentos podem ocorrem entre as mutações consecutivas i e i+1}
pontos_cruzamento[i] = mutação[i +1] – mutação[i]
{calcule número esperado de cruzamentos neste intervalo}

$$\lambda[i] = \frac{\text{pontos\_cruzamento}[i] * \lambda^{\text{true}}}{(T - n\_mut) - 1}$$

Figura A.1. Pseudo-código do DRIVeR (continua).

```
      {calcule a probabilidade de ocorrer um número par de cruzamentos neste intervalo}
      p_par_cruzmantos[i] = e – λ[i] * cosh(λ[i])
      {calcule a probabilidade de ocorrer um número ímpar de cruzamentos neste intervalo}
      p_impar_cruzamentos[i] = e – λ[i] * sinh(λ[i])
  end
λ^obs = 0
for i = 1, 2, 3, ..., 2^n_mut do {calcule a probabilidade de cada uma das variantes distintas ocorrer}
  begin
      soma = i
      p_variante[i] = 1
      num_cruzamentos_obs = 0
      for j = 1, 2, 3, ..., n_mut do
        begin
            {determine se esta posição da variante corresponde ao bit 0 ou 1}
            variante[j] = (soma – 2) * (soma / 2)
            if variante[j] == zero then
              begin
                  p_variante[i] = p_variante[i] * p_par_cruzamentos[j]
              end
            else
              begin
                  p_variante[i] = p_variante[i] * p_impar_cruzamentos[j]
                  {como este bit é 1, um cruzamento ocorreu}
                  num_cruzamentos_obs = num_cruzamentos_obs + 1
              end
            soma = (soma – variante[i]) / 2
        end
        {como existem duas variantes distintas que são mapeadas pela mesma seqüência binária, a
        probabilidade calculada deve ser dividida por 2, para representar
        apenas a probabilidade de uma delas}
        write(variante, p_variante[i] * 0.5) {armazena em arquivo a variante (representação binária) e
                                              sua probabilidade de ocorrência}
        {calcula o valor de λ^obs com base no número de cruzamentos
```

Figura A.1. Pseudo-código do DRIVeR (continução).

```
        existentes em cada variante e na probabilidade da mesma ocorrer}
        λ^obs = λ^obs + p_variante[i] * num_cruzamentos_obs
    end
  write(λ^obs) {escreve o número observado de cruzamentos}
  {Calcule o número esperado de variantes distintas na biblioteca}
  for j = 1, 2, 3, ..., 2^num_mut do
   begin
      x = 1 – (1 – (p_variante[i] * 0.5) )L
      C = C + x
   end
  write(C) {escreve o número esperado de variantes na biblioteca}
end
```

Figura A.1. Pseudo-código do DRIVeR (conclusão).

APÊNDICE B – ShuffIt

B.1. O software ShuffIt

O modelo ShuffIt descrito por Maheshri e Schaffer (2003) foi implementado por um software escrito na linguagem C utilizando o aplicativo DEV-C IDE. O *download* do software foi feito no endereço http://www.cchem.berkeley.edu/~schaffer/*shuffling*/. Alguns detalhes do software são apresentados nesta subseção, a fim de facilitar sua utilização. O software é composto por um conjunto de arquivos contendo a implementação das funções que contemplam o modelo proposto. A Tabela B.1 apresenta e descreve cada um destes arquivos.

Tabela B.1. Descrição dos arquivos/funções que implementam o ShuffIt.

Arquivo	Descrição
main.c	Função principal. Realiza entrada de dados e saída de resultados. Dispara a execução da reação de DNA *shuffling*.
DNADigest.c	Realiza a fragmentação das seqüências parentais (fita simples) de maneira consistente com a fragmentação produzida pela DNase I. A fragmentação é modelada como um processo de Poisson.
io.c	Implementa funções relacionadas à escrita dos resultados em arquivo de saída.
linkedlist.c	Implementa uma serie de funções para manipulação de uma estrutura de dados específica denominada "lista ligada", necessária para que os fragmentos de DNA sejam armazenados.
ranrotB.c	Função que implementa a geração de números randômicos. Os números randômicos são utilizados quando da fragmentação das seqüências parentais.
stringfncs.c	Implementa funções de manipulação de strings, ou seja, funções de manipulação de seqüências.
ungappedalignment.c	Dados dois fragmentos do conjunto ssDNA, essa função calcula a probabilidade de todos os pareamentos possíveis entre estes dois fragmentos com base na energia livre resultante de cada um dos eventos de pareamentos. Com base nas probabilidades, uma configuração particular de pareamento é escolhida, estendida e retornada.

São necessários os seguintes dados como entrada para a execução do ShuffIt:

- Seqüências de DNA correspondentes aos parentais juntamente com as suas concentrações na reação. Valores estes armazenados na variável *initial* ;
- Tamanho mínimo dos fragmentos submetidos ao *shuffling*, armazenado na variável MFS;
- Tamanho máximo dos fragmentos submetidos ao *shuffling*, armazenado na variável XFS;
- Tamanho médio dos fragmentos submetidos ao *shuffling*, armazenado na variável AFS;
- Densidade inicial dos parentais (antes da fragmentação por DNase I), armazenada na rho;
- Concentração de Mg^{2++} na reação, armazenado na variável Cmg;
- Concentração dos nucleotídeos livres disponíveis para as extensões realizadas pela polimerase, armazenada na variável CdNTP;
- Concentração de sal na reação, armazenada na variável Cmono;
- Temperatura de pareamento, armazenada na variável T.

Enquanto o DRIVeR requer que os parâmetros de entrada necessários à sua execução estejam armazenados em um arquivo de entrada específico, e o eShuffle requer que o usuário digite os dados de entrada no momento da sua execução (ver Apêndice C), ShuffIt o faz diferente. A informação quanto aos valores dos parâmetros de entrada do modelo deve ser feita nos arquivos que o implementa. Para se descobrir quais e onde se localizavam todas as variáveis das quais a reação de DNA *shuffling* é dependente, todos os sete arquivos que compõem o software ShuffIt (ver Tabela B.1) foram verificados. Os principais parâmetros de entrada que devem ser modificados pelo usuário para que a simulação pelo ShuffIt represente uma reação particular de DNA *shuffling* estão listadas na Tabela B.2. Somente após a correta atribuição de valores às variáveis é que o ShuffIt pode ser compilado e executado.

Para usuários utilizando o compilador DEV C/C++, a execução do ShuffIt pode ser feita apenas abrindo-se o projeto "Shuffling1.0". Esse projeto traz todos os arquivos de extensão ".c" reunidos e prontos para execução. Desta forma, basta que o botão *Execute* → *Run* seja acionado. Caso o usuário deseje executar o ShuffIt a partir de um outro compilador, os arquivos devem ser abertos um a um, compilados, "lincados" e em seguida executados, num processo semelhante ao descrito para a execução do eShuffle, no Apêndice C, Seção C.1.

Tabela B.2. Localização das variáveis que devem ser atualizadas segundo a reação de *shuffling* a ser simulada pelo ShuffIt.

Arquivo	Parâmetros a serem modificados segundo a reação de *shuffling* a ser simulada
main.c	MFS → tamanho mínimo dos fragmentos submetidos ao *shuffling*.
	XFS → tamanho máximo dos fragmentos submetidos ao *shuffling*.
	AFS → tamanho médio dos fragmentos submetidos ao *shuffling*.
	rho → densidade (g/L) dos parentais (DNA).
	Cmg → concentração de Mg^{2++}.
	CdNTP → concentração de nucleotídeos.
	Cmono → concentração de cátions monovalente.
	initial → sequências parentais (orientação 5' → 3').
ungappedalignment.c	T → temperatura de pareamento dada em Kelvin. O Valor padrão utilizado é de 308,15 ºK, que corresponde a 35ºC.

O número de ciclos de PCR executados pelo ShuffIt não é um parâmetros fixo, nem mesmo um parâmetro cujo valor é determinado automaticamente pelo software com base nos dados de entrada (tamanho das seqüências parentais e tamanho dos fragmentos, por exemplo). Desta forma, o usuário deve ainda modificar, no código do programa main.c, a linha que controla a execução do número de ciclos de PCR a serem simulados antes de realizar os procedimentos necessário à execução do ShufflIt.

Diversos testes foram realizados na tentativa de utilização do ShuffIt, todos sem sucesso. Dadas as seqüências parentais PurN e GarT (ver Tabela 5.5 do Capítulo 5) de tamanhos 636 pb e 609 pb, respectivamente e, após modificados todos os parâmetros de entrada necessários para que a simulação representasse a reação de DNA *shuffling* desejada (AFS = 45, T = 55ºC), iniciaram-se as tentativas no sentido de escolher um valor apropriado para o número de ciclos de PCR a serem executados pelo simulador. A simulação considerando 10 ciclos de PCR não produziu resultados, uma vez que os arquivos de saída ficaram vazios. Em seguida, tentou-se a execução considerando 14 ciclos de PCR (a utilização deste valor foi descrito em (MAHESHRI; SCHAFFER, 2003)). Após 5 horas e meia de execução, uma mensagem de erro indicando acesso indevido de uma posição de memória foi mostrada e a execução do programa finalizada. Uma terceira tentativa, agora considerando 30 ciclos de PCR, resultou na mesma mensagem de erro

após apenas 25 minutos de execução. Novamente, foi realizada a tentativa de simulação anterior (30 ciclos de PCR) e, agora, a mensangem de erro ocorreu após 2 horas e 20 minutos de execução. Os erros de execução ocorreram, provavelmente, devido à escolha inadequada do número de ciclos de PCR. O contato com um dos autores − Narendra − foi feito e o problema exposto (comunicação pessoal)[36]. Narendra afirma que o número de ciclos de PCR deve ser escolhido com base no tamanho das seqüências parentais e no tamanho dos fragmentos a fim de evitar que seqüências muito longas sejam remontadas. No caso da tentativa de simulação entre os parentais PurN e GarT, cuja maior delas (PurN) tem tamanho 636 pb e fragmentos de tamanho médio 45 pb foram utilizados, esperava-se que 14 ciclos de PCR fosse um número razoável que produziria seqüências remontadas de tamanho em torno dos tamanhos dos parentais[37].

Por não ter sido possível a utilização do software nas diversas tentativas realizadas, e por considerar a determinação do número adequado de ciclos de PCR uma tarefa dispendiosa, a disponibilização do ShuffIt no módulo de *Pre Shuffling* (Capítulo 4, Seção 4.2.2) não foi realizada. Pelo mesmo motivo, o software não teve seus resultados apresentados no estudo comparativo entre as estimativas dos demais softwares DRIVeR, eShuffle e SimAffling, feito no Capítulo 5. O pseudo-código no ShuffIt também não foi descrito.

[36] MAHESHRI, N. Questions about ShuffIt. Mensagem enviada por montera@dc.ufscar.br em 25 de Agosto de 2008.

[37] Segundo instruções do autor, acredita-se que o número de ciclos de PCR deva ser algo em torno do valor resultante de |tamanho_parental| / L.

APÊNDICE C – eShuffle

C.1. O software eShuffle

O modelo descrito por Moore e colaboradores foi implementado na linguagem Fortran. Como não há uma publicação descrevendo os detalhes do software eShuffle, e sim apenas descrevendo o modelo proposto Moore et al. (2000), alguns detalhes do software são apresentados neste apêndice, a fim de facilitar sua utilização. O software não se encontra disponível para *download* de forma que o código foi obtido por meio de solicitação a um dos autores. O software é composto por um conjunto de arquivos contendo a implementação das funções que contemplam o modelo proposto. A Tabela C.1 apresenta e descreve cada um desses arquivos.

Tabela C.1 – Descrição dos arquivos/funções que implementam o eShuffle.

Arquivo	Descrição
main.f	Função principal. Calcula a fração das seqüências *full-lenght* contendo 0, 1, 2,..., 10 cruzamentos nas seqüências resultantes do DNA *shuffling* entre as seqüências parentais.
global.f	Declaração das variáveis globais.
parin.f	Realiza a leitura dos parâmetros de entrada necessários à execução do eShuffle. Os seguintes parâmetros, na mesma ordem em que são apresentados, devem ser informados pelo usuário no momento da execução do eShuffle: L → tamanho dos fragmentos submetidos ao *shuffling*; T → temperatura de pareamento utilizada no experimento; B → tamanho do alinhamento entre os parentais; N → número de seqüências parentais; e File → nome do arquivo que armazena as seqüências parentais.
seqinput.f	Lê de um arquivo de entrada, cujo nome é fornecido pelo usuário (File), as seqüências parentais bem como os valores para os parâmetros do modelo NN que devem estar armazenados em um arquivo de nome "Gvalues".
match.f	Calcula a concentração relativa dos fragmentos na reação.
thermo.f	Calcula o valor de ΔG para uma determinada configuração de pareamento.

Antes de preparar o arquivo de entrada contendo as seqüências parentais, um alinhamento entre elas deve ser construído. Considerando duas seqüências parentais e o respectivo alinhamento, o arquivo de entrada deve obedecer ao seguinte formato: a primeira linha deve armazenar os primeiros 60 nucleotídeos correspondentes ao parental 1 no alinhamento; a segunda linha deve conter os primeiros 60 nucleotídeos correspondentes ao parental 2 no alinhamento; a terceira linha deve armazenar os próximos 60 nucleotídeos correspondentes ao parental 1 no alinhamento, ou seja, os nucleotídeos que vão da posição 61 até a posição 120; e assim por diante, até que as seqüências parentais estejam por completo armazenadas no arquivo.

Os valores para os parâmetros do modelo NN utilizados pelo eShuffle foram compilados de SantaLucia (1998). O modelo considera, além dos dezesseis NN pares onde apenas *matches* ocorreram (ALLAWI; SANTALUCIA, 1997), NN pares onde um único *mismatch* ocorre. A Tabela C.2, adaptada de Moore et al. (2001), descreve todos os pares NN considerados na implementação do modelo. Os pares NN e seus respectivos valores de ΔH e ΔS estão armazenados no arquivo Gvalues, fornecido juntamente com os outros arquivos que implementam o modelo.

Como um executável não é fornecido juntamente com os códigos fontes que implementam o modelo, estes precisaram ser compilados e executados a partir de um compilador Fortran para que o executável seja gerado. Considerando o compilador DIGITAL Visual Fortran 6.0, os seguintes passos devem ser seguidos a fim de executar o eShuffle:

- Abrir e compilar o arquivo main.f (botão *Build* → *Compile main.f*). EsSa compilação irá criar um 'projeto', no qual outros arquivos poderão ser incluídos;
- Abrir e compilar um a um os seguintes arquivos: global.f, parin.f, seqinput.f, match.f e thermo.f. Ao compilar cada um destes arquivos, os mesmos serão inseridos no projeto criado no momento que o arquivo main.f foi compilado;
- Por fim, o arquivo executável main.exe foi criado. Para executá-lo, basta clicar no botão *Build* → *Execute main.exe*.

O eShuffle não apresenta nenhuma interface com o usuário, ou seja, ao executar o eShuffle, nenhuma solicitação para entrada dos parâmetros é feita. É esperado, contudo, que o usuário informe os parâmetros de entrada, como mostrado na Tabela C.1 (descrição da função parin.f). Como saída, eShuffle apresenta na tela de execução a fração das seqüências remontadas que contém 0, 1, 2,..., 10 cruzamentos, bem como o número médio de cruzamentos por seqüência remontada para o experimento de *shuffling* simulado.

Tabela C.2 – Pares de bases e respectivos valores de ΔH e ΔS utilizados pelo eShuffle.

Matches			*Mismatches* simples											
Par	ΔH	ΔS	Par	ΔH	ΔS	Par	ΔH	ΔS	Par	ΔH	ΔS	Par	ΔH	ΔS
AA/TT TT/AA	-7,9	-22,2	AA/TA	1,2	1,7	GG/CG	-6,0	-15,8	GA/CG	-0,6	-1,0	TA/AC	3,4	8,0
AT/TA	-7,2	-20,4	CA/GA	-0,9	-4,2	TG/AG	1,6	3,6	GG/CA	0,5	3,2	TC/AA	7,6	20,2
TA/AT	-7,2	-21,3	GA/CA	-2,9	-9,8	AT/TT	-2,7	-10,8	TA/AG	0,7	0,7	AC/TT	0,7	0,2
CA/GT TG/AC	-8,5	-22,7	TA/AA	4,7	12,9	CT/GT	-5,0	-15,8	TG/AA	3,0	7,4	AT/TC	-1,2	-6,2
GT/CA AC/TG	-8,4	-22,4	AC/TC	0	-4,4	GT/CT	-2,2	-8,4	AA/TC	2,3	4,6	CC/GT	-0,8	-4,5
CT/GA AG/TC	-7,8	-21,0	CC/GC	-1,5	-7,2	TT/AT	0,2	-1,5	AC/TA	5,3	14,6	CT/GC	-1,5	-6,1
GA/CT TC/AG	-8,2	-22,2	GC/CC	3,6	8,9	AA/TG	-0,6	-2,3	CA/GC	1,9	3,7	GC/CT	2,3	5,4
CG/GC	-10,6	-27,2	TC/AC	6,1	16,4	AG/TA	-0,7	-2,3	CC/GA	0,6	-0,6	GT/CC	5,2	13,5
GC/CG	-9,8	-24,4	AG/TG	-3,1	-9,5	CA/GG	-0,7	-2,3	GA/CC	5,2	14,2	TC/AT	1,2	0,7
GG/CC CC/GG	-8,0	-19,9	CG/GG	-4,9	-15,3	GG/GA	-4,0	-13,2	GC/CA	-0,7	-3,8	TT/AC	1,0	0,7
			AG/TT	1,0	0,9	AT/TG	-2,5	-8,3	CG/GT	-4,1	-11,7	GG/CT	3,3	10,4
			CT/GG	-2,8	-8,0	GT/CG	-4,4	-12,3	TG/AT	-0,1	-1,7	TT/AG	-1,3	-5,3

Valor médio para *mismatch* duplo: ΔH = 2,8 k.cal/mol, ΔS = 6,5 cal/mol.K
Valor médio para o custo iniciação: ΔH = 2,4 k.cal/mol, ΔS = 1,3 cal/mol.K

C.2. Pseudo-código do algoritmo eShuffle

O código do programa eShuffle, originalmente em Fortran, foi reescrito na linguagem C. A análise e o estudo do pseudo-código auxilia na compreensão do modelo implementado. O pseudo-código inferido do programa original é apresentado na Figura C.1

procedure eShuffle (T_frag, Ta, T_alin, N, seq[ ], delta_H, delta_S, con[ ])

{**Entrada**: T_frag → tamanho dos fragmentos a serem submetidos ao *shuffling*
Ta → temperatura de pareamento
T_alin → tamanho do alinhamento entre os parentais

Figura C.1. Pseudo-código do eShuffle (continua).

```
            N → número de parentais
            seq[ ] → lista com as seqüências parentais
            delta_H → valores de variação de entropia para todas as duplas de pares de bases
                      possíveis de nucleotídeos
            delta_S → valores de variação de entalpia para todas as duplas de pares de bases
                      possíveis de nucleotídeos
            con[ ] → lista das concentrações iniciais dos parentais

{Saída: Probabilidade das seqüências resultantes do shuffling seja o resultado de
        0, 1, 2, 3, …, 10 cruzamentos entre os parentais e número médio de
        cruzamentos nas seqüências resultantes}
begin
  sobrep_min = 2 {sobreposição mínima exigida entre dois fragmentos para que o pareamento ocorra}
  {calcule as probabilidades de ocorrer cruzamentos para cada uma das possíveis variantes. São
   considerados todos os possíveis fragmentos cujo tamanho é igual a T_frag}
  for i = T_alin, T_alin−1, ..., T_frag do {localização do fragmentos}
    Begin
      for k = 1, 2, 3, … N do          {parental}
        Begin
          {verifique qual pareamento entre esses fragmentos resulta na menor variação de energia livre}
          for p = 1, 2, 3, ..., N do       {parental}
            Begin
              for q = sobrep_min, sobrep_min + 1, … T_frag do {todas as sobreposições possíveis}
                Begin
                  {calcule o complemento da região do fragmento do parental p (de tamanho i) que
                   irá se sobrepor com a região do fragmento do parental k}
                  for pos = i−q, … i do
                    complemento[pos] = Complemento(seq[p][pos])
                  {calcule a variação de entalpia e entropia desta sobreposição}
                  dH[p][q] = Cacula_deltaH(complemento, i, k, p, q, delta_H)
                  dS[p][q]  = Cacula_deltaS(complemento, i, k, p, q, delta_S)
                end {fim for q}
            end {fim for p}
          {cálculo da probabilidade de todos os tamanhos possíveis de sobreposição}
          for p = 1, 2, 3, …, N do
```

Figura C.1. Pseudo-código do eShuffle (continuação).

```
        for q = sobrep_min, sobrep_min + 1, … T_frag do
          Mprob[p][q] = 0
    A = 0
    Delta_T = (94 – Ta) / 1000
    for r = 0, 2, 3, …, 1000 do    {cálculo da integral ao longo da temperatura}
      begin
        T = 94 – (r * Delta_T)
        if r == 0  ∨  r == 1.000 then
            coeff = 1
        else
          if (r%2) == 0 then
            coeff = 2
          else
            coeff = 4
          erro = 1
          while erro > 10^-8 do
            begin
              {verificar a probabilidade dos pareamentos ocorrerem}
              pi   = 0
              sig1 = 0
              sig2 = 0
              var1 = T + 273,15  {convertendo a temperatura para Kelvin}
              R    = 1,987       {valor da constante dos gases perfeitos (unidade: cal*K^-1 *mol^-1)}
              var2 = R * var1
              var3 = xA0 * (1,0 – a)
              for p = 1, 2, 3, ..., N do
               begin
                 for q = sobrep_min, sobrep_min – 1, ..., T_frag – 1 do
                  begin
                    Kep[p][q] = exp(-(1.000 * dH[p][q] – dS[p][q] * var1) / var2)
                    {concentração final do fragmento na mistura}
                    xF[p][q] = xF0[p][q] / (1 + Keq[p][q] * var3)
                    var4 = Keq[p][q] * xF[p][q]
```

Figura C.1. Pseudo-código do eShuffle (continuação).

```
                pi = pi + var4
                sig1 = sig1 + (Keq[p][q] * xF[p][q]2 / xF0[p][q]) * dH[p][q] / var2
                sig2 = sig2 + var42 / xF0[p][q]
            end {fim for q}
        end {fim for p}
        novo_a = pi / (1 + pi )
        erro = a – novo_a
        if erro < 0 then
            Erro = – erro
        a = novo_a
    end {fim while}
if r == 0  then
    begin
      {probabilidade dos fragmentos se desnaturarem}
      for p = 1, 2, 3, ..., N do
          for q = sobrep_min, sobrep_min – 1, ..., T_frag – 1 do
              Pdes[p][q] = Keq[p][q] * xF[p][q] / pi
      a_t = – sig1 / ((1.0 + pi)2 – xA0 * sig2);
      for p = 1, 2, 3, ..., N do
          for q = sobrep_min, sobrep_min – 1, ..., T_frag – 1 do
              mprob[p][q] = mprob[p][q]+coeff*(delta_T/ 3.0) *(Keq[p][q]*xF[p][q] / pi) * a_t
    end {fim r}
{Ajustando a probabilidade dos pareamentos que ocorrem a uma temperatura
  maior ou igual a temperatura de desnaturação}
for p = 1, 2, 3, ..., N do
    for q = sobrep_min, sobrep_min – 1, ..., T_frag – 1 do
        mprob[p][q] = mprob[p][q] + a_den * Pden[p][q]

{Ajustando a probabilidade dos pareamentos que não produzem extensão}
for p = 1, 2, 3, ..., N do
    for posição = i−1, ..., i−1 do
        if (seq[p][posição] != seq[k][posição]) then
            for q = 0, 1, 2, ..., T_frag – 1 do
                mprob[p][q] = mprob[p][q] + a_den * Pden[p][q]
```

Figura C.1. Pseudo-código do eShuffle (continuação).

```
for v = sobrep_min, sobrep_min – 1, ..., T_frag – 1  do
    if ( i + T_frag – v > T_alin + 1 ) then
        for  p = 1, 2, 3, ..., N do
            mprob[p][v] = 0
norm = 0
for p = 1, 2, 3, ..., N do
   for q = sobrep_min, sobrep_min  – 1, ..., T_frag – 1 do
       norm = norm + mprob[p][q]
{Normalizando o valor das probabilidades}
 for  p = 1, 2, 3, ..., N do
    for q = sobrep_min, sobrep_min  – 1, ..., T_frag – 1 do
        mprob[p][q] = mprob[p][q] / norm

{calculando a probabilidade de ocorrerem 0, 1, 2,..., 10 cruzamentos na biblioteca
 resultante}
for x = 0, 1, 2, ..., 10 do
 begin
   P[x][k][i] = 0
    if (x <= Tam_alin – i + 1) then
      for v = sobrep_min, sobrep_min – 1, ..., T_frag – 1 do
        begin
          if (x == 0) then
             for m = 1, 2, 3, ..., N do
               begin
                  xok = 0
                   posição = i
                    while (xok == 0) ∧ (posição ≤ (i+T_frag – v – 1)) ∧ (posição ≤ T_alin))
                      begin
                         if seq_parental[k][posição] != seq_parental[m][posição] then
                            xok  = 1
                         posição = posição + 1
                      end
                    if xok == 0 ∧ i ≤ 1 then
                      P[x][k][i] = P[x][k][i] + mprob[m][v] * P[x][k][i + T_frag – v]
```

Figura C.1. Pseudo-código do eShuffle (continuação).

```
                        end {for m}
                    else {x != 0}
                      for m = 1, 2, 3, …, N do
                        xok = 0
                        posição = i
                        while (xok == 0 ∧ posição ≤ (i + T_frag – v – 1) ∧ (posição ≤ T_alin))
                         begin
                           if (seq[k][posição] != seq[m][posição]) then
                             xok  = 1
                           posição = posição + 1
                         end
                         if (xok == 0 ∨ i ≤ 1) then
                           P[x][k][i] = P[x][k][i] + mprob[m][v] * P[x][k][i + T_frag – v]
                         else
                           P[x][k][i] = P[x][k][i] + mprob[m][v] * P[x – 1][m][i + T_frag – v]
                  end {fim v}
              end {fim x}
          end {fim k}
      end {fim i}
    {calculando a probabilidade final de que a biblioteca construída, considerando a remontagem
     de fragmentos de tamanho T_frag, possua seqüências com 0, 1, 2,... 10 cruzamentos}
    for x = 0, 1, 2, 3, …, 10 do
        Pfinal[x] = 0

    for p = 0, 1, 2, 3, …, 10 do
      begin
        for q = 1, 2, 3, …, N do
          Pfinal[p] = Pfinal[p] + con[q]  * P[p][q][T_frag + 1]
          write(PFinal[p])
      end
    média = 0    {calculando o número médio de cruzamentos nas seqüências resultantes}
    for p = 0, 1, 2, 3, …, 10 do
      média = média + p * Pfinal[p]
    write(média)
end
```

Figura C.1. Pseudo-código do eShuffle (conclusão).

C.3. Influência do modelo NN nos resultados do eShuffle

A fim de avaliar a influência dos valores de ΔS e ΔH nos resultados estimados pelo eShuffle, diversas simulações foram executadas considerando diferentes condições para experimentos de DNA *shuffling* (diferentes tamanhos de fragmentos e diferentes temperaturas de pareamento). A implementação original do eShuffle foi modificada para que, além dos valores originalmente utilizados pelo software para ΔS e ΔH (ALLAWI; SANTALUCIA, 1997), os conjuntos de valores propostos por Sugimoto et al. (1996), SantaLucia et al. (1996) e Breslauer et al. (1986), também fossem testados e as diferenças nos resultados avaliadas. Contudo, os valores predefinidos para $\Delta H_{iniciação}$ e $\Delta S_{iniciação}$, bem como para dois pares de bases vizinhos onde *mismatches* ocorrem foram mantidos como na implementação original.

Nas simulações, os genes atzA (número de acesso no GenBank P72156) e triA (número de acesso no GenBank AAG41202), ambos compostos por 1425 pb sendo apenas 9 pb diferentes entre eles, foram utilizados como parentais. O *shuffling* entre esses dois genes foi descrito por Raillard et al. (2001). Contudo, o tamanho dos fragmentos e a temperatura de pareamento utilizadas no experimento não foram relatados e, por isso, assumiu-se que fragmentos de tamanho entre 10 e 50 pb (como sugerido no protocolo original de Stemmer ((STEMMER,1994a) e (STEMMER, 1994b)) foram utilizados. Em seu trabalho, Raillard apresenta a seqüência de aminoácidos de 25 seqüências resultantes do *shuffling*, sendo possível estimar em 2,3 o número médio de cruzamentos entre as seqüências remontadas. Os resultados estimados, ou seja, o número médio de cruzamentos esperados nas seqüências resultantes em cada um dos experimentos de DNA *shuffling* simulados, considerando cada situação específica (tamanho do fragmento, temperatura de pareamento e valores ΔH e ΔS utilizados) estão sumarizados na Tabela C.3.

O número médio de cruzamentos estimado pelo eShuffle em todas as condições simuladas foi superior ao número médio de cruzamentos observado na mostra de seqüências resultantes do DNA *shuffling* descrito por Raillard. A supestimação do número de cruzamentos estimado pelo modelo pode ser atribuído à não detecção da ocorrência de cruzamentos silenciosos (ver definição no Capítulo 2, seção 2.6.3) ou até mesmo pelo baixo número de seqüências da amostra analisada (25 seqüências).

Tabela C.3. Número médio de cruzamentos estimados pelas imulações do eShuffle para os parentais atzA e triA.

	Temperatura de pareamento (ºC)					
Tamanho dos Fragmentos (pb)	45	50	55	60	65	Valores de ΔH e ΔS
10	3,34	3,41	3,48	3,57	3,69	ALLAWI; SANTALUCIA (1997)
20	3,51	3,42	3,33	3,28	3,27	
30	3,55	3,52	3,47	3,34	3,12	
40	3,55	3,55	3,54	3,53	3,50	
50	3,41	3,42	3,42	3,42	3,43	
60	3,29	3,29	3,30	3,30	3,31	
70	3,26	3,26	3,27	3,27	3,29	
80	3,17	3,18	3,18	3,19	3,19	
10	3,29	3,33	3,38	3,45	3,53	SUGIMOTO (1996)
20	3,61	3,55	3,47	3,36	3,28	
30	3,58	3,60	3,60	3,51	3,5	
40	3,55	3,55	3,55	3,55	3,54	
50	3,40	3,40	3,41	3,41	3,42	
60	3,29	3,41	3,30	3,30	3,30	
70	3,23	3,23	3,23	3,24	3,25	
80	3,14	3,15	3,15	3,15	3,16	
10	3,34	3,40	3,48	3,57	3,68	SANTALUCIA (1996)
20	3,54	3,46	3,37	3,32	3,31	
30	3,56	3,54	3,50	3,40	3,21	
40	3,56	3,56	3,56	3,55	3,53	
50	3,43	3,43	3,44	3,44	3,45	
60	3,31	3,31	3,32	3,32	3,33	
70	3,28	3,30	3,30	3,30	3,32	
80	3,21	3,21	3,21	3,22	3,23	
10	3,35	3,33	3,33	3,37	3,43	BRESLAUER (1986)
20	3,75	3,73	3,70	3,65	3,59	
30	3,62	3,62	3,61	3,60	3,59	
40	3,60	3,59	3,59	3,59	3,59	
50	3,44	3,44	3,44	3,45	3,45	
60	3,27	3,27	3,27	3,27	3,27	
70	3,20	3,21	3,21	3,21	3,21	
80	3,10	3,10	3,10	3,10	3,10	

Considerando as simulações com fragmentos de tamanho 10 pb, em todos os casos, independente dos valores de ΔH e ΔS utilizados, os resultados estimados pelo eShuffle mostram que, quanto maior a temperatura de pareamento maior é o número médio de cruzamentos nas seqüências remontadas. Contudo, como comentado por Volkov e Arnold (2000), a remontagem

de pequenos fragmentos, além de ser menos eficiente, requer temperaturas de pareamento mais baixas e, por isso, esperava-se que, nestas simulações, o número de cruzamentos diminuísse com o aumento da temperatura de pareamento. Ao contrário, como pode ser visto no Gráfico C.1, para fragmentos de tamanho 10 pb, quanto maior a temperatura de pareamento, maior o número médio de cruzamentos estimado pelo eShuffle para todos os diferentes valores de ΔH e ΔS. O número médio de cruzamentos estimado pelo eShuffle quando da utilização do parâmetros ΔH e ΔS segundo Allawi e SantaLucia (1997) e SantaLucia (1996) são praticamente os mesmos para todos os tamanhos de fragmentos. Para estas simulações, Breslauer resultou em números médios de cruzamentos mais baixos do que os demais modelos.

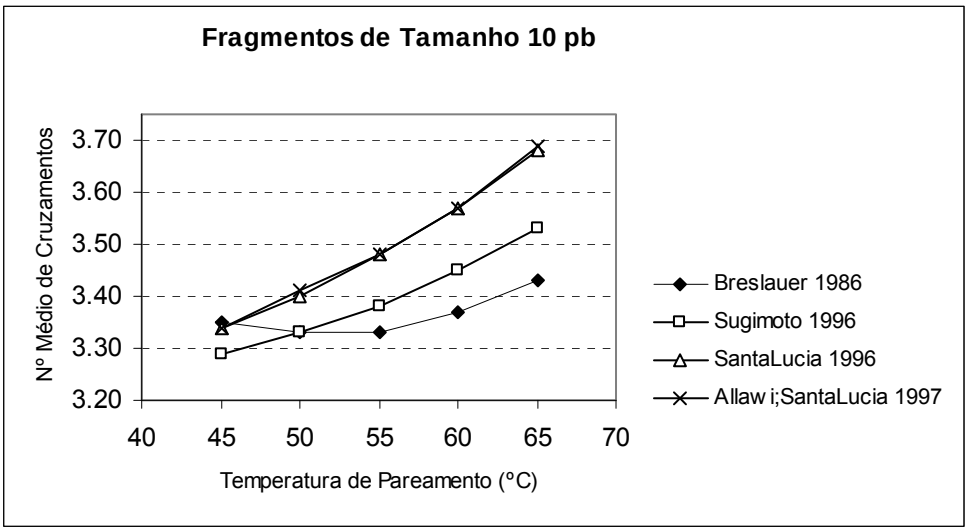


Gráfico C.1. Relação entre o número médio de cruzamentos estimado pelo eShuffle e a temperatura de pareamento considerando diferentes propostas para os valores dos parâmetros ΔH e ΔS e a remontagem de fragmentos de tamanho 10 pb.

Nas simulações considerando fragmentos de tamanhos 20, 30 e 40 pares de bases, a tendência do aumento do número de cruzamentos com a diminuição da temperatura de pareamento foi observada em todos os quatro modelos para os parâmetros NN, como mostram os gráficos de Gráfico C.2(a) a Gráfico C.2(c). Observou-se que, ao contrário dos resultados apresentados no Gráfico C.1, foi o modelo de Breslauer, seguido por Sugimoto, que resultaram num maior número médio de cruzamentos para todas as temperaturas de pareamento simuladas.

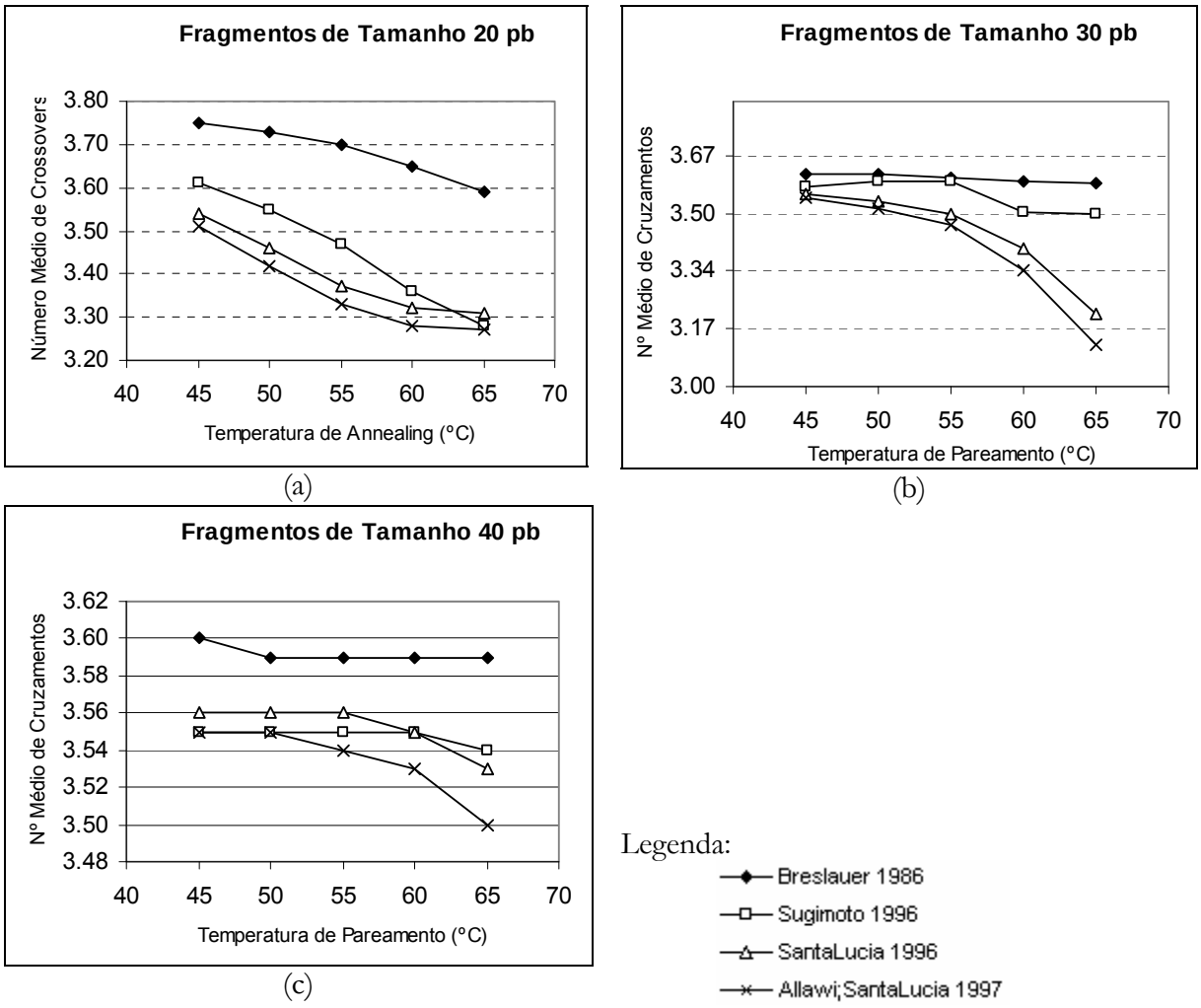


Gráfico C.2. Relação entre o número médio de cruzamentos estimado pelo eShuffle e a temperatura de pareamento considerando diferentes propostas para os valores dos parâmetros ΔH e ΔS e a remontagem de fragmentos de tamanho (a) 20 pb, (b) 30 pb e (c) 40 pb.

Assim como as simulações para fragmentos de tamanho 10 pb, para fragmentos maiores (50, 60, 70 e 80 pb), as simulações mostraram um pequeno aumento no número estimado de cruzamentos à medida que a temperatura de pareamento aumenta como mostram os gráficos Gráfico C.3(a) a Gráfico C.3(d).

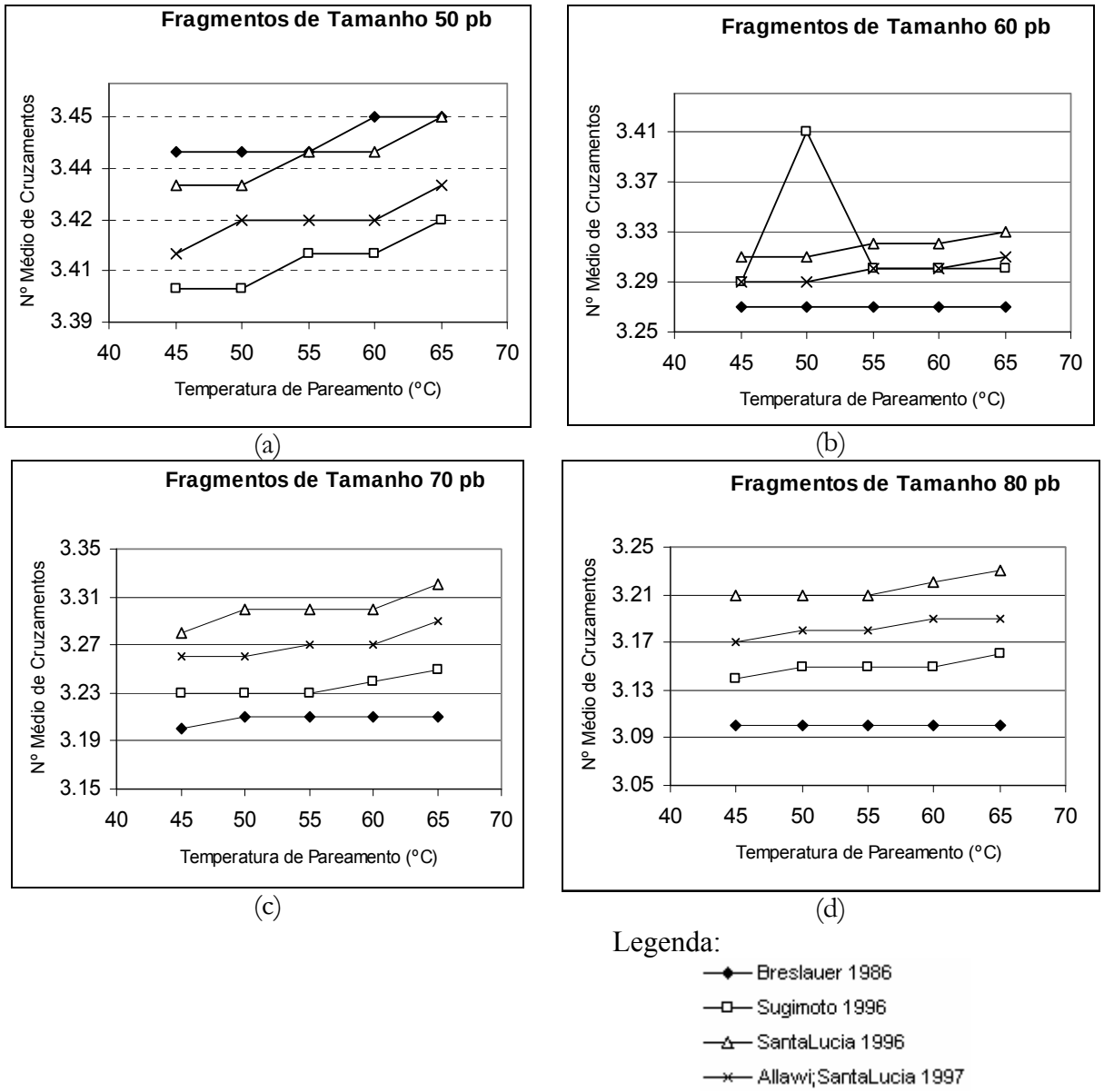


Gráfico C.3. Relação entre o número médio de cruzamentos estimado pelo eShuffle e a temperatura de pareamento considerando diferentes propostas para os valores dos parâmetros ΔH e ΔS e a remontagem de fragmentos de tamanho (a) 50 pb, (b) 60 pb (c) 70 pb e (d) 80 pb.

Com base nos resultados das simulações apresentadas, é possível afirmar que o número médio de cruzamentos estimado, considerando-se fixos o tamanho do fragmento e a temperatura de pareamento, sofre variações quando diferentes valores para os parâmetros do modelo NN são utilizados. Contudo, a ordem de grandeza das diferenças observadas não é significativa especificamente para o contexto que DNA *shuffling*, onde os resultados de simulações devem ser interpretados como valores aproximados e não determístico. Desta forma, conclui-se que a escolha de um modelo ou outro para a realização da simulação é arbitrária.

Apesar do foco principal das simulações apresentadas neste apêndice ser a análise da influência dos diferentes valores para os parâmetros do modelo NN sobre o número médio de cruzamentos estimado pelo eShuffle, este estudo revelou que o modelo implementado pelo eShuffle, em se tratando de simulações utilizando fragmentos pequenos (10 pb), bem como fragmentos maiores de tamanho entre 50 pb e 80 pb, não está de acordo com a afirmação de que, sob baixas temperaturas, a remontagem dos fragmentos fica facilitada (o que aumenta o número de cruzamentos) uma vez que sobreposições com *mismatches* são favorecidas.

APÊNDICE D – Determinação de *Primers* utilizando *Simulated Annealing*

D.1. Considerações iniciais

Este apêndice introduz o método de determinação de pares de *primers* baseado no algoritmo de *Simulated Annealing* descrito em Montera e Nicoletti (2008).

Primers são pequenas cadeias de nucleotídeos complementares a trechos de moléculas de DNA que se deseja amplificar (ver Figura 1.14 do Capítulo 1). Os *primers* são fatores determinísticos em experimentos baseados em PCR, tais como, síntese *in vitro* de DNA e diversos experimentos de Evolução Molecular Direta (DNA *shuffling*, *Random priming recombination*, mutação sítio-dirigida, etc.). A determinação dos *primers* a serem utilizados envolve questões referentes ao tamanho do *primer*, % de bases C e G, especificidade, presença de bases repetidas contínuas na seqüência do *primer*, entre outros.

A Seção D.2 apresenta uma breve discussão sobre as questões envolvidas na determinação de *primers*, ou seja, no projeto de *primers*. A Seção D.3 apresenta a proposta de uma implementação baseada no algoritmo de *Simulated Annealing* para o projeto de *primers*. Por fim, algumas conclusões são apresentadas na Seção D.4.

D.2. Questões envolvidas no projeto de *primers*

A determinação e escolha de um *primer* ou par de *primers* a ser utilizado em um experimento específico requer uma série de cuidados quanto ao tamanho, composição, temperatura de pareamento e especificidade dos *primers*, entre outros. Não existem valores específicos pré-definidos para os parâmetros envolvidos no projeto de *primers*, contudo, intervalos de valores para esses parâmetros são de senso comum. Segue uma breve descrição dos parâmetros envolvidos no projeto de *primers*.

***Repeats*, *Runs* e Estruturas secundárias**

A ocorrência repetida de uma subseqüência de nucleotídeos em posições consecutivas na seqüência de um *primer* é chamada *repeat*. *Repeats* devem ser evitados uma vez que eles podem favorecer o pareamento entre o *primer* e o *template* em posições não desejadas (evento também

chamado de *misprimer*), como mostrado na Figura D.1. *Runs* são definidos como a repetição consecutiva de uma única base na seqüência do *primer*.

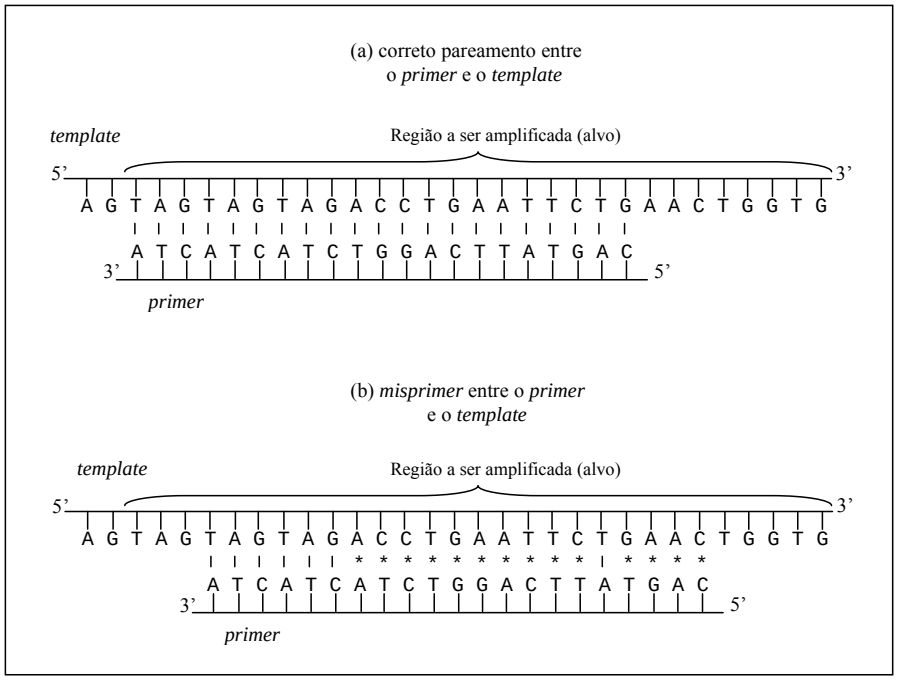


Figura D.1. *Primer* com *repeats* (ATC). Os *matches* ocorridos no pareamento entre o *primer* e o *template* são representados pelo símbolo |, enquanto que os *mismatches* são representados por *.

Se os *primers forward* (F) e *reverse* (R) apresentarem trechos complementares entre si, esses podem se parear, em detrimento do pareamento entre o *primer* e o *template*, formando uma estrutura secundária chamada *hetero-dimer*. Caso o pareamento ocorra entre dois *primers* F ou dois *primers* R, ocorre a formação de uma estrutura secundária denominada *self-dimer*. *Primers* longos podem se auto-parear, resultando em estruturas secundárias denominadas *hairpins*.

Especificidade e Tamanho do *primer*

Um *primer* é específico se ele se pareia com o *template* apenas na região específica para a qual ele foi projetado (ver região alvo da Figura D.1). *Primers* não específicos acarretam a produção (amplificação) de fragmentos de DNA não correspondentes à região alvo.

O tamanho de um *primer* influencia a sua especificidade (*primers* maiores são mais específicos); custo de produção (quanto maior o *primer*, mais elevado é o seu custo); estabilidade da formação *template–primer* (quanto maior o *primer*, mais fortemente ele estará unido ao *template* devido ao maior número de pontes de hidrogênio resultantes desta ligação); e formação de estruturas secundárias (*primers* maiores são mais propensos à formação de estruturas secundárias).

Não existe um tamanho fixo ótimo para um *primer*. *Primers* variando entre 18 e 30 bases são os melhores (KAMEL; ABD-ELSALAM, 2003).

%CG e Composição da Extremidade 3'

O conteúdo de bases Citosina (C) e Guanina (G) de um *primer* determina a temperatura na qual a reação de pareamento deve ocorrer basicamente pelo fato do pareamento destas bases ocorrer devido à formação de três pontes de hidrogênio enquanto que o pareamento entre as bases A e T ocorre devido à formação de duas pontes de hidrogênio. Quanto maior o número de bases C e G, mais fortemente "ligados" estarão o *primer* e o *template*. Valores entre 40% e 60% de CG na composição dos *primers* são preferidos.

A preferência por uma base C ou G na extremidade 3' do *primer* é justificada por ser esta a extremidade na qual a enzima polimerase inicia a extensão do *primer*. Como o pareamento C–G é mais estável (devido ao maior número de pontes de hidrogênio formadas entre essas duas bases), espera-se que a polimerase inicie o processo de síntese mais eficientemente neste caso.

Temperatura de *Melting* e Temperatura de *Annealing*

A temperatura de *melting* (T_m) é definida como a temperatura na qual metade dos fragmentos de DNA está na forma desnaturada, ou seja, não pareados, e a outra metade está pareada. A temperatura de pareamento ou *annealing* (T_a) é a temperatura na qual ocorre o pareamento entre *primer* e *template*.

Existem diversas aproximações para o cálculo da T_m de um *primer*, sendo possível, de uma maneira geral, dividi-las em três classes distintas:

(1) Básicas (consideram apenas a composição do *primer*);
(2) Dependentes do Sal (consideram a concentração de sal na reação onde os pareamentos ocorrem); e
(3) Baseadas na Termodinâmica da reação (utilizam-se do modelo *Nearest Neighbor*).

As equações D.1, D.2 e D.3 apresentam exemplos de fórmulas básica, dependentes do sal e baseadas em termodinâmica para o cálculo da T_m, propostas por Wallace et al. (1979), Howley et al. (1979) e Rychlik et al. (1990), respectivamente.

$$T_m = 2*(|A|+|T|)+4*(|C|+|G|) \tag{D.1}$$

$$T_m = 100{,}5 + 41*\frac{|C|+|G|-6{,}4}{|A|+|T|+|C|+|G|} - \frac{820}{|A|+|T|+|C|+|G|} + 16{,}6*\log[Na^+] \tag{D.2}$$

$$T_m = \frac{\Delta H}{\Delta S + R*\ln\frac{\gamma}{4}} - 273{,}15 + 16{,}6*\log[Na^+] \tag{D.3}$$

Nas equações (D.2) e (D.3), $[Na^+]$ é a concentração de sal na reação. Na eq. (D.3), R = 1,987 cal/ºC*mol é a constante molar dos gases, γ é a concentração do *primer* na solução. Nas equações D.1 e D.2, $|A|$, $|T|$, $|C|$ e $|G|$ representam, respectivamente, o número de bases A, T, C e G presentes nos *primers*. Como pode ser observado na eq. (D.3), a fórmula para o cálculo da T_m baseada no modelo *Nearest Neighbor* (NN) utiliza-se dos conceitos de variação de entalpia (ΔS) e entropia (ΔH). O modelo NN, bem como a determinação dos valores de ΔS e ΔH, foram previamente discutidos no Capítulo 3, Seção 3.4.2.

D.3. Proposta de um algoritmo para o projeto de *primers* baseado no conceito de *Simulated Annealing*

O algoritmo *Simulated Annealing* (SA) é muito utilizado na busca por soluções de problemas de otimização, ou seja, problemas que envolvem a busca em grandes espaços de busca. O problema de projeto de *primers* foi abordado como um problema de otimização, cuja busca no espaço de possíveis soluções é realizada por um procedimento baseado no SA.

Problemas de otimização buscam encontrar soluções que maximizam ou minimizam determinada(s) função. No contexto de projeto de *primers*, a função proposta para avaliar a adequabilidade de um par de *primers forward* e *reverse* (f, r) mede quão distantes determinados valores calculados para esse par (como por exemplo, tamanho, %CG, T_m, diferença entre T_m do *primer* f e do *primer* r e composição da extremidade 3') estão de valores pré-definidos. A função considera ainda a presença ou não de *repeats*, *runs*, possibilidade de formação de *self-dimer*, *hetero-dimer*, etc.

Antes de apresentar a função utilizada para medir a adequabilidade ou *fitness* de um par de *primers*, um conjunto de métricas para a avaliação de um *primer* precisa ser definido, como mostra a Tabela D.1.

Tabela D.1. Descrição das métricas implementadas para avaliação de um *primer* ou de um par de *primers*.

Métrica*	Descrição
len(p)	Determina o comprimento do *primer* p.
CG(p)	Determina o %CG do *primer* p.
T_m(p)**	Determina a T_m do *primer* p.
$T_m$dif(f, r)	Determina a diferença absoluta entre a T_m do *primer* f e r.
3'_end(p)	Verifica a existência de uma base C ou G na extremidade 3´ do do *primer* p.
run(p)	Verifica a existência de *runs* no *primer* p.
repeat(p)	Verifica a existência de *repeats* no *primer* p.
spec(p)	Verifica a especificidade do *primer* p.
sec(f, r)	Verifica a possiblidade de formação de estruturas secundária nos *primers* f e r.

highest_cost(G_i)	Associa um custo à formação (caso aconteça) de uma estrutura secundária do tipo i, para i = 1, ... , 5 sendo:
	i = 1, *hetero-dimer* entre os *primers* f e r;
	i = 2, *self-dimer* entre *primers* f;
	i = 3, *self-dimer* entre *primers* r;
	i = 4, *hairpin* no *primer* f;
	i = 5, *hairpin* no *primer* r.

* considere p como sendo o *primer* f ou r.
** A T_m(p) foi calculada como a média resultante das fórmulas apresentada nas equações (D.1), (D.2) e (D.3), visto que não há um consenso sobre qual delas é mais adequada.

A fim de avaliar a dequabilidade de um determinado par de *primers,* é preciso verificar sua conformidade em relação a um conjunto de valores pré-definidos para o tamanho, %CG, T_m, diferença de T_m entre os *primers* f e r, e composição da extremidade 3' além de verificar a existência (ou não) de estruturas como *runs*, *repeats*, possibilidade de formação de estruturas secundárias e especificidade. A Tabela D.2 descreve as variáveis cujos valores devem ser pré-definidos (pelo especialista humano) antes da busca por *primers*.

Tabela D.2. Variáveis cujos valores devem ser pré-definidos antes da busca por *primers*.

Parâmetro	Intervalo de valores possíveis	Descrição
LENGTH_INTERVAL	[MIN_LEN, MAX_LEN]	Tamanho mínimo e máximo permitido para um *primer*.
%CG_CONTENT	[MIN_CG, MAX_CG]	Porcentagem mínima e máxima de CG permitida na composição de um *primer*.
Tm	[MIN_Tm, MAX_Tm]	Valor mínimo e máximo permitidos para a T_m de um *primer*.
MAX_DIF		Máxima diferença permitida entre T_m do *primer* f e T_m do *primer* r.
3'_END		Preferência (3'_END = 1) ou não (3'_END = 0) por uma base C ou G na extremidade 3' dos *primers* f e r.

A função para avaliação de um par de *primers* utilizada na implementação do SA é dada na eq. (D.4). A função tot_cost mede quão próximos os valores das métricas calculadas para o par de *primer* estão dos valores pré-estabelecidos e quão bom é o par de *primers* em relação à sua especificidade, possibilidade de formação de estruturas secundárias e presença de *runs* e *repeats*, de tal forma que, quanto maior o valor de tot_cost associado a um par de *primers*, menor sua adequabilidade como solução do problema.

$$\begin{aligned} \text{tot\_cost(f, r)} = {} & \text{len\_cost(f)} + \text{len\_cost(r)} + \%\text{CG\_cost(f)} + \\ & + \%\text{CG\_cost(r)} + 3*(T_m\text{\_cost(f)} + T_m\text{\_cost(r)}) + \\ & + T_m\text{dif\_cost(f, r)} + 3'\text{\_end\_cost(f)} + \\ & + 3'\text{\_end\_cost(r)} + \text{run\_cost(f)} + \text{run\_cost(r)} + \\ & + \text{repeat\_cost(f)} + \text{repeat\_cost(r)} + \text{spec\_cost(f)} + \\ & + \text{spec\_cost(r)} + \text{sec\_struc\_cost(f, r)} \end{aligned} \tag{D.4}$$

Considerando novamente p como sendo o *primer* f ou o *primer* r, tem-se as seguintes definições para as funções utilizadas para a determinação do valor de tot_cost(f, r):

$$\text{len\_cost}(p) = \begin{cases} 0, \text{ se } \text{MIN\_LEN} \le \text{len}(p) \le \text{MAX\_LEN} \\ \text{MIN\_LEN} - \text{len}(p), \quad \text{se } \text{len}(p) < \text{MIN\_LEN} \\ \text{len}(p) - \text{MAX\_LEN}, \text{ se } \text{len}(p) > \text{MAX\_LEN} \end{cases}$$

$$\%CG\_cost(p) = \begin{cases} 0, \text{ se } MIN\_CG \le CG(p) \le MAX\_CG \\ MIN\_CG - CG(p), \text{ se } CG(p) < MIN\_CG \\ CG(p) - MAX\_CG, \text{ se } CG(p) > MAX\_CG \end{cases}$$

$$T_m\_cost(p) = \begin{cases} 0, \text{ se } MIN\_T_m \le T_m(p) \le MAX\_T_m \\ MIN\_T_m - T_m(p), \text{ se } T_m(p) < MIN\_T_m \\ T_m(p) - MAX\_T_m, \text{ se } T_m(p) > MAX\_T_m \end{cases}$$

$$T_m dif\_cost(f,r) = \begin{cases} 0, \text{ se } T_m dif(f,r) \le MAX\_DIF \\ T_m dif(f,r) - MAX\_DIF, \text{ caso contrário} \end{cases}$$

$$3'\_end\_cost(p) = \begin{cases} 0, \text{ se } 3'\_end(p) = 1 \\ 5, \text{ caso contrário} \end{cases}$$

$$run\_cost(p) = \begin{cases} 0, \text{ se } run(p) = 0 \\ 5 * \text{número de runs, caso contrário} \end{cases}$$

$$reapeat\_cost(p) = \begin{cases} 0, \text{ se } reapet(p) = 0 \\ 5 * \text{número de repeats, caso contrário} \end{cases}$$

$$spec\_cost(p) = \begin{cases} 0, \text{ se } spec(p) = 1 \\ 5 * \text{número sítio alternativos, caso contrário} \end{cases}$$

$$sec\_struc\_cost(f,r) = \begin{cases} 0 \text{ se } sec(f,r) = \text{false} \\ \sum_{i=1}^{5} highest\_cost(G_i), \text{ caso contrário} \end{cases}$$

O pseudo-código do algoritmo SA implementado para o problema de projeto de *primer* é apresentado na Figura D.2. O algoritmo se inicia escolhendo um par de *primers* randomicamente (referenciado como atual) tal que m = |f_atual| e n = |r_atual|, para MIN_LEN ≤ m, n ≤ MAX_LEN, e o *primer* f_atual corresponde as m primeiras bases da seqüência de DNA que se deseja amplificar (alvo) e o *primer* r_atual corresponde as n últimas bases complementares da seqüência de DNA alvo. Em seguida, o custo associado a este par de *primer* é calculado.

```
procedure SAPrimer
 begin
    f_atual     = find_primer(MIN_LEN, MAX_LEN)
    r_atual     = find_primer(MIN_LEN, MAX_LEN)
    cost_atual = tot_cost(f_atual, r_atual)
    T = 200
    decreasing_factor = 0,999
    while (T > 0,01)
        f_novo     = find_primer_neighbor(len(f_atual))
        r_novo     = find_primer_neighbor(len(r_atual))
        cost_novo = tot_cost(f_novo, r_novo)
        if (cost_novo < cost_atual) then
            // o par de primers atual passa a ser o novo par
            // encontrado pelo procedimento find_primer_neighbor()
            f_atual     = f_novo
            r_atual     = r_novo
            cost_atual = cost_novo
        else
            num = random( )
            if (num < exp(-ΔE/T)) then
               f_atual     = f_novo
               r_atual     = r_novo
               cost_atual = cost_novo
        T = T * decreasing_factor
 end
```

Figura D.2. Pseudo-códido do algoritmo SA implementado para o problema de projeto de *primers*.

A cada passo, o algoritmo escolhe, randomicamente, um novo par de *primers* na vizinhança do par atual como sendo qualquer par de *primers* tal que $|f\_atual| - 3 \leq |f\_novo| \leq |f\_atual| + 3$ e $|r\_atual| - 3 \leq |r\_novo| \leq |r\_atual| + 3$. O custo associado a este novo par de *primers* é calculado e comparado com o custo da solução atual e aquele, cujo custo associado é menor, se torna o par de *primers* atual. Contudo, mesmo que o par atual tenha um menor custo associado, a nova solução encontrada pode ainda se tornar a solução atual dependendo do valor de uma função de probabilidade, que é dependente do parâmetro temperatura (T) e do valor de ΔE, onde ΔE representa a diferença absoluta entre o valor de custo associado à nova solução e a solução atual. O algoritmo implementa a possibilidade de aceite de uma solução com maior custo associado como uma medida que visa diminuir as chances de encontrar uma solução que represente apenas um mínimo local (solução sub-ótima). Note, porém, no pseudo-código, que as chances de se aceitar soluções piores diminui à medida que o número de execuções do algoritmo se aproxima do final; dessa forma, tem-se a garantia de que boas soluções não serão mais desconsideradas com a proximidade do fim das buscas por novas soluções.

Os valores T = 200 e decreasing_factor = 0,999 foram determinados empiricamente e estão de acordo com valores propostos na literatura.

D.4. Conclusões

A busca de pares de *primers* utilizando a implementação proposta baseada em SA mostrou-se eficaz do ponto de vista da solução encontrada. Mesmo para os casos nos quais não existem pares de *primers* que respeitem todas as restrições (por exemplo, quando é impossível encontrar *primers* que, respeitem, ao mesmo tempo, os limites impostos para %CG e tamanho), o SA sempre encontra a melhor solução possível, ou seja, o par de *primer* que mais se aproxima do par ótimo sempre será encontrado.

APÊNDICE E – Medida para Avaliar a Adequabilidade de Parentais Candidatos ao Processo de DNA *shuffling* – uma Proposta Baseada em Mutações

E.1. Considerações iniciais

A medida de adequabilidade de seqüências candidatas ao processo de *shuffling* descrita na Seção 4.2.2.1.2 do Capítulo 4 foi comparada com outras duas medidas, uma baseada nos conceitos de similaridade e distância entre seqüências e outra baseada no conceito de complexidade de Kolmogorov.

A medida de similaridade entre duas seqüências X e Y, representada por s(X, Y), expressa quão 'parecidas' entre si são as seqüências X e Y. A similaridade pode ser calculada por meio da soma de *matches, mismatches* e *gaps* existentes no alinhamento entre X e Y, como visto no Capítulo 4, Seção 4.2.2.1. Similarmente, a distância entre duas seqüências X e Y, representada por d(X, Y), expressa o conceito de quão distintas são as seqüências X e Y. Essas duas medidas podem ser consideradas complementares de maneira que s(X, Y) = 1 – d(X, Y). Modelos mais elaborados para o cálculo da distância, entretanto, se utilizam de matrizes de substituição (ver Jukes e Cantor (1969), Kimura (1980), Kishino e Hasegawa (1989) e Felsenstein e Churchill (1996)). As matrizes de substituição descrevem a taxa na qual cada caractere de uma seqüência (nucleotídeo ou aminoácido) é substituído por outro.

As três próximas seções E.2, E.3 e E.4 descrevem, respectivamente, a medida baseada no conceito de distância entre seqüências, a medida baseada no conceito de complexidade de Kolmogorov e a medida proposta neste trabalho, que é a baseada na distância média que separa mutações consecutivas entre os parentais.

Um conjunto de 37 seqüências de DNA codificadoras para metalopeptidases de serpente foram utilizadas para comparações entre as três medidas, descritas na Seção E.5. As SVMPs (*Snake Venom Metallopeptidades*) são uma família de enzimas que desempenham diversas atividades biológicas como degradação de fatores coaguladores do sangue (CHEN et al., 2004), e apoptose de diferentes tipos celulares (WU; HUANG, 2003). Estas enzimas têm sido intensamente pesquisadas, devido sua relevância patológica e aplicações potenciais.

E.2. Medida de Similaridade Baseada no Conceito de Distância

Para o cálculo da medida baseada no conceito de distância o programa DNAdist foi utilizado. DNAdist é um dos programas do pacote PHYLIP (*Phylogeny Inference Package*). O pacote PHYLIP agrupa diversos programas de computador para o cálculo de distâncias entre seqüências bem como para inferência de filogenia. O pacote pode ser gratuitamente obtido no endereço http://evolution.genetics.washington.edu/phylip.html.

O DNAdist calcula a distância entre seqüências de DNA ou proteínas utilizando diferentes matrizes de substituição. Para os experimentos conduzidos, a matriz F84 (FELSENSTEIN; CHURCHILL, 1996) foi utilizada. DNAdist espera como entrada um alinhamento entre as seqüências para as quais a distância (duas-a-duas) deverá ser calculada. Assim, o programa foi executado para o conjunto de 37 SVMPs e uma matriz de distâncias, referenciada como $M_{distância}$, foi produzida. Nesta matriz, a notação $M_{distância}$ $[S_i][S_j]$ representa a distância calculada entre as seqüências S_i e S_j, para $1 \leq i, j \leq 37$ e $i \neq j$.

E.3. Medida de Similaridade Baseada no Conceito de Complexidade de Kolmogorov

Diferentemente das outras duas medidas de distância utilizadas neste estudo, a medida baseada no conceito de Complexidade de Kolmogorov é uma medida de distância que não se baseia no alinhamento entre as seqüências. Métodos para a comparação de seqüências não baseados em alinhamento não são muito comuns, embora alguns poucos resultados publicados (VINGA; ALMEIDA, 2003) evidenciam que essa abordagem pode ser promissora (KOCSOR et al., 2006) , (LI et al., 2001). A distância entre duas seqüências X e Y como proposta por Li et al. (2001) é dada pela equação (E.1), na qual K(X) é a complexidade de Kolmogorov de X, K(X|Y) é a complexidade condicional de Kolmogorov de X dado Y e K(XY) é a complexidade de Kolmogorov da seqüência resultante da concatenação de X e Y.

$$d(X,Y) = 1 - \frac{K(X) - K(X \mid Y)}{K(XY)} \qquad \text{(E.1)}$$

A complexidade condicional de Kolmogorov K(X|Y) é interpretada como o tamanho do menor programa de computador que tem como saída X quando Y é dado como entrada, considerando um computador universal. K(X) deve ser vista como K(X|ε) tal que ε é a seqüência vazia.

Kocsor et al. (2006) afirmam que a complexidade de Kolmogorov é um limite teórico que pode apenas ser aproximado. Nos estudos realizados por Kocsor, bem como no estudo apresentado neste apêndice, a complexidade condicional de Kolmogorov foi aproximada utilizando-se algoritmos de compactação de forma que K(X|Y) representa o comprimento da seqüência X resultante da compactação de Y. Mais especificamente, o programa de compactação *GenCompress* foi utilizado. *GenCompress* pode ser gratuitamente obtido no endereço http://www.bioinformatics.uwaterloo.ca/downloads/gencompress. Dessa forma, a distância entre duas seqüências calculada com base na complexidade de Kolmogorov pode ser aproximada pela equação (E.2).

$$d(X,Y) = 1 - \frac{\text{GenCompress}(X \mid \varepsilon) - \text{GenCompress}(X \mid Y)}{\text{GenCompress}(XY)} \tag{E.2}$$

Qualquer outro programa de compactação poderia ser utilizado. Para fins comparativos, o programa de compactação ppmz2, que pode ser gratuitamente obtido no endereço http://www.cbloom.com/src/ppmz.html, também foi utilizado para o cálculo da distância. Porém a fórmula proposta por Cilibrasi e Vitanyi (2005), descrita pela eq. (E.3), na qual o programa de compactação é o ppmz2, foi utilizada.

$$d(X,Y) = 1 - \frac{\text{ppmz2}(XY) - \min\{\text{ppmz2}(X), \text{ppmz2}(Y)\}}{\max\{\text{ppmz2}(X), \text{ppmz2}(Y)\}} \tag{E.3}$$

Como não houve diferenças significativas na utilização das fórmulas descritas pelas equações (E.2) e (E.3), apenas a fórmula descrita pela eq. (E.2) foi utilizada. Após o cálculo das medidas entre todos os pares de seqüências, a matriz $M_{Kolmogorov}[S_i][S_j]$, para $1 \leq i, j \leq 37$ e $i \neq j$ foi obtida.

E.4. Proposta da Medida de Similaridade Baseada em Mutações

A medida de similaridade baseada no número e na distância entre as mutações consecutivas existentes entre duas seqüências foi descrita no Capítulo 4, Seção 4.2.2.1.2. A medida estabelece que a relação de proximidade entre duas seqüências é inversamente proporcional à distância média que separa mutações consecutivas, ou seja, quanto mais distantes entre si estão estão as mutações, mais próximas estão as seqüências avaliadas.

E.5. Agrupamento (*Clustering*) de Seqüências

Para visualizar a relação entre as 37 seqüências de SVMPs estabelecidas pela utilização de cada uma das três medidas descritas, as 37 seqüências foram agrupadas pelo método *Neighbor-Joining* (SAITOU; NEI, 1987) implementado pelo programa Neighbor, também pertencente ao pacote PHYLIP.

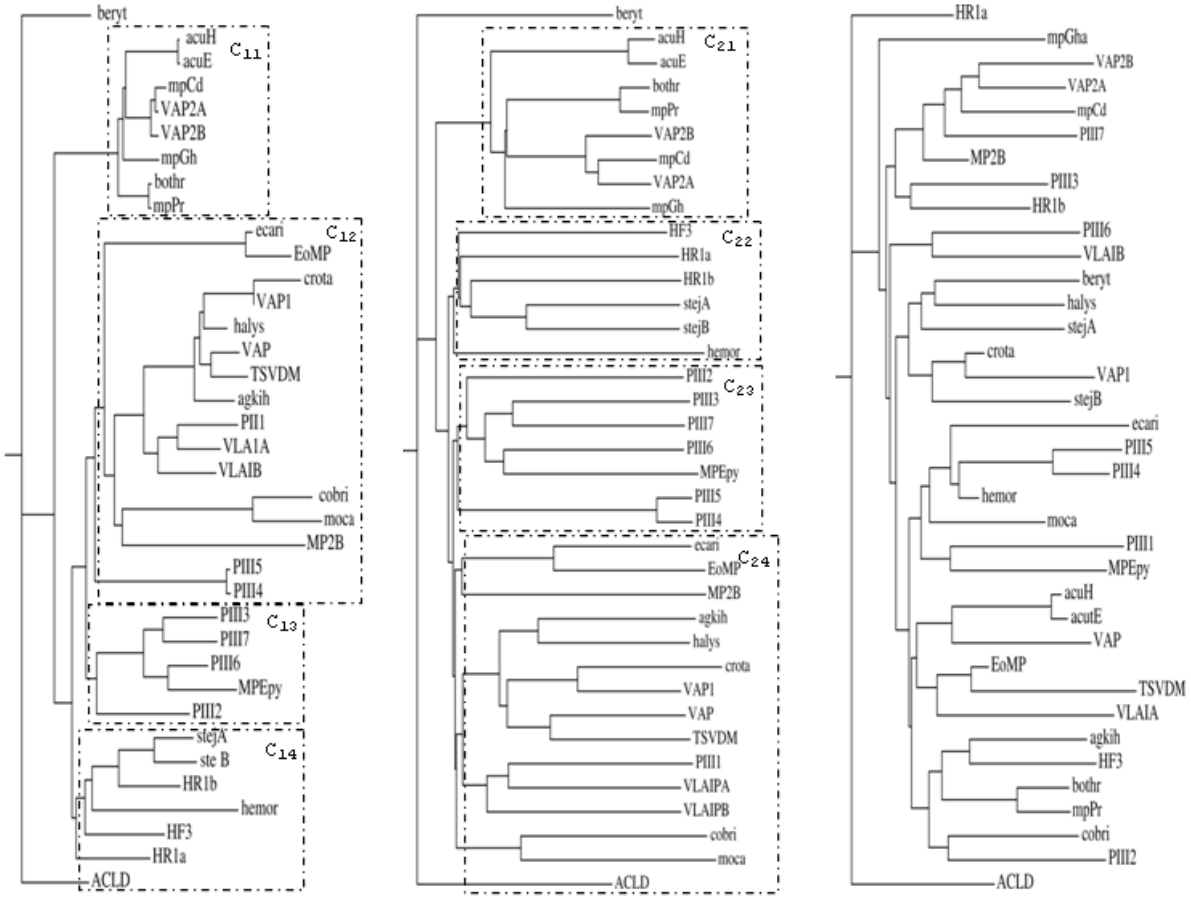
Dada uma matriz M de distâncias dois-a-dois entre um grupo de p seqüências, o método *Neighbor-Joing* constrói o agrupamento entre as p seqüências da seguinte maneira: primeiramente, as duas seqüências cuja distância é mínima dentre todos os outros pares de seqüências são agrupadas em um único par ou *cluster*. A distância deste novo par para todas as outras seqüências restantes ($p - 2$) é recalculada. Esse processo continua até que todas as seqüências pertençam a um único grupo. A relação entre as p seqüências identificadas pelo programa Neighbor pode ser graficamente visualizada pela execução do programa Drawgram (pacote PHYLIP), que desenha uma árvore de relacionamento entre as seqüências[38].

A Figura E.1 (a), (b) e (c) mostra as relações entre as 37 seqüências de SVMPs resultantes da execução do programa Neighbor seguido pelo programa Drawgram, considerando as matrizes de distância $M_{distância}$, $M_{Kolmogorov}$ e $M_{mutação}$, respectivamente. Os diagramas a) e b) são bastante similares, apesar de serem o resultado de duas medidas com abordagens completamente distintas, sendo apenas uma delas baseada no alinhamento entre seqüências (diagrama da Figura E.1 (a)).

A fim de analisar os diagramas (a) e (b), alguns de seus *clusters* foram assinalados e nomeados. A principal diferença entre os diagramas (a) e (b) consiste apenas nas posições relativas dos *clusters* (observe, por exemplo, os *clusters* C_{12} e C_{24}, C_{14} e C_{22}) e de alguns elementos dentro dos *clusters*. Considere, por exemplo, as posições relativas das seqüências stejA e stejB nos *clusters* C_{14} e C_{22}, respectivamente. A diferença de suas posições relativas é devido aos diferentes valores calculados para $M_{distância}$[stejA][stejB] e $M_{Kolmogorov}$[stejA][stejB]. Entretanto, de uma forma geral, as informações contidas nos diagramas (a) e (b) podem ser consideradas iguais.

Diferenças evidentes podem ser encontradas na comparação dos diagramas (a) e (b) com o diagrama (c). Estas diferenças eram, entretanto, esperadas uma vez que o propósito da medida $M_{mutação}$ é de avaliar as seqüências em relação, especificamente, ao problema de DNA *shuffling*, enquanto as outras medidas são comumente utilizadas para a inferência da relação de filogenia.

[38] Em problemas de filogenia, esta árvore é chamada árvore de filogenia.



(a) $M_{distância}$ (b) $M_{Kolmogorov}$ (c) $M_{mutação}$

Figura E.1. Diagramas criados pelo programa Neighbor e o programa Drawgram para o conjunto de 37 SVMPs. (a) Relacionamento resultante da medida $M_{distância}$. (b) Relacionamento resultante da medida $M_{Kolmogorov}$. (c) Relacionamento resultante da medida $M_{mutação}$.

E.6. Resultados

Como comentado na Seção E.1, o objetivo da investigação de medidas foi buscar evidenciar uma que possibilitasse avaliar a adequabilidade de seqüências candidatas à parentais em experimentos de DNA *shuffling*. Uma medida baseada em mutações foi proposta e comparada com outras duas medidas, utilizadas genericamente com o objetivo de expressar a relação de proximidade entre seqüências.

É no contexto de experimentos de DNA *shuffling* que os diagramas resultantes das três medidas devem ser avaliados. Quanto mais próximas duas seqüências estão em um agrupamento, maior é a relação entre elas. Contudo, nem sempre é possível estabelecer uma relação direta entre duas seqüências pelo diagrama. Considere, por exemplo, o diagrama da Figura E.1 (c). Não é possível dizer exatamente qual é a seqüência mais relacionada à seqüência MP2B. Sabe-se,

entretanto, que essa seqüência está mais relacionada ao grupo contendo as seqüências PIII7, mpCd, VAP2A e VAP2B do que com qualquer outra seqüência do diagrama. No contexto de DNA *shuffling*, é possível dizer que a escolha de qualquer uma das seqüências do conjunto PIII7, mpCd, VAP2A e VAP2B como parental, juntamente com a seqüência MP2B, resultaria em melhores resultados do que a escolha da seqüência MP2B e qualquer outra seqüência do diagrama.

A fim de validar os resultados obtidos com a medida proposta e compará-la com as outras duas medidas apresentadas, o software eShuffle foi utilizado para estimar o número médio de cruzamentos nas seqüências resultantes do *shuffling* entre os pares de seqüências sugeridos pela utilização de cada uma das três medidas.

Os experimentos descritos a seguir foram conduzidos da seguinte maneira: sempre que um par de seqüências (S_i, S_j) é selecionado[39] de um dos diagramas (a), (b) ou (c) por ser o par mais próximo, este mesmo par é buscado nos outros diagramas. Se o par (S_i, S_j) for encontrado, é armazenado; caso contrário, o par mais próximo (S_i, S_k) e/ou (S_k, S_j) para $i \neq j \neq k$, caso exista, é armazenado. Uma vez selecionados os pares de seqüências, o eShuffle foi executado para avaliar a adequabilidade destes pares como parentais em experimentos de DNA *shuffling*.

No diagrama (a) da Figura E.1 tem-se que a seqüência mais relacionada a seqüência cobri é a seqüência moca e no diagrama (c) é a PIII2. A execução do eShuffle resultou em um número médio de 0,84 cruzamentos por seqüência resultante do *shuffling* entre os cobri – moca *versus* um número médio de 5,99 cruzamentos para os parentais cobri – PIII2. A Tabela E.1 apresenta o número médio de cruzamentos estimados pelo eShuffle para pares de seqüências selecionadas como descrito no parágrafo anterior.

Para todos os pares de seqüências mais próximas apresentados na Tabela E.1, exceto para o par VAP2A–VAP2B, considerando a simulação do eShuffle para fragmentos de tamanho f = 35 pb, o número estimado de cruzamentos nas seqüências resultantes do *shuffling* entre o par de parentais (S_i, S_j) sugerido pela medida $M_{mutação}$ é maior que o número estimado de cruzamentos resultantes de qualquer outro par (S_i, S_k) ou (S_k, S_j), para $i \neq j \neq k$ e $1 \leq i, j, k \leq p$ sugerido pela medida $M_{distância}$ ou pela medida $M_{Kolmogorov}$. As diferenças entre o número de cruzamentos estimados para cada um dos pares de parentais sugeridos pelas três medidas avaliadas ficam mais evidentes no Gráfico E.1.

[39] Note que $(S_i, S_j) = (S_j, S_i)$.

Tabela E.1. Estimativas do número médio de cruzamentos considerando pares de parentais sugeridos pelas medidas $M_{mutação}$, $M_{distância}$ e $M_{Kolmogorov}$.

Par de seqüências	Valor de $M_{mutação}$	Medida que resultou no par de parentais	Nº médio de cruzamentos (eShuffle)	
			f = 45 bp	f = 35 bp
VAP2B–VAP2A	0,027927	$M_{Mutação}$	3,35	5,53
VAP2A–mpCd	0,028050	$M_{distância}$, $M_{Kolmogorov}$	3,33	6,57
PIII3–HR1b	0,036479	$M_{Mutação}$	6,11	6,28
PIII3–PIII7	0,049598	$M_{distância}$, $M_{Kolmogorov}$	4,34	3,88
PIII6–VLAIB	0,042155	$M_{Mutação}$	6,06	5,37
PIII6–MPEpy	0,052288	$M_{distância}$, $M_{Kolmogorov}$	2,16	2,42
beryt–halys	0,038961	$M_{Mutação}$	6,25	5,83
beryt–ACLD	0,058127	$M_{distância}$, $M_{Kolmogorov}$	1,60	0,53
halys–agkih	0,055556	$M_{Kolmogorov}$	1,40	0,87
crota–VAP1	0,021002	$M_{distância}$, $M_{Kolmogorov}$, $M_{Mutação}$	4,63	5,36
PIII5–PIII4	0,017857	$M_{distância}$, $M_{Kolmogorov}$, $M_{Mutação}$	4,79	4,79
PIII1–MPEpy	0,046875	$M_{Mutação}$	4,94	5,15
PIII1–VLAIA	0,061179	$M_{distância}$, $M_{Kolmogorov}$	0,21	0,08
acuE–acuH	0,003734	$M_{distância}$, $M_{Kolmogorov}$, $M_{Mutação}$	2,04	2,04
EoMP–TSVDM	0,025907	$M_{Mutação}$	3,39	4,63
EoMP–ecari	0,052097	$M_{distância}$, $M_{Kolmogorov}$	1,28	0,67
TSVDM–VAP	0,055819	$M_{distância}$, $M_{Kolmogorov}$	0,11	0,05
bothr–mpPr	0,014925	$M_{distância}$, $M_{Kolmogorov}$, $M_{Mutação}$	2,50	2,50
cobri–PIII2	0,040904	$M_{Mutação}$	5,99	5,80
cobri–moca	0,046845	$M_{distância}$, $M_{Kolmogorov}$	0,84	0,63

E.7. Considerações e Perspectivas

A investigação realizada com foco na determinação da adequabilidade de parentais a serem submetidos a experimentos de DNA *shuffling*, além de contribuir para um melhor entendimento do processo, resultou na proposta de uma nova medida específica para avaliar a adequabilidade de pares de seqüências como parentais neste tipo de experimento. Diferente de outras medidas convencionais que avaliam duas (ou mais) seqüências com base em suas características evolutivas, a medida proposta tem foco apenas em experimentos de DNA *shuffling*. A especificidade da

medida contribui para seu melhor desempenho de avaliação conforme discutido anteriormente (ver Gráfico E.1).

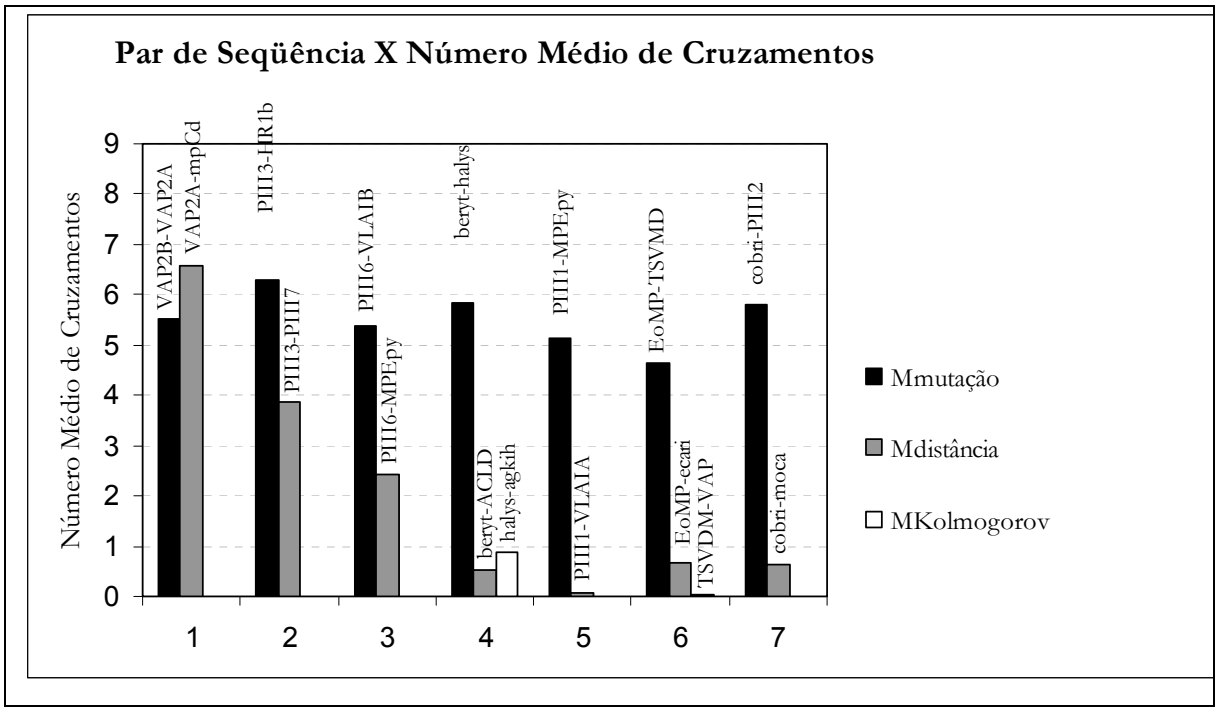


Gráfico E.1. Relação entre pares de parentais e número médio de cruzamentos estimados pelo eShuffle (para fragmentos de tamanho 35 pb) nas seqüências resultantes do DNA *shuffling* entre os parentais.

A medida proposta é baseada no número e na distância entre mutações consecutivas identificadas entre duas seqüências parentais. Essa medida foi comparada com outras duas medidas utilizando-se para tal um grupo de 37 enzimas. Dos 40 testes realizados, em apenas um a simulação *in silico* do DNA *shuffling* pelo software eShuffle utilizando como parentais pares de seqüências sugeridas pelas medidas $M_{distância}$ e $M_{kolmogorov}$ resultou num número médio de cruzamentos maior. Em todos os outros casos, a medida proposta sugeriu pares de parentais que resultam em um número esperado de cruzamentos bem maior do que o esperado pelos pares sugeridos pelas duas outras medidas.

O esquema proposto de análise dos parentais (agrupamento das seqüências utilizando a medida proposta seguida da visualização gráfica do agrupamento e da simulação *in silico* do processo de DNA *shuffling*) pode ser particularmente importante quando se trabalha com um grande conjunto de seqüências. É importante lembrar, também, que nenhuma outra medida de avaliação da adequabilidade de parentais focalizando especificamente o problema de DNA *shuffling*, foi encontrada na literatura.

APÊNDICE F – Conceitos de Estatística

F.1. Tentativa de Bernoulli, seqüência de Bernoulli e processo de Poisson

As definições que seguem foram compiladas de Beck (2006).

Situações nas quais apenas um, dentre dois resultados é possível, como no caso da fragmentação das seqüências parentais onde, para cada ligação fosfodiéster entre dois nucleotídeos esta ligação é quebrada ou não, são chamadas de Tentativas de Bernoulli, sendo p a probabilidade de ocorrência do evento de interesse e q = 1 – p a probabilidade de não ocorrência. A repetição independente de experimentos do tipo tentativa de Bernoulli dá origem a uma seqüência chamada de Binomial ou Seqüência de Bernoulli, caracterizada pelas seguintes suposições:

- cada tentativa possui apenas dois resultados possíveis;
- a probabilidade p de cada tentativa permanece constante ao longo das n tentativas;
- as tentativas são estatisticamente independentes.

Por serem independentes, dada uma seqüência W composta por n tentativas, a probabilidade de que aconteçam x ocorrências do evento e (n – x) não-ocorrências é dada por:

$$P[W] = p^{x} q^{(n-x)} \tag{F.1}$$

O número total de seqüências de Bernoulli distintas resultantes de x ocorrências e (n – x) não-ocorrências de um determinado evento é dado pela combinação de x ocorrências em n:

$$\binom{n}{x} = \frac{n!}{x!(n-x)!} \tag{F.2}$$

Do total de seqüências de Bernoulli nas quais aconteceram x ocorrências e (n – x) não-ocorrências de um determinado evento (dado pela eq. (F.2)), a ocorrência de acada uma delas é equiprovável, de forma que a probabilidade de ocorrência de uma seqüência particular deste tipo é dada por:

$$P[\{x \text{ ocorrências em } n \text{ tentativas}\}] = \binom{n}{x} p^x q^{(n-x)} \tag{F.3}$$

Sendo X a variável aleatória que conta o número x de ocorrências em n realizações deste evento, a função de densidade de probabilidade[40] de X é dada por:

$$P[\{X = x\}] = P[\{x \text{ ocorrências em } n \text{ realizações}\}] = \binom{n}{x} p^x q^{(n-x)} \tag{F.4}$$

A função de distribuição cumulativa de probabilidades é dada por:

$$P[\{X \leq x\}] = \sum_{k=0}^{x} \binom{n}{k} p^k q^{(n-k)} \tag{F.5}$$

Um caso particular da seqüência de Bernoulli ocorre quando os eventos acontecem ao longo de um contínuo, com intervalos de tempo ou espaço tendendo a zero, o número de tentativas de Bernoulli tende ao infinito ($n \to \infty$) e a probabilidade de ocorrência do evento de interesse tende a zero ($p \to 0$). Este caso particular da seqüência de Bernoulli é conhecido como processo de Poisson. A variável aleatória que conta o número médio de ocorrências do evento em um determinado intervalo de tempo t é dada por $\lambda = np$. A constante λ caracteriza o processo de Poisson. A taxa média (v) de ocorrência do evento de interesse ao longo do intervalo de tempo t é dada por:

$$v = \frac{\lambda}{t} \text{ ou } \lambda = vt = np \tag{F.6}$$

Com o número de tentativas de Bernoulli tendendo ao infinito ($n \to \infty$), pode-se mostrar (ANG; TANG, 1975, pg.116) que a probabilidade P[{X = x}], dada pela eq. (F.4), que é a função de densidade de probabilidade da variável aleatória X, e que descreve o número de ocorrências do evento ao longo do intervalo de tempo t, tende a:

$$P[\{X = x\}] = P[\{x \text{ ocorrências em } t\}] = \frac{(vt)^x}{x!} \exp[-vt] = \frac{\lambda^x}{x!} \exp^{-\lambda} \tag{F.7}$$

A função de distribuição cumulativa de probabilidades é dada por:

[40] A função de densidade de probabilidade é uma função utilizada para representar a distribuição de probabilidade de uma variável aleatória contínua.

$$P[\{X \le x\}] = \sum_{k=0}^{x} \frac{\lambda^x}{x!} \exp^{-\lambda} \tag{F.8}$$

O processo de Poisson foi utilizado para geração de números aleatórios em torno do valor médio dos tamanhos dos fragmentos desejados para a simulação do experimento de DNA *shuffling*. Considere n números aleatórios gerados em torno do valor médio L. O Gráfico F.1 apresenta um histograma para 5.000 valores aleatórios gerados pela distribuição de Poisson em torno do valor médio L = 158. Quanto mais distante um valor estiver valor médio, mais rara é a sua ocorrência dentre os valores aleatórios gerados. Para efeitos de simulação da fragmentação das seqüências parentais, o usuário pode escolher utilizar-se apenas de fragmentos cujo tamanho está compreendido entre um valor mínimo e máximo, dados o conjunto de números gerados em torno de L.

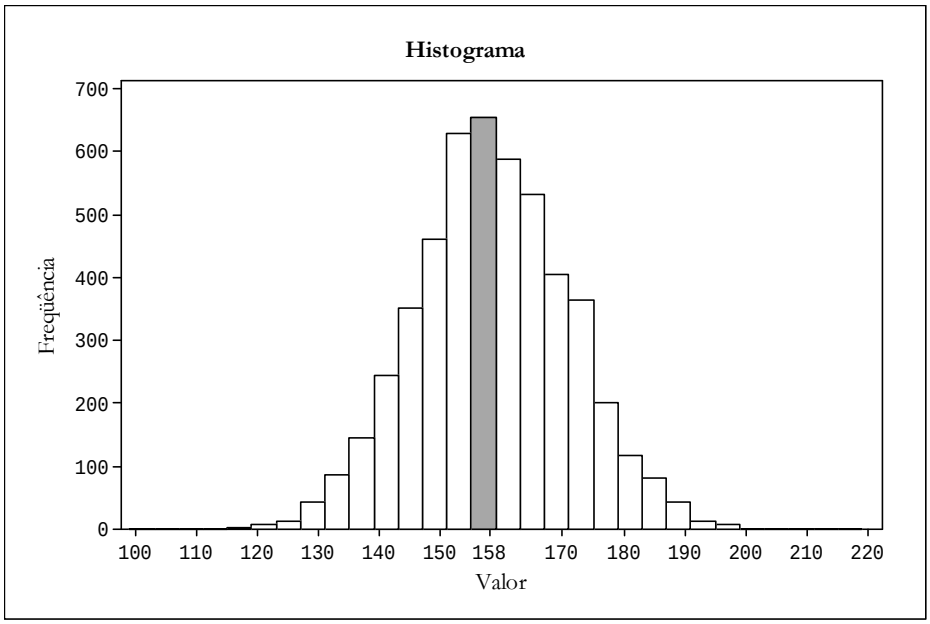


Gráfico F.1. Distribuição dos números aleatórios gerados segundo o processo de Poisson em torno do valor médio 158.

F.2. Geração de números aleatórios

A fragmentação considerando fragmentos aleatórios cujo tamanho está compreendido num intervalo de tamanho máximo e mínimo (Min, Max) foi implementada pela seqüência de equações (F.9), (F.10) e (F.11).

$$\text{número1} = \frac{\text{rand}(\,)}{\text{RAND\_MAX} + 1} \tag{F.9}$$

$$número2 = número1 * (Max - Min + 1) \quad \text{(F.10)}$$

$$número\_aleatório = Min + número2 \quad \text{(F.11)}$$

tal que:

- rand( ): é uma função implementada pela maioria das linguagens de programação que retorna um valor inteiro aleatório entre 0 e RAND_MAX;
- RAND_MAX: é o valor máximo retornado pela função rand( ). Seu valor, considerando a linguagem de programação C, é 32767.