Universidade de Aveiro
2010
Departamento de Electrónica, Telecomunicações e Informática

Miguel Monsanto Pinheiro

Sistema computacional para o estudo da estrutura primária e redesenho de genes

Universidade de Aveiro
2010

Departamento de Electrónica, Telecomunicações e Informática

Miguel Monsanto Pinheiro

Sistema computacional para o estudo da estrutura primária e redesenho de genes

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Engenharia Informática, realizada sob a orientação científica do Doutor José Luís Oliveira e Doutor Manuel dos Santos, Professores associados da Universidade de Aveiro

o júri

presidente

Prof. Doutor Armando da Costa Duarte
Professor Catedrático do Departamento de Química da Universidade de Aveiro

Prof. Doutor Rui Pedro Sanches de Castro Lopes
Professor Coordenador do Departamento de Informática e Comunicações, do Instituto Politécnico de Bragança

Prof. Doutor Sara Alexandra Cordeiro Madeira,
Professora Auxiliar do Departamento de Engenharia Informática, do Instituto Superior Técnico.

Prof. Doutor Carlos Manuel Azevedo Costa,
Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

co-orientador

Prof. Doutor Manuel António da Silva Santos,
Professor Associado do Departamento de Biologia da Universidade de Aveiro.

orientador

Prof. Doutor José Luís Guimarães Oliveira,
Professor Associado do Departamento de Electrónica,Telecomunicações e Informática da Universidade de Aveiro

agradecimentos

A todos aqueles que de alguma forma contribuíram para a elaboração desta dissertação o meu muito obrigado:

José Luís Oliveira;
Manuel dos Santos;
Gabriela Moura;
Adelaide Valente;
Vera Afreixo;
Maria de Lurdes Monteiro;
Victor Monteiro;

Aos meus pais e companheira Teresa Monteiro. A eles dedico este trabalho.

palavras-chave

Contexto de codões, codões, análise residual, análise de grupos, bioinformática.

resumo

Um dos maiores avanços científicos do século XX foi o desenvolvimento de tecnologia que permite a sequenciação de genomas em larga escala. Contudo, a informação produzida pela sequenciação não explica por si só a sua estrutura primária, evolução e seu funcionamento. Para esse fim novas áreas como a biologia molecular, a genética e a bioinformática são usadas para estudar as diversas propriedades e funcionamento dos genomas.

Com este trabalho estamos particularmente interessados em perceber detalhadamente a descodificação do genoma efectuada no ribossoma e extrair as regras gerais através da análise da estrutura primária do genoma, nomeadamente o contexto de codões e a distribuição dos codões. Estas regras estão pouco estudadas e entendidas, não se sabendo se poderão ser obtidas através de estatística e ferramentas bioinfomáticas.

Os métodos tradicionais para estudar a distribuição dos codões no genoma e seu contexto não providenciam as ferramentas necessárias para estudar estas propriedades à escala genómica. As tabelas de contagens com as distribuições de codões, assim como métricas absolutas, estão actualmente disponíveis em bases de dados. Diversas aplicações para caracterizar as sequências genéticas estão também disponíveis. No entanto, outros tipos de abordagens a nível estatístico e outros métodos de visualização de informação estavam claramente em falta.

No presente trabalho foram desenvolvidos métodos matemáticos e computacionais para a análise do contexto de codões e também para identificar zonas onde as repetições de codões ocorrem. Novas formas de visualização de informação foram também desenvolvidas para permitir a interpretação da informação obtida.

As ferramentas estatísticas inseridas no modelo, como o *clustering*, análise residual, índices de adaptação dos codões revelaram-se importantes para caracterizar as sequências codificantes de alguns genomas.

O objectivo final é que a informação obtida permita identificar as regras gerais que governam o contexto de codões em qualquer genoma.

keywords

Codon context, codon, residual analysis, cluster analysis, bioinformatics software.

abstract

One of the major scientific advances of the twentieth Century was the sequencing of several genomes. However, the raw data alone produced from these genome sequencing efforts does not explain its primary structure, its evolution or its functioning. In order to do this, molecular biology, genetics and bioinformatics approaches have to be used for genome analysis.

We are particularly interested in understanding the general rules that govern accurate decoding by the ribosome and gene evolution through global analysis of genome primary structure features, such as codon context and usage. However, the general rules that govern codon usage and codon context remain largely elusive, raising the question: can those rules be unraveled using genomic scale approaches by combining bioinformatics, statistical and computer visualization tools?

Traditional methods, used for codon usage and context analysis, do not provide the tools to carry out detailed gene primary structure analysis at a genomic scale. Codon usage tables, using absolute metric, are available in public databases for any sequenced gene or genome and free-ware software for multivariate analysis of codon and amino acid usage is also readily available, however sophisticated statistical and data visualization tools are clearly missing.

We developed a model supported by mathematical and statistical tools for automated analysis of codon context and codon usage and also to find tri-nucleotide repeats within coding regions on a genomic scale. A sophisticated graphical application system has also been developed to help data visualization and interpretation.

The statistical tools incorporated in the system for data clustering, residual analysis and codon adaptation index determination will allow the obtention of global views of the important sequence features.

It is expected that the data obtained will allow the identification of general rules that govern codon context and codon usage in any genome.

Índice

Lista de Figuras

Lista de Tabelas

Lista de Acrónimos

A	Adenina
ALN	Clustal Alignment
API	Application Programming Interface
ATP	Adenosine triphosphate
BLAST	Basic Local Alignment Search Tool
BLAT	BLAST-like Alignment Tool
BLOSUM	Blocks Substitution Matrix
C	Citosina
CAI	Codon Adaptation Index
DDBJ	DNA Data Bank of Japan
DNA	Deoxyribonucleic acid
DRIPS	Defective ribosomal products
EBI	European Bioinformatics Institute
EMBL	European Molecular Biology Laboratory
ENc	Número de codões efectivo
FTP	File Transfer Protocol
GAP	Gap opening penalty
GC	Percentagem de Gs e Cs numa determinada sequência
GCG	Genetics Computer Group
GenBank	Genetic Sequence Databank
GEP	Gap Extend Penalty
GO	Gene Ontology
G	Guanina

HEG-DB	Highly Expressed Genes Databases
HGVBASE	Human Genome Variation Database
HSP	High-scoring Segment Pair
INSDC	International Nucleotide Sequence Database Collaboration
ISA	Iterative Signature Algorithm
IUPAC	International Union of Pure and Applied Chemistry
KEGG	Kyoto Encyclopaedia of Genes and Genomes
MBH	Mutual Best Hit
MDC	Mapas de diferença de contexto
MFC	Microsoft Foundation Class
MSF	Multiple Sequence Files
mpiBLAST	Message Passing Interface BLAST
NAR	Nucleic Acids Research
NIG	National Institute of Genetics
NIH	National Institutes of Health
NCBI	National Center for Biotechnology Information
ORF	Open Reading Frame
PAM	Percent Accepted Mutation
PIR	Protein Information Resource
PHP	Hypertext Pre-processor
PHYLIP	PHYLogeny Inference Package
PSI-BLAST	Position-Specific Iterative BLAST
REBASE	Restriction Enzyme Database
RNA	Ribonucleic acid
rRNA	Ribosomal RNA

SNP	Single Nucleotide Polymorphism
SSAHA	Sequence Search and Alignment by Hashing
SCANPS	Scan Protein Sequence
T	Tirosina
TrEMBL	Translated European Molecular Biology Laboratory
tRNA	Transfer RNA
U	Uracilo
UniParc	Universal Protein Resource Archive
UniProt	Universal Protein Resource
UniRef	UniProt Reference Clusters
WGS	Whole Genome ShotGun

Capítulo 1

1 Introdução

Com o desenvolvimento das tecnologias de descodificação de genomas, assim como o forte investimento em programas sequenciação abrangendo os mais variados organismos, abriu-se uma oportunidade única para estudar o funcionamento e evolução dos seres vivos. Desde a primeira sequenciação completa de um organismo em 1977 [1], muito já se progrediu até ao presente momento. Actualmente existem centenas de genomas sequenciados, e com o aparecimento de novas técnicas de sequenciação esse número tenderá a aumentar a um ritmo elevado. Contudo, o grande volume de informação contido nos genomas criou novos desafios e questões, cuja resolução requer o desenvolvimento de metodologias matemáticas e ferramentas bioinformáticas capazes de lidar com grandes volumes de informação.

Os organismos presentes no nosso planeta têm uma característica única: todo o seu funcionamento é regulado por informação que está presente dentro da própria célula, o ácido desoxirribonucleico, mais conhecido por *Deoxyribonucleic acid* (DNA). Portanto, o DNA através de codificação quaternária, representada por quatro letras (A, C, T, G) conhecidas como os nucleótidos, consegue fornecer a informação necessária para manter a célula vida, mesmo em condições adversas. Devido aos mecanismos intracelulares serem de tal forma complexos é necessário subdividir os vários processos para começar compreender o seu funcionamento. Um dos mecanismos importantes, que ocorre dentro das células, é a conversão da informação presente no DNA para estruturas tridimensionais conhecidas como proteínas. A tradução pressupõe a combinação de três letras que formam

um codão (AAA, AAC, AAG, AAT, ACA, …). Não é necessária a totalidade da sequência genética presente no DNA para o processo de tradução, somente partes específicas da sequência são traduzidas, conhecidas por genes. Os genes, são pequenos pedaços do DNA do genoma, que vão servir de fonte de informação para construir as proteínas.

Como em qualquer mecanismo existe a probabilidade de erro. Vários tipos de erros foram já identificados como a falha de sincronização, paragem súbita do mecanismo de tradução e mal formação da proteína resultante de uma má tradução. No entanto, esta é uma área onde o conhecimento é limitado e onde novas descobertas poderão ter enorme impacto na sociedade pois ajuda a compreender processos como o envelhecimento, o cancro e outras doenças.

1.1 Enquadramento

O código genético é formado por arranjos com repetições dos elementos ATGC escolhendo 3, resultando em 64 elementos que são usados para identificar os 20 aminoácidos, elementos base das proteínas. Cada codão é reconhecido por um anti-codão, o qual vai transportar um aminoácido para a construção da proteína. Esta conversão de 64 para 20 apresenta redundância o que implica que um determinado aminoácido possa ser traduzido a partir de mais do que um codão (codões sinónimos). Por exemplo os codões TTA, TTG, CCT, CCA, CCC e CCG codificam o mesmo aminoácido, a leucina. Contudo, a escolha entre diferentes codões sinónimos não altera a estrutura do produto final, a proteína. É também sabido que os codões sinónimos não estão distribuídos aleatoriamente. A não aleatoriadade do uso de codões e a variação desses valores entre espécies sugere que existe alguma preferência e rejeição entre codões sinónimos, segundo demonstram diversos estudos realizados em sequências genéticas [2, 3]. Por outro lado, codões utilizados frequentemente são descodificados por anti-codões cuja concentração celular é elevada, sugerindo a existência de uma relação entre a frequência de utilização dos codões nos genes e a abundância dos anti-codões que os descodificam. Para além desta restrição, a análise dos codões vizinhos de um determinado codão (contexto do codão) sugere que os codões influenciam os seus vizinhos a montante e a jusante.

Estas observações levantam a hipótese de que o contexto dos codões influencia a velocidade e fidelidade de tradução [4, 5]. Considerando que a qualidade da tradução é

critica para assegurar a correcta transferência de informação do DNA para as proteínas, conhecer melhor este processo é imprescindível para revelar as regras que governam o funcionamento das células. Mais importante ainda, conhecendo-se as regras que regem o contexto de codões, poderão diminuir-se os erros na descodificação genética e melhorar a expressão das proteínas [6]. Quer isto dizer, poderemos obter um maior número de cópias de determinada proteína para o mesmo gene.

As soluções existentes para quantificar os codões e seu contexto não proporcionam os meios necessários para elucidar sobre as relações existentes, nem permitem fazer um estudo profundo sobre a estrutura primária do DNA. Tabelas de contagem de codões, usando contagens absolutas, ou médias ponderadas, já podem ser encontradas na Internet. Aplicações para calcular os níveis de codões dos genomas e outras estatísticas são também frequentemente utilizadas. Contudo, o estudo do contexto de codões é um assunto que se encontra por explorar.

1.2 Objectivos

O objectivo deste trabalho é a criação de métodos computacionais para avaliar o contexto dos codões, contemplando também a junção de outros índices actualmente já utilizados para caracterizar as sequências genéticas. Assim, será possível integrar variada informação que caracteriza as regiões codificantes do genoma, cruzando-a com nova informação obtida através do modelo proposto.

As ferramentas criadas deverão permitir a extracção de regras em espécies específicas ou entre espécies, permitindo também a obtenção de regras transversais aos três reinos da vida existentes, nomeadamente as eubactérias, arqueas e eucarióticas.

Esses modelos serão integrados numa aplicação gráfica possibilitando o fácil acesso à informação obtida, assim como a sua extracção em simples ficheiros de texto. A aplicação deverá ser construída de forma a proporcionar uma utilização intuitiva apoiada em paradigmas familiares.

Posteriormente, deverá ser possível reflectir os dados obtidos na alteração das sequências genéticas de forma a reduzir as áreas onde a propensão ao erro de tradução será mais

elevada. Com esta abordagem fica aberto o caminho para testar os dados obtidos por meios informáticos em laboratório validando assim o modelo matemático.

1.3 Estrutura da dissertação

Este documento está dividido em sete capítulos que se encontram organizados em três partes distintas. A primeira parte, dedica-se ao trabalho de síntese, incluindo a apresentação do documento e exposição do estado da arte em análise de sequências genómicas. A segunda parte do documento dedica-se à descrição do contributo do trabalho, i.e., o modelo proposto, a apresentação da aplicação desenvolvida e alguns resultados obtidos. Por último, teremos uma parte destinada às conclusões. De seguida descreve-se sucintamente o conteúdo de cada um dos capítulos seguintes.

O capítulo dois introduz os conceitos principais associados à organização de uma célula e os princípios básicos do seu funcionamento. São realçadas as características mais relevantes para este trabalho focando mais o processo da tradução dos genes nas proteínas.

No capítulo três apresentam-se as várias ferramentas disponíveis para estudar sequências genéticas. Actualmente, existem muitas ferramentas informáticas de análise genética, sendo utilizadas em inúmeros estudos. No entanto, iremos realçar as ferramentas que estão directamente relacionadas com o trabalho em causa, assim como as fontes de dados necessárias e seus formatos.

A proposta do modelo de caracterização de sequências genéticas é apresentada no quarto capítulo. Começa-se por apresentar a proposta de como extrair informação sobre o contexto de codões, assim como a sua comparação entre diferentes espécies. Será também apresentado um modelo que possibilita cruzar os contextos entre sequências homólogas de diferentes organismos. Para finalizar, será descrito um algoritmo que permitirá alterar a sequência genética de forma a reflectir os dados obtidos através do modelo proposto.

O capítulo cinco contém a descrição da aplicação desenvolvida segundo o modelo proposto. Será apresentada globalmente a sua arquitectura nomeadamente em forma de diagrama de classes, seguido de uma descrição mais pormenorizada das diversas funcionalidades que a aplicação oferece.

A exposição de alguns resultados obtidos com este trabalho é realizada no capítulo seis. Serão apresentadas algumas regras de descodificação para três organismos distintos e ainda regras transversais aos três reinos da vida. Um pequeno estudo sobre análise de *biclustering* é também exposto.

Por último, o capítulo sete resume os resultados e evidencia os ganhos que foram alcançados. São também descritas algumas ideias e direcções para num trabalho futuro.

Capítulo 2

2 Fundamentos sobre biologia celular

2.1 Introdução

Os processos biológicos envolvidos nas formas de vida são de tal forma complexos, que se torna necessário isolar os diversos processos de forma a compreendê-los melhor.

São vários os processos que ocorrem em simultâneo dentro de uma célula, mas sem dúvida alguma, os mais importantes são os processos que estão envolvidos com a molécula que transporta toda a codificação genética, o ácido desoxirribonucleico (DNA - *Deoxyribonucleic acid*). Esta molécula contém todo o código responsável pelo seu funcionamento, bastando surgir uma pequena alteração num local importante para pôr em risco a sobrevivência ou para assumir um papel para a qual não estava programada. As alterações no DNA são constantes, impostas por vários elementos presentes no meio ambiente ou simplesmente por falhas nos processos envolvidos no processamento interno do código genético. No entanto, as alterações no DNA são também responsáveis pela evolução, transformando-se por vezes em características que trazem vantagem competitiva transmitindo-se à sua descendência.

No presente capítulo iremos apresentar uma breve síntese sobre alguns conceitos de biologia celular necessária para a compreensão dos mecanismos envolvidos no funcionamento da célula. Será descrita uma visão geral sobre os processos principais que envolvem o DNA nomeadamente aqueles que são responsáveis pelos erros de tradução e

que conduzem ao aparecimento de doenças. Os problemas aqui levantados servirão igualmente de motivação para as propostas que serão feitas nesta tese.

2.2 Biologia molecular

A célula é a mais pequena porção do organismo capaz de vida independente [7]. Todos os seres vivos são compostos de uma ou mais células. Alguns seres microscópicos são compostos por uma única célula, como as bactérias, mas os animais e plantas são formados por muitos milhões de células organizadas em tecidos e órgãos capazes de interagir entre si, de forma a cooperarem para um fim comum, a sobrevivência.

Apesar da grande variedade de seres vivos, as células que os compõem têm muito em comum. Existem dois tipos principais de células: as procariotas e as eucariotas. As procariotas possuem um único compartimento, encerrado numa membrana, e não possuem quaisquer estruturas ou compartimentos bem definidos no citoplasma, material gelatinoso contido no interior da membrana plasmática (celular). Exemplos de organismos procariotas são as eubactérias, comummente chamadas bactérias, e as arqueobactérias também conhecidas por arqueas. Quanto às células eucariotas, características das plantas, fungos e animais, contêm compartimentos bem definidos especializados em funções vitais para a célula (Figura 1). Estes compartimentos são envolvidos por membranas e têm o nome de organitos, criando uma série de micro ambientes dentro da célula, nos quais as reacções químicas se podem processar com a máxima eficiência diminuindo assim a probabilidade de erro [8].

O organito mais importante das células eucariotas é o núcleo, o qual encerra o material genético. É o núcleo que contém informação codificada em moléculas de ácido desoxirribonucleico (DNA), sendo esta informação responsável por regular a actividade da célula, assegurar a sua duplicação e transmitindo-se à sua descendência. No caso das células procariotas, devido à inexistência de núcleo, o seu DNA vagueia dentro da célula sem estar encerrado num local específico.

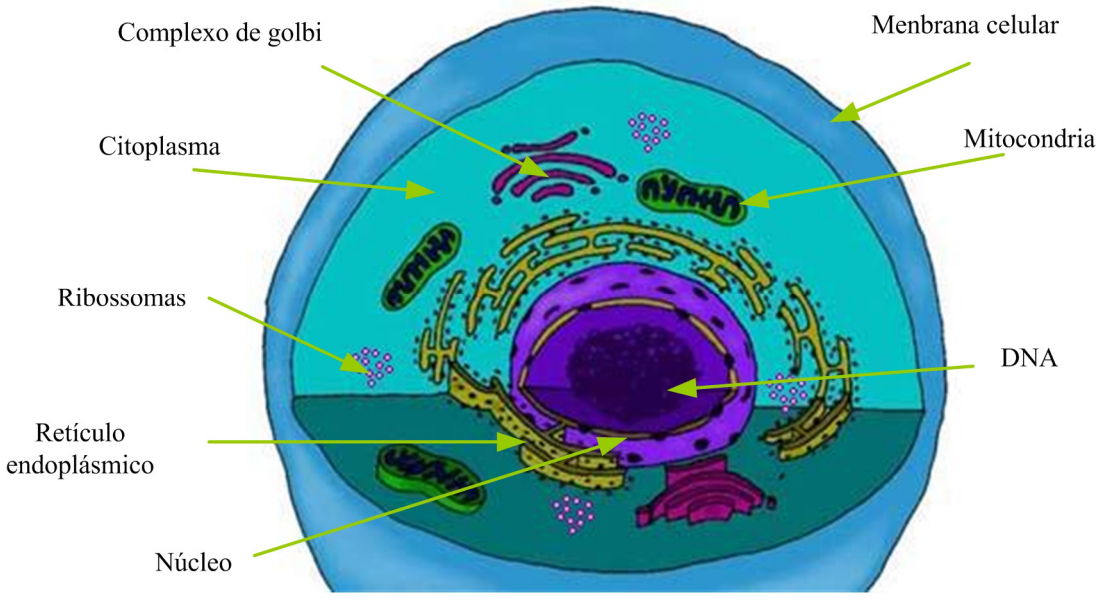

Figura 1 – Estrutura de uma célula eucariota.

Outros organitos semi-autónomos das células eucariotas são as mitocôndrias, responsáveis pela produção de energia durante a respiração aeróbica (processo responsável pela formação da molécula trifosfato de adenosina, ATP, a qual contém uma ligação de fosfato de alta energia). Basicamente, transformam a energia dos alimentos em energia biologicamente útil. A mitocôndria contém o seu próprio código genético e novas mitocôndrias surgem no interior das células pela própria divisão das mitocôndrias existentes [9]. Na reprodução sexuada o DNA presente no núcleo resulta na recombinação do DNA proveniente do lado masculino e do lado feminino, mas no caso da mitocôndria, o DNA vem sempre do lado feminino. Baseado neste facto, investigadores da Universidade da Califórnia concluíram que todos os humanos eram descendentes de uma única mulher que viveu na África há cerca de 150 mil anos, que passou a ser chamada de Eva Mitocondrial [10]. Devido a esta particularidade a clonagem é particularmente difícil, porque do ponto de vista genético teríamos de ter um organismo que tivesse o mesmo DNA nuclear e mitocondrial. Ao se clonar o núcleo, as mitocôndrias provêm do óvulo feminino, não resultando uma clonagem total da célula. No entanto, as mitocôndrias, apesar de terem um papel importante no metabolismo energético da célula, têm pouca influência nas características genéticas, já que mais de 99% do nosso DNA está no núcleo.

O ribossoma, outro dos organitos presentes na célula, é a entidade responsável pela síntese de proteínas. Cada ribossoma é composto por duas sub-unidades, a grande e a pequena, tema que será aprofundado nesta tese.

Existem outros organitos, mas não têm relevância para o estudo que se segue.

2.2.1 As bases da vida

O genoma é o conjunto de todo o DNA presente na célula, contento todos os genes presentes num organismo e possuindo toda a informação necessária para expressar proteínas necessárias ao funcionamento da célula. Ao conjunto de todas as proteínas dá-se o nome de proteoma. O genoma está organizado em cromossomas, podendo os organismos ter mais do que um cromossoma. Os humanos possuem 24 pares de cromossomas distintos, enquanto as bactérias possuem normalmente um cromossoma em forma de círculo. Cada cromossoma é formado por um par de moléculas de DNA, ligadas por quatro bases distintas: A, C, G e T (adenina, citosina, guanina e timina), também chamados nucleótidos (Figura 2) [11].

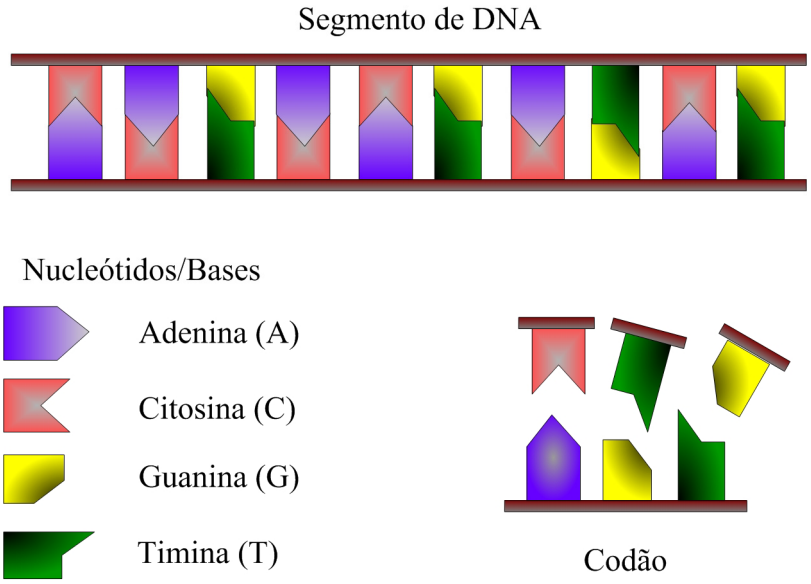

Figura 2 – Exemplo de um segmento de DNA. A adenina emparelha com a citosina e a guanina emparelha com a timina.

Os cromossomas têm partes codificantes e partes não codificantes. Os genes são as partes codificantes contendo a informação para produzir proteínas ou cadeias polipeptídicas (conjunto de aminoácidos justapostos). O gene especifica a sequência de aminoácidos de

uma proteína ou de cadeias polipeptídicas (Figura 4), ou seja, os genes contêm a informação que define a ordem dos aminoácidos, que se emparelham lado a lado, para formar as proteínas.

De forma a fazer correspondência para aminoácidos, as bases (A, C, T, G) agrupam-se três a três, formando codões. Cada codão corresponde a um aminoácido, mas um aminoácido pode corresponder a mais do que um codão existindo assim uma relação de um para muitos. O ribossoma lê conjuntos de 3 letras e não uma letra de cada vez. Ou seja, o código genético é organizado em arranjos com repetições de4 letras escolhendo 3. Por exemplo, a sequência AACGGCCCACUG é lida como AAC-GGC-CCA-CUG, sendo que o tripleto AAC codifica o aminoácido asparagina, GGC glicina, CCA prolina e CUG leucina. As quatro bases combinadas, três a três, dão 64 codões possíveis, mais do que o necessário para codificar os 20 aminoácidos diferentes. Existem mais de 200 aminoácidos presentes na natureza, mas normalmente são necessários apenas 20 para sintetizar as proteínas necessárias ao metabolismo das células [11].

Diz-se que o código genético é "degenerado" devido aos aminoácidos serem traduzidos por mais do que um codão (Figura 3), definindo-se como codões sinónimos os codões que codificam o mesmo aminoácido. O nome dos aminoácidos assim como as suas abreviações estão descriminados no anexo 8.1.

Alguns codões desempenham papéis especiais, como, por exemplo, o codão ATG indica o início do gene, embora este codão possa também aparecer em outras posições do gene. No entanto, existem outros codões para indicar o início do gene mas são específicos de determinados organismos a que pertencem. Os codões TAA, TGA e TAG são utilizados para assinalar o fim de um gene, não podendo aparecer em outras posições do gene. São chamados codões de terminação.

O código genético apresentado na tabela da Figura 3 é designado como código padrão, porque a maioria dos organismos seguem esta correspondência. No entanto, existem excepções. Por exemplo, para alguns organismos o codão TGA, que o código padrão corresponde a um codão de terminação, corresponde ao aminoácido cisteína. Existem tabelas fornecidas pelo *National Center for Biotechnology Information* (NCBI) contendo todas as alterações até ao momento identificadas em relação à tabela padrão [12],

revelando-se de grande importância para investigadores que pretendem estudar os processos em torno da descodificação de sequências genéticas.

TTT TTC Phe	TCT TCC TCA TCG Ser	TAT TAC Tyr	TGT TGC Cys
TTA TTG Leu		TAA TAG Stop	TGA Stop
			TGG Trp
CTT CTC CTA CTG Leu	CCT CCC CCA CCG Pro	CAT CAC His	CGT CGC CGA CGG Arg
		CAA CAG Gln	
ATT ATC ATA Ile	ACT ACC ACA ACG Thr	AAT AAC Asn	AGT AGC Ser
ATG Met		AAA AAG Lys	AGA AGG Arg
GTT GTC GTA GTG Val	GCT GCC GCA GCG Ala	GAT GAC Asp	GGT GGC GGA GGG Gly
		GAA GAG Glu	

Figura 3 - Tabela que contém a correspondência padrão entre codões e aminoácidos.

2.2.2 Processo de tradução

O gene é traduzido em proteína por um processo que envolve duas fases, a transcrição e a tradução, seguindo sempre o sentido de transcrição para a tradução (Figura 4). No processo de transcrição, o DNA do gene é utilizado como molde para a síntese de uma cadeia complementar. Posteriormente, os intrões, sequências de bases não codificantes presentes no gene, são removidos da cadeia por meio de enzimas num processo chamado *splicing*, restando apenas os chamados exões (sequências que codificam a proteína). O processo de *splicing* só existe nos organismos eucariotas, não existindo nos organismos procariotas. Posteriormente obtém-se uma molécula de ácido ribonucleico mensageiro (mRNA) que é transportada para o exterior do núcleo através dos seus poros, dando início ao processo de tradução de proteínas. O mRNA também possui as quatro bases, mas com a timina (T)

substituída pelo uracilo (U). A molécula mRNA contém o código para expressar uma proteína específica.

Nos organismos procariotas não existem intrões reduzindo o gene a um único exão, tornando o processo de processamento do RNA mais simples do que nos organismos eucariotas. No caso específico das células humanas, por exemplo, o gene que codifica uma determinada proteína no fígado pode não produzir a mesma proteína no músculo, devido à possibilidade de *splicing* alternativo, possibilitando diferentes combinações de exões.

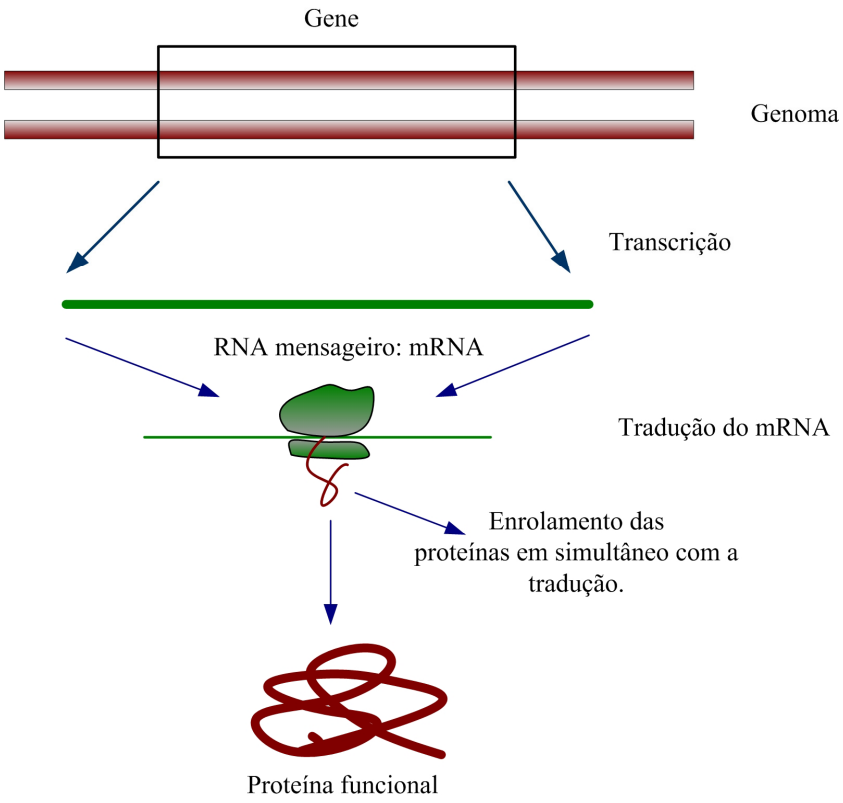

Figura 4 – Diagrama ilustrando o processo de tradução de um gene numa proteína.

O processo de tradução é realizado no citoplasma, por uma maquinaria celular específica chamada de ribossoma (Figura 5).

Cada ribossoma é formado por duas subunidades, a pequena e a grande, normalmente conhecidas como as subunidades 30S e 50S, respectivamente, nas bactérias. Estas subunidades são muito complexas contendo na sua estrutura RNAs ribossomais (rRNA) e várias proteínas. Quando as duas subunidades se combinam para formar o ribossoma, proporcionam o meio certo para a síntese das proteínas, disponibilizando os locais específicos de ligação para os portadores de aminoácidos e para a cadeia de mRNA. Nas

bactérias, os ribossomas iniciam o processo de tradução ligando-se à sequência Shine-Dalgarno [13] que está localizada a 6-7 nucleótidos a montante do codão de iniciação ATG presente no gene, permitindo assim a sincronização com o codão iniciação. A sequência Shine-Dalgarno é composta pelo padrão AGGAGG nos organismos procariotas. Nos organismos eucariotas, não existe esta sequência mas o codão de iniciação está num contexto específico que é importante para o seu reconhecimento pelo ribossoma.

Os aminoácidos são transportados para o ribossoma por uma outra forma de RNA, o RNA de transferência (tRNA). Há pelo menos um tipo de molécula de tRNA para transportar os vinte diferentes aminoácidos presentes na célula. Cada molécula de tRNA apresenta uma sequência de três bases, chamada de anti-codão, que emparelham com as bases dos codões de mRNA para codificar o aminoácido transportado por cada tRNA. Assim, o tRNA funciona como um tradutor, transportando os aminoácidos correctos definidos pelos codões presentes no mRNA.

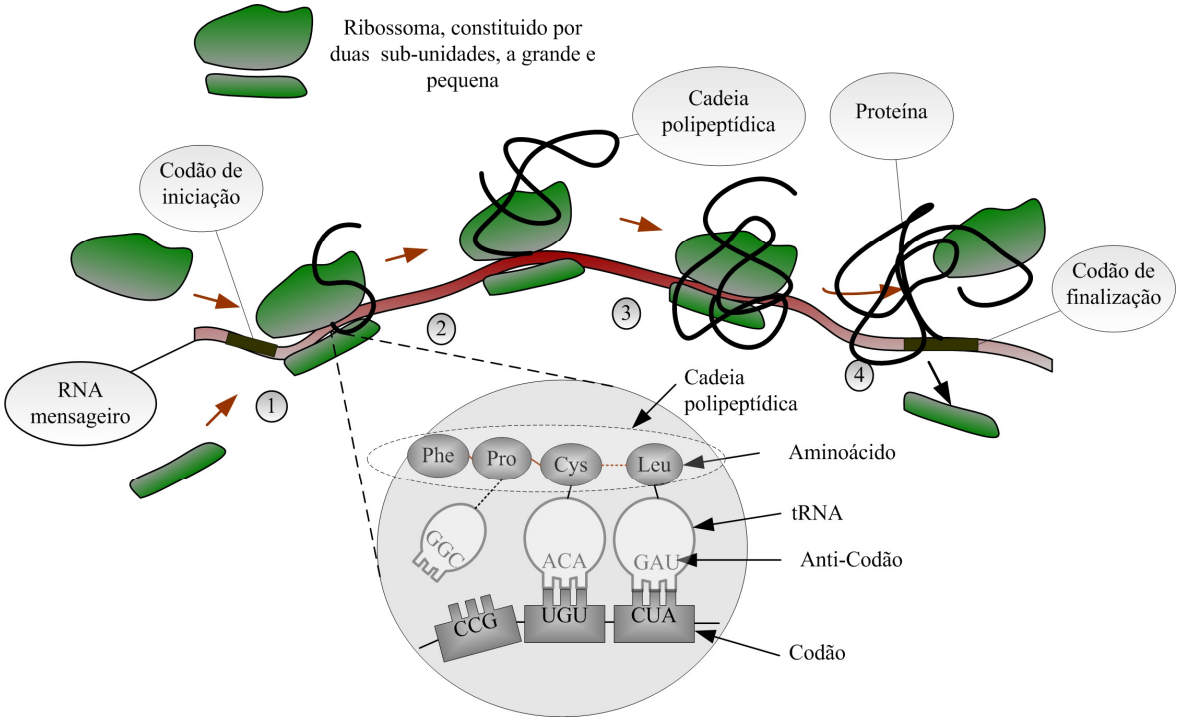

Figura 5 - Processo de tradução do RNA mensageiro envolvendo o RNA de transferência construindo posteriormente cadeias polipeptídicas.

À medida que cada aminoácido é trazido para o ribossoma pelo seu tRNA, o centro catalítico do ribossoma, chamado centro da peptidil-transferase, promove a formação da ligação peptídica entre os aminoácidos, construindo-se assim uma cadeia polipeptídica. A

tradução prossegue até se chegar a um codão de terminação; nessa altura, a cadeia polipeptídica é libertada do ribossoma e as duas subunidades separam-se. Uma molécula de mRNA pode ter mais de um ribossoma a traduzi-lo em simultâneo, formando assim mais do que uma cadeia polipeptídica. As cadeias polipeptídicas recém formadas enrolam-se sobre si próprias formando proteínas com estruturas terciárias funcionais.

As sequências que formam o RNA de transferência (tRNA) têm um comprimento típico entre 75 a 90 nucleótidos e são codificados por genes específicos chamados tDNA. Os tRNAs têm uma estrutura secundária em forma de folha de trevo e uma estrutura terciária em forma de L invertido (Figura 6).

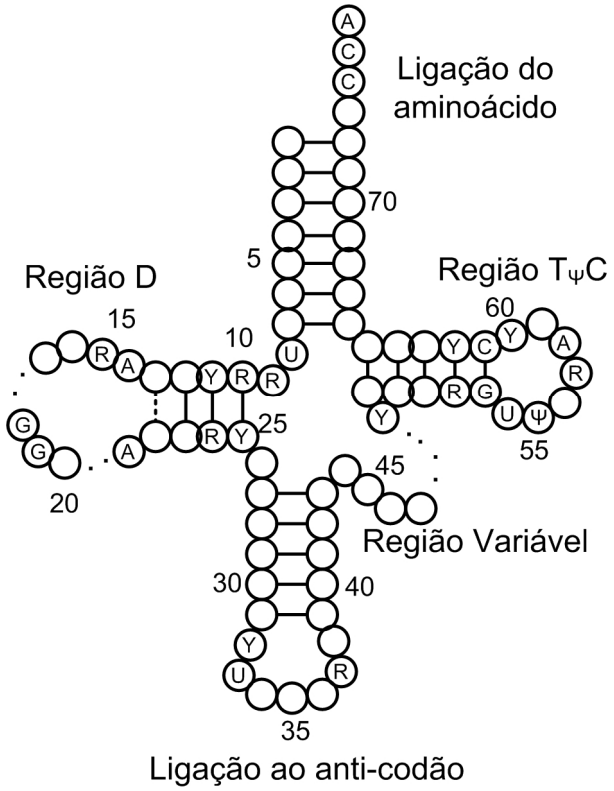

Figura 6 – Estrutura secundária de um tRNA. A letra R corresponde aos nucleótidos A ou G e a Y corresponde aos nucleótidos C ou T.

O *anticodon loop* contém o anti-codão que lê o codão do mRNA. Para cada codão deveria existir um tRNA diferente, uma vez que pelas regras do Watson-Crick [14] (base A liga-se ao T e o C ao G) cada codão teria um anti-codão associado. Mas o processo de leitura (descodificação) dos codões pelos anti-codões utiliza regras não convencionais de emparelhamento das bases, chamadas regras de *wobble*. Por exemplo, o par G-U é frequente neste tipo de emparelhamento [15]. As regras de *wobble* reduzem o número de

tRNAs de 61 para 42. Em alguns casos em que o tamanho do genoma foi substancialmente reduzido por selecção natural, só há 22 anti-codões para descodificar os 61 codões. Nestes casos, os organismos utilizam regras muito flexíveis de descodificação chamadas *Extended Wobble* [16]. Um caso clássico são as mitocôndrias que têm um genoma com um tamanho mínimo. Com base neste conhecimento, foi desenvolvido um estudo para identificar as regras que regem as substituições dos tRNA nos eucariotas e procariotas [17].

Todos estes passos têm um objectivo muito específico: a construção de proteínas. As proteínas regulam os mais variados processos que ocorrem dentro da célula, estando directamente implicadas no funcionamento das células. Se algum problema ocorre nos processos de transcrição ou tradução pode levar à síntese de proteínas com estrutura diferente da esperada, pondo em causa o bom funcionamento da célula ou mesmo de um conjunto de células.

2.2.3 Proteínas: erros de síntese

Existem vários tipos de proteínas que podem ser agrupadas de acordo com as suas funções celulares (Tabela 1). O tamanho de uma proteína pode variar entre 30 a 10.000 aminoácidos, mas normalmente está compreendido entre 50 a 2000 aminoácidos [8].

O processo de tradução dos genes para formar proteínas assume assim um papel central no processo da vida, sendo, por esta razão, objecto de intenso estudo. Note-se, que a perda de controlo da fidelidade de replicação/tradução nos organismos pode ser catastrófico, criando um completo caos na célula, conduzindo-a a uma morte prematura ou fazendo-a assumir outra função para a qual não estava definida.

No entanto, erros de tradução existem e fazem parte do processo natural de síntese das proteínas. Por exemplo, a bactéria *Escherichia coli* sintetiza as suas proteínas com um erro de 10^{-3} a 10^{-4} por cada codão traduzido, em condições normais de crescimento [19]. Erros como falha de sincronismo de leitura e falha de reconhecimento do codão de terminação estão compreendidos nos limites entre $3x10^{-4}$ a 10^{-5} e 10^{-3} a 10^{-6}, respectivamente [20]. Quando submetidos a condições de stress, nomeadamente falta de aminoácidos, estes erros aumentam consideravelmente [21], evidenciando que a taxa de erros na natureza será bem maior do que nas condições estáveis existentes no laboratório. Além disso, 30% das proteínas recém sintetizadas em células de mamíferos, nomeadamente HeLa, linfáticas e

dendríticas são defeituosas, *Defective ribosomal products* (DRIPS), que surgem a partir de erros, como a falha de sincronização da grelha de leitura ou falha de leitura pelo ribossoma em zonas onde a tradução se faz a uma velocidade mais lenta do que a normal [22]. Como a célula utiliza 45% de energia ATP para sintetizar proteínas, 30% de DRIPS representa um consumo energético de 11% [22].

Tabela 1 - Lista dos vários grupos de proteínas [18].

Grupo	Função	Exemplos
Enzimas	Funções catalíticas.	As células do fígado contêm milhares de enzimas, cada uma responsável para catalizar determinada reacção. Transformar o aminoácido de fenilalanina em tirosina.
Estruturais	Providencia mecanismos de suporte para células e tecidos.	As células exteriores como pele, cabelo, etc.; células interiores, como os tendões.
Transporte	Transporta pequenas moléculas ou iões.	No sangue a hemoglobina transporta o sangue, *aserun albumin* transporta o lípidos, entre outros.
Motoras	Gera movimento nas células e tecidos.	O *myosin* providencia a força para os movimentos nos animais.
Armazenamento	Armazena pequenas moléculas e iões.	O ferro é armazenado no fígado através da proteína *ferritin*; *ovalbumin* na clara do ovo é usado como fonte de aminoácidos para o desenvolvimento de aves embrionárias.
Sinais	Transporta sinais de célula para célula.	Muitas das hormonas que regulam os factores de crescimento são proteínas. Por exemplo, a insulina é uma proteína que controla os níveis de glucose no sangue.
Regulação genética	Ligada ao DNA, liga ou desliga a expressão de determinado gene.	O gene repressor *lacl* silencia o gene que está envolvido no ciclo da lactose.
Funções especiais	Muito variadas.	Os organismos desenvolvem proteínas com aplicações muito específicas. Por exemplo, a *antifreeze protein* dos peixes do Árctico e Antárctico protegem o seu sangue do congelamento.

Os mecanismos acima mencionados são ainda mal conhecidos, contudo o seu estudo pode trazer vantagens significativas na síntese de proteínas recombinantes, área emergente da biologia sintética. As proteínas recombinantes têm aplicações biotecnológicas e terapêuticas importantes que vão desde a produção de insulina, interferão gama até a proteínas com novas funções [23].

Apesar do enorme progresso na produção de proteínas recombinantes, há certas proteínas que não se conseguem produzir em organismos heterólogos [24]. Este problema poderá ser somente uma falta de adaptação do gene heterólogo à maquinaria de síntese proteica do organismo hospedeiro, existindo a possibilidade de alteração do gene, corrigindo assim a sequência de codões mas mantendo a sequência de aminoácidos da proteína. Tal é possível porque o código genético é degenerado. Tais alterações na sequência de codões permitem aumentar significativamente a produção de proteínas recombinantes [24]. Outros problemas poderão estar associados a toxicidade da proteína recombinante para o hospedeiro, dificultando a sua solução.

Uma das questões em aberto, relativamente ao erro de descodificação do mRNA pelo ribossoma, é se ele ocorre aleatoriamente ou se ocorre em zonas específicas do mRNA. Se o erro ocorrer preferencialmente em zonas localizadas, será possível identificá-lo?

2.3 Análise da estrutura primária dos genes

A descodificação dos genomas abriu caminhos para a compreensão da estrutura primária, permitindo compreender e identificar as forças evolutivas que têm moldado a sua evolução. Existem actualmente várias formas para caracterizar e analisar estruturas primárias nas zonas codificantes. Esta tese irá focar-se no estudo do contexto de codões e na sua importância para a estrutura primária dos genes.

2.3.1 Caracterização das zonas codificantes

No código genético existem 61 (mais 3 codões de terminação) codões disponíveis para os 20 aminoácidos do código genético. Devido à existência de redundância, diferentes organismos revelam especial preferência por um dos vários codões possíveis que codificam o mesmo aminoácido, existindo padrões distintos de espécie para espécie [25]. As razões para estas preferências continuam a ser debatidas no campo da evolução molecular, não existindo até ao momento uma explicação clara. No entanto, vários factores foram já sugeridos na tentativa de explicar o aparecimento do *codon usage*, conceito que reflecte a preferência de um codão em detrimento de outros possíveis para o mesmo aminoácido [3]. A eficiência no processo de tradução, fidelidade, tendência imposta pelas mutações [26], e diminuição de erro no processo de tradução [27, 28] são as razões mais

comuns para explicar a diferença entre o número de codões que traduzem o mesmo aminoácido. Como resultado destes factores, espera-se que esta tendência venha a ser mais pronunciada nos genes que são traduzidos com maior frequência.

Existem duas formas distintas de construir índices para o *codon usage*: i) recorrendo a todos os genes presentes no organismo; ii) seleccionando um grupo de genes, normalmente os mais expressos, tentando eliminar o ruído introduzido pelos genes de expressão reduzida. Estes índices reduzem a não distribuição normal dos codões a simples tabelas, simplificando a sua aplicação.

No campo da bioinformática, muitos índices foram propostos e usados para analisar *codon usage* [29]:

- *Relative Synonymous Codon Usage* (RSCU) - índice calculado através do rácio entre a frequência de um codão e o seu valor esperado se a sua distribuição fosse uniforme entre os seus sinónimos. Valores próximos de 1 indicam que determinado codão tem uso pleno entre os seus sinónimos. Existe uma base de dados pública onde é possível calcular os RSCU de vários organismos [30]. A equação (1) traduz o cálculo do RSCU com os seguintes parâmetros: *xi* - a contagem para o codão *i* numa determinada sequência; *Sk* - o número de codões sinónimos para o aminoácido *k*; *Cl* - contém o número de codões presente para o aminoácido *k*.

$$W_i = \frac{x_i}{\frac{1}{S_k} \sum_{Cl(j)=k} x_j} \quad (1)$$

- *Codon Adaptation Index* (CAI) - índice que mede a adaptação de um determinado gene mediante o *codon usage* obtido através de genes de elevada expressão, prevendo assim o nível de expressão genética para o gene em análise. Não é mais do que a média geométrica do índice RSCU para todos os codões presentes no gene. O RSCU, neste caso, é obtido através de genes considerados de elevada expressão no organismo em análise. Os valores obtidos estão no intervalo [0..1], em que o zero indica um gene de baixa expressão e o máximo um gene de elevada expressão [31]. O CAI para uma determinada sequência é calculado com a equação seguinte: L - o comprimento da sequência em codões; W_i - o RSCU para o codão i.

$$CAI = \left(\prod_{i=1}^{L} W_i \right)^{\frac{1}{L}} \quad (2)$$

- Frequência óptima de codões (Fop) - índice que mede o rácio entre os codões óptimos e os seus sinónimos. Os codões considerados óptimos são os que contêm mais cópias de anti-codões disponíveis entre os seus sinónimos. O índice varia entre 1, onde são usados todos os codões considerados óptimos, e 0 onde não são usados codões considerados óptimos [32].

$$Fop = \frac{N_{optimal\_codons}}{N_{sysnonymou\_codons}} \quad (3)$$

- Número efectivo de codões (N_c) - contabiliza os codões efectivamente presentes num determinado gene, não tendo em conta a sua frequência [33].
- Codão raro - codão cuja existência está num limite abaixo de cinco para mil contabilizando todos os codões presentes no organismo.

Outros índices, relacionados também com o *codon usage*, são os níveis de GC presentes nas zonas codificantes. Estes índices contabilizam a percentagem de GC dos genes em relação a todos os nucleótidos presentes.

2.4 Contexto de codões

A frequência dos codões mostra que cada organismo favorece a utilização de determinados codões e reprime o uso de outros. Os nucleótidos que flanqueiam os codões também influenciam o *codon usage*, com uma forte influência no primeiro e último codão. Assim sendo, os codões influenciam os seus vizinhos a montante e a jusante, originando assim o contexto entre codões. O nucleótido na terceira posição do primeiro codão e o primeiro nucleótido do segundo codão ($N_1N_2N_3$ $N_1N_2N_3$) influenciam-se mutuamente, originando o contexto N_3-N_1 [34, 35]. Contrariamente ao *codon usage*, as forças que modelam o contexto de codões, com a excepção dos codões de iniciação e terminação [36], estão pouco compreendidas. Os poucos estudos realizados até ao momento comprovam que o contexto de codões está relacionado directamente com os erros de descodificação [37-39].

Estas observações levantam a hipótese de que o contexto dos codões influencia a velocidade e fidelidade de descodificação das zonas codificantes pelos anti-codões no ribossoma. No entanto, ainda não é claro se o contexto de codões é utilizado para regular a velocidade de tradução das zonas codificantes, se influencia o ribossoma durante o processo de tradução e como os genes com prevalência de maus contextos são traduzidos sob condições stress. Considerando a necessidade de alta fidelidade no processo de tradução, compreender estas regras é de uma importância extrema, pois irão ser compreendidos os constrangimentos impostos aos genes pela maquinaria de tradução, durante o seu processo evolutivo. Compreendendo as limitações do processo de tradução, imposto por cada organismo, será possível redesenhar genes com o objectivo de os expressar em organismos heterólogos. Assim sendo, uma análise do contexto à escala genómica poderá evidenciar leis gerais que governam a fidelidade de descodificação do código genético.

Concluindo, o contexto tem em conta a relação entre codões algo que não é proporcionado pela análise do *codon usage,* que só contempla os codões de forma isolada negligenciando as suas dependências.

2.5 Evolução molecular

Todos os seres vivos presentes no nosso planeta partilham um conjunto significativo de processos moleculares, incluindo a forma como as sequências estão codificadas no DNA e a transcrição para o RNA, processo de tradução que envolve a leitura do RNA mensageiro pelo ribossoma. Muitas características bioquímicas, incluindo variadas enzimas e proteínas são muito similares na sua sequência de aminoácidos, estrutura tridimensional e função enzimática. Por exemplo, o proteoma (conjunto de todas as proteínas de um organismo) da mosca da fruta *Drosophila melanogaster* tem 13.601 proteínas podendo ser agrupadas em 8.065 famílias devido à partilha de sequências idênticas de aminoácidos. O proteoma do verme *Caenorhabditis elegans* contém 18.424 proteínas, podendo se agrupadas em 9.453 famílias. Comparando os dois organismos, existem aproximadamente 5.000 proteínas muito próximas, que podem ser consideradas como tendo uma função idêntica no processo metabólico das células. Também existem 3000 proteínas que são semelhantes a proteínas

do fungo *Saccharomyces cerevisiea* [9]. Todas estas particularidades fazem com que exista um ponto unificador na evolução das espécies. A Figura 7 evidencia três reinos, sendo eles:

- Eucariotas – este grupo inclui todos os organismos onde as células contêm o núcleo e as mitocôndrias isolados por membranas. O seu DNA está organizado em cromossomas dentro do núcleo. As plantas, animais e fungos são alguns dos organismos que fazem parte deste reino;
- Arqueobactérias – este grupo foi inicialmente criado para englobar os organismos que produziam gás metano ou que viviam em ambientes extremamente agressivos, como as fontes de água quente ou ambientes com elevados níveis de salinidade. No entanto, também se encontram organismos pertencentes a este grupo que vivem em ambientes não agressivos. Ao contrário dos eucariotas este grupo de organismos não tem membranas a isolar o núcleo ou mitocôndrias, embora o processo de replicação de DNA seja similar ao do reino das eucariotas;
- Eubactéria – contém os organismos unicelulares, reproduzindo-se por fissão binária, não tendo qualquer membrana para definir o núcleo à semelhança dos organismos que pertencem ao reino das arqueobactérias.

Os dois últimos reinos constituem um super reino com o nome de procariotas, referindo-se aos organismos sem núcleo.

O genoma dos seres vivos contém a sua história evolutiva escrita na sua parte codificante e não codificante. Por esta razão, O DNA é uma excelente material para identificar semelhanças e diferenças entre os seres vivos e entre os genes dos mesmos. Genes homólogos têm um passado comum na sua história evolutiva.

A análise da homologia genética passa por identificar genes semelhantes recorrendo a algoritmos matemáticos, actualmente disponíveis em aplicações comuns. A correcta identificação de homólogos faz-se através da análise filogenética das sequências dos genes. Homologia e semelhança são conceitos diferentes, genes homólogos podem ter uma semelhança alta ou baixa, e a presença de uma elevada semelhança não quer dizer que esses dois genes são homólogos, podem simplesmente ter convergido durante a evolução. Genes parálogos e genes ortólogos são subdivisões dos genes homólogos. Dois genes são ortólogos caso apresentem um antepassado comum. Os genes mitocondriais, por exemplo,

são considerados ortólogos, pois acredita-se que a origem da mitocôndria tenha ocorrido num estágio inicial da história da vida, antes de divergirem nos organismos eucariotas actuais.

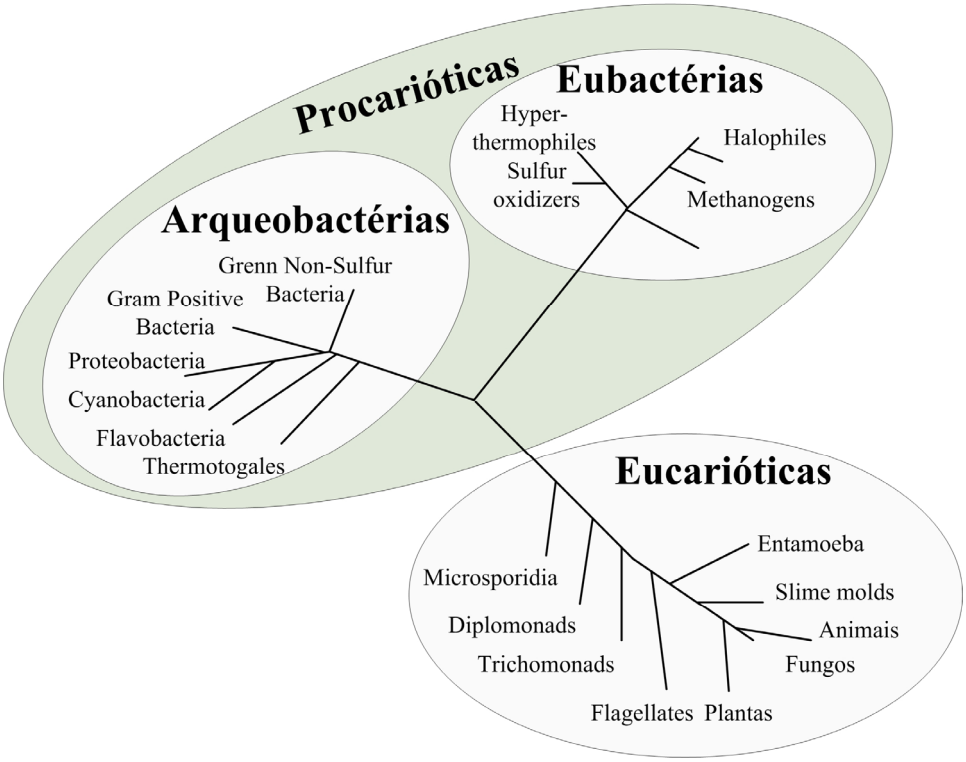

Figura 7 – Relação evolutiva entre vários grupos de espécies organizados em reinos.

Os genes parálogos surgem devido à duplicação de genes na mesma espécie que depois divergem. No entanto, nalguns casos, pequenas alterações a nível dos nucleótidos são suficientes para conferir ao gene uma função diferente.

Existe ainda outro caso de homologia, conhecida por genes xenólogos. Ocorre quando um gene de um organismo é introduzido noutro organismo, fenómeno conhecido como transferência horizontal. Isto pode ocorrer por acção de retrovirus.

A utilização de genes pertencentes às mesmas famílias, parálogos, para análise de sequências de uma mesma região do genoma, não é aconselhável porque as cópias dos genes podem possuir diferentes pressões selectivas, ter diferentes histórias evolutivas e até mesmo ocupar cromossomas diferentes. Por exemplo, em *Chaetognatha*, um grupo de invertebrados marinhos, há duas classes de genes ribossomais 28S. Ambas as cópias

parecem ser funcionais e possuem taxas evolutivas diferentes, o que causou grandes problemas de interpretação na história evolutiva deste grupo [40].

2.6 Conclusões

No presente capítulo apresentou-se uma pequena introdução sobre conceitos gerais de biologia molecular necessários ao estudo da caracterização das zonas codificantes dos organismos. A linguagem utilizada e a profundidade na abordagem dos conceitos estão necessariamente relacionadas com a natureza desta tese, não se pretendendo que atinja o nível requerido para uma tese em biologia. O objectivo desta introdução é o enquadramento das questões biológicas que suportaram o desenvolvimento das ferramentas informáticas descritas nos próximos capítulos.

Começou-se por apresentar a organização de uma célula, assim como os processos envolvidos no processamento da molécula de DNA. Os mecanismos envolvidos na tradução das zonas codificantes do genoma, que são responsáveis pela informação necessária à síntese das proteínas, foram também explicados, focando-se posteriormente nos erros da tradução, apresentando-se também as suas implicações. Foram ainda apresentados os vários índices que actualmente se utilizam para quantificar o *codon usage*. No entanto, esses índices não têm em conta a relação de dependência que existe entre codões, levando-nos a aprofundar o estudo nesse sentido.

No capítulo seguinte iremos apresentar as ferramentas computacionais disponíveis actualmente para efectuar estudos genómicos. Iremos também apresentar os formatos de dados mais comuns, assim como, os locais onde é possível obter informação necessária ao presente estudo.

Capítulo 3

---

3 Modelos de dados e suas ferramentas

3.1 Introdução

A sequenciação dos genomas veio trazer oportunidades únicas de descoberta e estudo dos processos que governam a vida. No entanto, a quantidade de informação gerada é imensa e a sua organização é complexa.

Uma simples bactéria pode conter aproximadamente três milhões de bases e conter 3.000 genes. Se passarmos para os organismos mais complexos, como os organismos eucariotas, o número facilmente duplica ou triplica, tendo de se definir formatos para a organização dos dados vão surgindo, assim como, locais públicos onde se pode guardar a informação para que possa estar facilmente acessível.

Ferramentas para tratar e explorar a informação são também importantes, surgindo inúmeras ferramentas para os mais variados estudos. No entanto, como em todas as áreas, muitas vão-se tornando populares enquanto outras não passarão de meros estudos académicos.

No presente capítulo vão ser apresentados os formatos de dados mais comuns assim como as bases de dados mais divulgadas, onde se poderá obter informação para efectuar estudos na área da genética, genómica e bioinformática. Várias ferramentas são também apresentadas, evidenciando-se aquelas que estejam directamente relacionadas com o estudo a desenvolver.

3.2 Base de Dados de biologia molecular

A informação sobre biologia molecular encontra-se dispersa por várias bases de dados e nos mais diversos formatos. Podemos dividir a informação em dois grandes grupos, os suportados por sistemas de ficheiros e os suportados pelas bases de dados relacionais.

As sequências genéticas que estão disponíveis através de ficheiros, normalmente disponíveis através de ligações *ftp*, tem a vantagem de se poder importar os ficheiros directamente e trabalhar com a informação em modo *off-line*. Quando se pretende trabalhar com a informação genética global de um determinado organismo é a melhor forma, pois essa informação normalmente ocupa vários Mbytes. Contudo, é sempre preciso estar atento à saída de novas versões, que substituem as anteriores, para evitar o risco de trabalhar com informação desactualizada.

As base de dados relacionais, disponíveis na *web* e acedidas directamente ou através de *web services*, são uma forma também vulgar de encontrar as sequências genéticas. Quando se quer trabalhar com as sequências globais de um determinado organismo o processo torna-se lento devido à importação de toda informação. Tem a grande vantagem de a informação estar sempre actualizada.

Desde 2000 que a revista *Nucleic Acids Research* (NAR) publica anualmente um artigo exclusivamente dedicado às bases de dados de biologia molecular disponíveis na *web* [41]. A última publicação de 2009 anuncia um total de 1230 bases de dados, algumas dedicadas somente a um organismo específico, outras mais generalistas, ainda outras dedicadas somente à informação de *microarrays* [42], outras sobre estruturas genéticas, mas todas com um objectivo comum, oferecer um acesso fácil e rápido à informação genética. No relatório do ano 2009 surgiram mais 58 bases de dados e alteração de conteúdos em 73 [41]. A lista completa encontra-se em (http://www3.oup.co.uk/nar/database/c).

Contudo, iremos centrar a nossa atenção somente em base de dados relativas às sequências genéticas, mais concretamente sequências de nucleótidos e aminoácidos. Este foco irá reduzir muito significativamente as bases de dados em análise, pois muitas das bases de dados apresentados na lista da NAR [43] contêm informação sobre doenças, *microarrays* [42], reacções enzimáticas, estruturas 3D de proteínas, informação médica, entre outros temas.

3.2.1 Base de dados de sequências

A três principais bases de dados de sequências de nucleótidos são a *European Nucleotide Sequence Database* (EMBL) pertencendo ao *European Bioinformatic Institute* (EBI), *DNA Data Bank of Japan* (DDBJ) pertencente ao *National Institute of Genetics* (NIG) no Japão, e o *Genetic Sequence Databank* (GenBank) que pertence ao *National Center for Biotechnology Information* (NCBI) nos Estados Unidos. As instituições que as regem mantêm um acordo de cooperação, mediado pelo *International Nucleotide Sequence Database Collaboration* (INSDC) [44], sendo composto por membros pertencentes aos consórcios proprietários das bases de dados.

A base de dados *Genetic Sequence Databank* (GenBank) contém sequências de nucleótidos de mais de 260.000 organismos obtidas através de submissões individuais, efectuadas por laboratórios espalhados por todo o mundo, ou submissões automáticas geradas por projectos de sequenciação em larga escala, como o caso do genoma humano [45]. As submissões são efectuadas através da aplicação *web* BankIt, ou através da aplicação Sequin, disponibilizada para trabalhar autonomamente mas muito mais poderosa do que a anterior. A atribuição de identificação única para as sequências recebidas, fica a cargo da equipa do GenBank.

Desde a sua criação, o GenBank duplica o seu tamanho a cada 18 meses, contendo, em 2008, aproximadamente 80 biliões de nucleótidos de 76 milhões de sequências, com 15 milhões de sequências adicionadas durante o ano de 2007 [45]. Os genomas de procariotas completos continuam também a crescer rapidamente, contando com mais 200 depositados durante o ano de 2007 somados aos 570 já existentes. Quanto aos organismos eucariotas, contém 190 incluindo o humano. A informação é acessível através da aplicação RefSeq, que permite o acesso não redundante às sequências de nucleótidos e suas anotações. No entanto, também é possível efectuar a procura através de similaridade entre sequências. É possível também obter a informação através de ligações *ftp*, podendo descarregar os ficheiros das sequências disponíveis no GenBank. A informação disponibilizada através do *ftp* é actualizada a cada dois meses.

O EMBL [46] é a base de dados europeia, sob alçada do EBI, que contém sequências de nucleótidos obtidas na literatura ou através de submissão individual. A submissão é efectuada por investigadores ou grupos de sequenciação através da aplicação WebIn. O

acesso à informação poderá ser feita através de um conjunto de ferramentas *web* disponibilizados para o efeito ou através do *ftp*.

A DDBJ [47] é a base de dados da responsabilidade do Japão, contento também sequências de nucleótidos, em linha com as anteriores apresentadas. A submissão é efectuada por investigadores ou grupos de sequenciação através da aplicação SAKURA ou através da aplicação Sequin. O acesso à informação pode ser através da *web*, *ftp* ou através de APIs disponibilizadas pelo instituto.

Diariamente são efectuadas trocas de informação entre as três bases de dados, ao abrigo do protocolo de colaboração INSDC, permitindo uma transversalidade das sequências genéticas existentes nas três bases de dados supra mencionadas.

Outra base de dados de referência é o *Kyoto Encyclopedia of Genes and Genomes* (KEGG) [48]. Embora o KEEG englobe um conjunto de serviços, como ontologias ou *pathways* [49], contém também uma parte dedicada somente a sequências de nucleótidos. As sequências são obtidas através de base de dados públicas, mas maioritariamente são obtidas através do NCBI-RefSeq. Os dados estão disponíveis através de *ftp*, ou através da aplicação *web* DBGET.

À parte das principais bases de dados sobre nucleótidos existem também algumas bases de dados direccionadas somente para um organismo, ou classes de organismos específicos, como o caso do Ensembl [50]. O Ensembl é um projecto de colaboração entre o EMBL-EBI e o *Wellcome Trust Sanger Institute*, que visa desenvolver um sistema que permita efectuar anotações automáticas em genomas eucariotas. Em 2008 continha trinta e sete genomas, disponibilizados através de uma aplicação *web*, o BioMart, ou através de uma API em *perl*. Através destas aplicações é possível obter variada informação sobre os genomas em causa, que vão deste as sequências completas, genes, intrões, exões, entre outras possibilidades.

Existem outras bases de dados dedicadas somente a um organismo como é o caso da EcoGene [51] do organismo *Escherichia coli K-12*, a *Candida Genome Database* CGD [52] ou a *Saccharomyces Genome Database* SGD [53]. Existe um conjunto vasto de bases de dados dedicadas a cada organismo, mas mencionamos três por pertencerem aos organismos mais estudados em laboratório.

As bases de dados específicas de organismos, não têm somente as sequências de nucleótidos, mas também as sequências das suas proteínas, bibliografia associada, assim como um conjunto vasto de anotações.

Relativamente às bases de dados de proteínas, existe um consórcio formado em 2002, com o nome de UniProt, que englobou as três bases de dados mais relevantes [54]: i) Swiss-Prot, contém todas as suas sequências manualmente anotadas; ii) TrEMBL, pertencente ao EBI, contém as sequências de proteínas automaticamente anotadas; iii) PIR-PSD, base de dados de proteínas pertencente à Georgetown University Medical Center.

A UniProt divide-se em três áreas, cada uma especializada em diferentes tipos de utilização. A *UniProt Knowledgebase* (UniProtKB) é o centro da informação, englobando a Swiss-Prot e a TrEMBL, disponibilizando uma vasta informação sobre proteínas, assim como a sua função, classificação e referências cruzadas para possíveis envolvimentos em doenças ou processos metabólicos. A *UniProt Reference Clusters* (UniRef) agrega as sequências próximas numa só, optimizando o processo de procura. O *UniProt Archive* (UniParc) contém somente as sequências não redundantes, sem qualquer notação. As proteínas existentes contêm todas as alterações que foram sendo efectuadas ao longo da construção da proteína, podendo assim obter um histórico da sua formação na base de dados.

A *Translated EMBL* (TrEMBL), é a maior base de dados de proteínas e foi obtida através da conversão automática da base de dados de nucleótidos da EMBL. A conversão automática não é totalmente fiável, podendo resultar em muitas proteínas hipotéticas, ou então com poucas anotações. No momento em que uma proteína é revista e anotada manualmente passa a fazer parte da Swiss-Prot.

A Swiss-Prot foi criada em 1986 por Amos Bairoch durante o seu doutoramento, e tem sido mantida pelo *Swiss Institute of Bioinformatics* e pelo EBI [55]. A Swiss-Prot só contém proteínas anotadas manualmente, com elevada qualidade de informação e com baixa redundância, possibilitando também uma fácil integração com outras bases de dados.

A *Protein Information Resource* (PIR) [56] foi a primeira base de dados contendo proteínas anotadas e funcionalmente classificadas. A informação da PIR pode ser actualmente encontrada na UniProt, podendo também ser acedido directamente através da *web* ou obtendo a informação através de *ftp*.

Por último, referimos uma base de dados muito importante para o estudo da estrutura primária dos genes. A base de dados *Highly Expressed Genes Database* (HEG-DB) [57] contém os genes com elevados níveis de expressão para aproximadamente duzentos organismos disponibilizando também as tabelas de *codon usage* e os respectivos CAI. No entanto, só contém organismos procariotas. O acesso é realizado através da sua página *web*, obtendo os valores em ficheiros texto. Estes genes têm especial interesse, porque, como genes muito expressos, deverão ter um baixo nível de erro na sua tradução.

3.3 Formato de dados

3.3.1 Sequências de nucleótidos ou aminoácidos

As bases de dados e aplicações de bioinformática são desenvolvidas para trabalhar com sequências, necessitando de uma normalização para troca de informação sem necessidade de grandes adaptações. Contudo, existe uma ferramenta READSEQ [58], disponibilizada pelos serviços do EBI, que permite a conversão entre os vários formatos de sequências.

Existem diferentes tipos de normalizações mediante o tipo de informação que se pretende expor, ou mediante a fonte de dados. Se a informação for referente a sequências de nucleótidos ou aminoácidos os formatos mais comuns são FASTA, GenBank, Embl ou EMBOSS.

O formato FASTA é formado por simples sequências contendo aminoácidos ou nucleótidos. A primeira linha, que começa com o símbolo “>”, refere-se à identificação da sequência, contendo o nome da sequência e a sua descrição, normalmente separados com o símbolo “|” (Figura 8). As linhas seguintes contêm a sequência. Normalmente, as sequências têm 60 caracteres por linha, terminando quando aparece novamente um símbolo “>”, que indica o início da identificação de uma nova sequência. Cada ficheiro pode conter uma ou mais sequências sendo o formato bastante intuitivo e permitindo alterações com um simples editor de texto.

```
>AB000263 |acc=AB000263|descr=Homo sapiens mRNA complete cds.|len=368
ACAAGATGCCATTGTCCCCCGGCCTCCTGCTGCTGCTGCTCTCCGGGGCCACGGCCACCGCTGCCCTGCC
CCTGGAGGGTGGCCCCACCGGCCGAGACAGCGAGCATATGCAGGAAGCGGCAGGAATAAGGAAAAGCAGC
CTCCTGACTTTCCTCGCTTGGTGGTTTGAGTGGACCTCCCAGGCCAGTGCCGGGCCCCTCATAGGAGAGG
AAGCTCGGGAGGTGGCCAGGCGGCAGGAAGGCGCACCCCCCCAGCAATCCGCGCGCCGGGACAGAATGCC
CTGCAGGAACTTCTTCTGGAAGACCTTCTCCTCCTGCAAATAAAACCTCACCCATGAATGCTCACGCAAG
TTTAATTACAGACCTGAA
```

Figura 8 – Sequência no formato FASTA.

O formato EMBL é usado para sequências de aminoácidos e nucleótidos sendo estruturado por linhas (Figura 9). Cada linha tem um código inicial composto por dois caracteres indicando o tipo de informação que se segue. Cada sequência começa incondicionalmente com o identificador "ID" e termina com a linha que contém "//". A sequência está identificada pelo marcador "SQ", assim como o seu comprimento.

```
ID   AB000263 standard; RNA; PRI; 368 BP.
XX
AC   AB000263;
XX
DE   Homo sapiens mRNA for prepro cortistatin like peptide, complete cds.
XX
SQ   Sequence 368 BP;
     acaagatgcc attgtccccc ggcctcctgc tgctgctgct ctccggggcc acggccaccg        60
     ctgccctgcc cctggagggt ggccccaccg gccgagacag cgagcatatg caggaagcgg       120
     caggaataag gaaaagcagc ctcctgactt tcctcgcttg gtggtttgag tggacctccc       180
     aggccagtgc cgggcccctc ataggagagg aagctcggga ggtggccagg cggcaggaag       240
     gcgcaccccc ccagcaatcc gcgcgccggg acagaatgcc ctgcaggaac ttcttctgga       300
     agaccttctc ctcctgcaaa taaaacctca cccatgaatg ctcacgcaag tttaattaca       360
     gacctgaa                                                               368
//
```

Figura 9 – Sequência no formato EMBL.

GenBank é uma base de dados da *National Institutes of Health* (NIH), como já foi referido anteriormente. No entanto, o nome também é atribuído à estrutura dos seus ficheiros (Figura 10). A estrutura do GenBank permite uma descrição muito completa das sequências que agrega, que vai desde o nome do organismo, à taxonomia, mutações e repetições, identificação das zonas de codificação entre outras áreas de interesse existentes nas sequências. Inclui ainda referências bibliográficas sobre quem identificou a sequência. O ficheiro é composto por um conjunto de marcadores que identifica o seu conteúdo. Existem vários tipos de marcadores específicos, sendo os mais comuns: i) LOCUS, contém o nome da sequência; ii) DEFINITION, contém o nome do organismo a que pertence a sequência; iii) FEATURES, contém as posições dos genes na sequência assim como a sua tradução para aminoácidos, entre outras características; iv) ORIGIN, contém a sequência

em nucleótidos. Existem outros marcadores para descrever outro tipo de dados, no entanto, não existe obrigatoriedade de estarem todos presentes nos ficheiros de formato GenBank.

```
LOCUS       AB000263                368 bp    mRNA    linear   PRI 05-FEB-1999
DEFINITION  Homo sapiens mRNA for prepro cortistatin like peptide, complete
            cds.
ACCESSION   AB000263
FEATURES             Location/Qualifiers
     source          1..368
                     /organism="Homo sapiens"
                     /mol_type="genomic DNA"
                     /db_xref="taxon:115711"
     gene            10..320
                     /locus_tag="CP0001"
                     /db_xref="GeneID:963624"
     CDS             10..320
                     /locus_tag="CP0001"
                     /note="similar to GB:L24386 SP:P45622 PID:416155;
                     identified by sequence similarity; putative"
                     /codon_start=1
                     /transl_table=11
                     /product="delta-aminolevulinic acid dehydratase"
                     /protein_id="NP_444554.1"
                     /db_xref="GI:16752297"
                     /db_xref="GeneID:963624"
                     /translation="MSSLTLSRRPRRNRKTAAIRDLLAETHLSPKDLIAPFFVKYGNN
                      VSGEYAMILSAFQQGWLDKETLFHESLIAIKRAGADMIISYSAPFILELLHQGFEF"
ORIGIN
        1 acaagatgcc attgtccccc ggcctcctgc tgctgctgct ctccggggcc acggccaccg
       61 ctgccctgcc cctggagggt ggccccaccg gccgagacag cgagcatatg caggaagcgg
      121 caggaataag gaaaagcagc ctcctgactt tcctcgcttg gtggtttgag tggacctccc
      181 aggccagtgc cgggcccctc ataggagagg aagctcggga ggtggccagg cggcaggaag
      241 gcgcaccccc ccagcaatcc gcgcgccggg acagaatgcc ctgcaggaac ttcttctgga
      301 agaccttctc ctcctgcaaa taaaacctca cccatgaatg ctcacgcaag tttaattaca
      361 gacctgaa
//
```

Figura 10 – Sequência genética no formato GenBank

O formato PIR, também conhecido por NBRF, é um formato muito parecido ao FASTA, iniciado com o símbolo '>' seguido de um código composto por dois caracteres. O código descreve qual o tipo de sequência que contém o ficheiro, variando entre sequências de aminoácidos e nucleótidos. As sequências terminam sempre com símbolo '*' (Figura 11).

```
>P1;CRAB_ANAPL
ALPHA CRYSTALLIN B CHAIN (ALPHA(B)-CRYSTALLIN).
  MDITIHNPLI RRPLFSWLAP SRIFDQIFGE HLQESELLPA SPSLSPFLMR
  SPIFRMPSWL ETGLSEMRLE KDKFSVNLDV KHFSPEELKV KVLGDMVEIH
  GKHEERQDEH GFIAREFNRK YRIPADVDPL TITSSLSLDG VLTVSAPRKQ
  SDVPERSIPI TREEKPAIAG AQRK*
```

Figura 11 – Sequência no formato PIR.

O formato PDB é amplamente utilizado para descrever a estrutura terciária de proteínas (Figura 12). Basicamente este ficheiro contém as coordenadas atómicas da proteína obtidas através da cristalografia em Raio-X. Posteriormente, estes ficheiros são lidos por aplicações capazes de apresentar proteínas em formato 3D.

```
COMPND   MOL_ID: 1;
COMPND   2 MOLECULE: GLUTATHIONE SYNTHETASE;
COMPND   3 CHAIN: NULL;
COMPND   4 SYNONYM: GAMMA-L-GLUTAMYL-L-CYSTEINE\:GLYCINE LIGASE
COMPND   5 (ADP-FORMING);
COMPND   6 EC: 6.3.2.3;
COMPND   7 ENGINEERED: YES

COMPND    MOL_ID: 1;
COMPND   2 MOLECULE: S-ADENOSYLMETHIONINE SYNTHETASE;
COMPND   3 CHAIN: A, B;
COMPND   4 SYNONYM: MAT, ATP\:L-METHIONINE S-ADENOSYLTRANSFERASE;
COMPND   5 EC: 2.5.1.6;
COMPND   6 ENGINEERED: YES;
COMPND   7 BIOLOGICAL_UNIT: TETRAMER;
COMPND   8 OTHER_DETAILS: TETRAGONAL MODIFICATIONs
```

Figura 12 – Formato PDB que descreve a estrutura terciária de proteínas.

O formato UniProtKB/Swiss-Prot é específico para proteínas, tentando seguir a estrutura do EMBL o mais próximo possível (Figura 13). A estrutura é orientada à linha, contendo um código composto com dois caracteres que corresponde a um significado, dando a possibilidade de adicionar várias anotações à proteína. No ficheiro é possível descrever variada informação associada à sequência, que vai desde o nome do organismo, referências, posição da sequência em relação ao genoma, entre outros. O código 'SQ' indica o começo da proteína agrupada em blocos de dez aminoácidos (Figura 13). A primeira linha do ficheiro tem sempre o código 'ID', contendo o nome da sequência assim como o seu estado actual de revisão e o seu comprimento em aminoácidos. O marcador *Reviewed* indica se a proteína já foi anotada manualmente por um curador. Como foi referido anteriormente na secção 3.2.1 deste capítulo, a base de dados UniProtKB\TrEMBL contém proteínas que ainda não foram revistas manualmente pelos

curadores, passando para a base de dados UniProtKB\Swiss-Prot no momento da sua revisão manual. No caso da proteína ainda não ter sido revista manualmente o seu estado é *Unreviewed*.

```
ID   FOSB_MOUSE              Reviewed;         338 AA.
AC   P13346;
DT   20-FEB-2007, entry version 54.
DE   Protein fosB.
GN   Name=Fosb;
OS   Mus musculus (Mouse).
OX   NCBI_TaxID=10090;
RN   [1]
RP   NUCLEOTIDE SEQUENCE [MRNA].
RX   MEDLINE=89251612; PubMed=2498083;
RA   Zerial M., Toschi L., Ryseck R.-P., Schuermann M., Mueller R.,
RA   Bravo R.;
RT   "The product of a novel growth factor activated gene, fos B, interacts
RT   with JUN proteins enhancing their DNA binding activity.";
RL   EMBO J. 8:805-813(1989).
RP   NUCLEOTIDE SEQUENCE [GENOMIC DNA].
RX   MEDLINE=92158623; PubMed=1741260; DOI=10.1093/nar/20.2.343;
RA   Lazo P.S., Dorfman K., Noguchi T., Mattei M.-G., Bravo R.;
DR   EMBL; X14897; CAA33026.1; -; mRNA.
KW   DNA-binding; Nuclear protein.
FT   CHAIN         1    338       Protein fosB.
FT                                /FTId=PRO_0000076477.
SQ   SEQUENCE   338 AA;  35977 MW;  E9D031A4BEAE48EC CRC64;
     MFQAFPGDYD SGSRCSSSPS AESQYLSSVD SFGSPPTAAA SQECAGLGEM PGSFVPTVTA
     ITTSQDLQWL VQPTLISSMA QSQGQPLASQ PPAVDPYDMP GTSYSTPGLS AYSTGGASGS
     GGPSTSTTTS GPVSARPARA RPRRPREETL TPEEEEKRRV RRERNKLAAA KCRNRRRELT
     DRLQAETDQL EEEKAELESE IAELQKEKER LEFVLVAHKP GCKIPYEEGP GPGPLAEVRD
     LPGSTSAKED GFGWLLPPPP PPPLPFQSSR DAPPNLTASL FTHSEVQVLG DPFPVVSPSY
     TSSFVLTCPE VSAFAGAQRT SGSEQPSDPL NSPSLLAL
//
```

Figura 13 – Formato UniProtKB/Swiss-Prot

3.3.2 Alinhamentos

Genetics Computer Group/Multiple Sequence Files (GCG/MSF) é um formato que representa sequências alinhadas produzidas por aplicações pertencentes à GCG *Wisconsin Package* [59]. O formato tem de obedecer a determinadas especificações. O início do ficheiro pode conter vários comentários iniciados com o marcador "\\". A primeira linha contém o marcador "MSF:" com várias descrições das sequências que se seguem, seguido obrigatoriamente de uma linha em branco. Seguidamente contém os nomes das sequências alinhadas assim como o seu comprimento e o peso obtido no alinhamento, também

conhecido como coeficiente de fusão. Por fim estão representadas a sequências alinhadas agrupadas em blocos de dez aminoácidos (Figura 14).

```
MSF:  510  Type: P    Check:  7736   ..

 Name: ACHE_BOVIN oo  Len:  510  Check:  7842  Weight:  16.0
 Name: ACHE_HUMAN oo  Len:  510  Check:  8553  Weight:  17.8
 Name: ACHE_MOUSE oo  Len:  510  Check:   229  Weight:  12.5
 Name: ACHE_RAT oo  Len:  510  Check:  8410  Weight:  14.2
 Name: ACHE_XENLA oo  Len:  510  Check:  2702  Weight:  39.2

//

ACHE_BOVIN      MAGALLCALL LLQLLGRGEG KNEELRLYHY LFDTYDPGRR PVQEPEDTVT
ACHE_HUMAN      MARAPLGVLL LLGLLGRGVG KNEELRLYHH LFNNYDPGSR PVREPEDTVT
ACHE_MOUSE      MAGALLGALL LLTLFGRSQG KNEELSLYHH LFDNYDPECR PVRRPEDTVT
ACHE_RAT        MTMALLGTLL LLALFGRSQG KNEELSLYHH LFDNYDPECR PVRRPEDTVT
ACHE_XENLA      MESGVRILSL LILLHNSLAS ESEESRLIKH LFTSYDQKAR PSKGLDDVVP

ACHE_BOVIN      ISLKVTLTNL ISLNEKEETL TTSVWIGIDW QDYRLNYSKG DFGGVETLRV
ACHE_HUMAN      ISLKVTLTNL ISLNEKEETL TTSVWIGIDW QDYRLNYSKD DFGGIETLRV
ACHE_MOUSE      ITLKVTLTNL ISLNEKEETL TTSVWIGIDW HDYRLNYSKD DFAGVGILRV
ACHE_RAT        ITLKVTLTNL ISLNEKEETL TTSVWIGIEW QDYRLNFSKD DFAGVEILRV
ACHE_XENLA      VTLKLTLTNL IDLNEKEETL TTNVWVQIAW NDDRLVWNVT DYGGIGFVPV
```

Figura 14 – Alinhamentos no formato GCG/MSF

A aplicação “CLUSTAL” grava as sequências alinhadas no formato presente na Figura 15. A primeira linha começa sempre com a palavra "CLUSTAL", que é a aplicação que produziu este tipo de ficheiros. Cada bloco começa com o nome das sequências seguido das sequências alinhadas por linhas. A última linha indica qual a qualidade do alinhamento por coluna. O asterisco apresenta o alinhamento máximo, indicando que contém somente um aminoácido nessa coluna. Os dois pontos ou ponto indica que determinada coluna contém aminoácidos pertencentes a grupos considerados conservados ou semi-conservados, respectivamente. Os aminoácidos são considerados conservados ou semi-conservados se partilharem determinadas propriedades físicas entre eles. No caso de não existir alinhamento nessa coluna a linha da qualidade contém um espaço em branco.

```
CLUSTAL W 2.1 multiple sequence alignment

FOSB_MOUSE      ITTSQDLQWLVQPTLISSMAQSQGQPLASQPPAVDPYDMPGTSYSTPGLSAYSTGGASGS 60
FOSB_HUMAN      ITTSQDLQWLVQPTLISSMAQSQGQPLASQPPVVDPYDMPGTSYSTPGMSGYSSGGASGS 60
                ********************************.***************:*.**:******

FOSB_MOUSE      TLISSMAQSQGQPLASQPPAVDPYDMPGTSYS 94
FOSB_HUMAN      TLISSMAQSQGQPLASQPPVVDPYDMPGTSYS 94
                ******************.************
```

Figura 15 – Alinhamento produzido pelo Clustal

O formato PHYLIP, sendo também um formato que representa sequências alinhadas, é muito próximo do formato do formato do Clustal. No entanto, não tem a última linha que indica a qualidade do alinhamento. A informação também é apresentada por blocos, mas só o primeiro bloco é que contém o nome das sequências (Figura 16). O formato PHYLIP pertence a um pacote de trinta e cinco aplicações especializadas na análise e construção de árvores filogenéticas com o nome de *PHYLogeny Inference Package* (PHYLIP) [60, 61].

```
     7    96
SalmoSalar ATGGCTGAGG CATTCGCAGG CACATGGAAC CTGAAGGACA GCAAGAACTT
TetraodonN ATGGCCGAAG CTTTTGCGGG AACGTGGAAC CTCGTCAAAA GCGAAAAATT
FundulusHe ATGGTCGAGG CTTTCGTGGG CACGTGGAAC CTGAAGGAGA GCGAGAATTT
TaeniaSoli ---ATGGAGC CATTCATCGG TACCTGGAGG ATGGAGAAGA GTGAGGGTTT
RattusNorv ATGTGCGACG CCTTTGTGGG GACCTGGAAA CTCGTCTCCA GTGAGAACTT
DanioRerio ATGGTTGACA AATTCGTAGG AACGTGGAAA ATGACCACCA GCGACAACTT
XenopusLae ATGGTGGATC AATTCGTGGG CTCCTGGAAG CTGACTGACA GCCAGGGATT

           AGTGA----- ---------- ---------- ---------- ------
           ---------- ---------- ---------- ---------- ------
           AG-------- ---------- ---------- ---------- ------
           CGTAA----- ---------- ---------- ---------- ------
           NAGCCAAGGG ACGAGGCCCT GGACTGAAAT TTGCATCAAA ACTTGA
           AGGCATGA-- ---------- ---------- ---------- ------
           AGGCATAA-- ---------- ---------- ---------- ------
```

Figura 16 – Formato PHYLIP

A lista que contempla os formatos mais comuns assim como a sua extensão encontra-se no anexo 8.3.

3.4 Ferramentas para estudo da estrutura primária

Actualmente existem várias soluções disponíveis para o estudo da estrutura primária das sequências genómicas. As soluções podem-se dividir em duas categorias: bibliotecas para

construir soluções para um problema específico ou aplicações executáveis direccionadas ao estudo de um problema em particular.

A grande vantagem de se utilizar as bibliotecas é poder ter liberdade de construir ou realçar determinado aspecto em que o utilizador se pretende focar. Relativamente às aplicações fechadas o utilizador terá de se cingir às opções fornecidas. Caso a aplicação tenha o código aberto, poderá modificar o código para construir as opções que necessita. Isto pressupõe o conhecimento da linguagem, na qual a aplicação foi construída, e supostamente o consumo de largas horas para compreender o que já está realizado.

Existem algumas aplicações disponíveis no meio académico que calculam vários índices imprescindíveis para caracterizar os genes. Iremos de seguida apresentar as aplicações consideradas mais relevantes para o estudo da estrutura primária de genes.

CodonW [62] é a aplicação mais completa para caracterizar a estrutura primária de um determinado gene ou genoma. Esta aplicação foi desenvolvida no âmbito de uma tese de doutoramento, estando a aplicação disponível no *sourceforge* para *download* e posterior desenvolvimento (http://codonw.sourceforge.net/). A aplicação consegue obter variados índices, como o CAI, *codon usage*, %GC entre outros [62]. A informação obtida é guardada em ficheiros podendo o utilizador fazer outras análises sobre a informação obtida. Não existe qualquer tipo de apresentação gráfica para visualizar os resultados obtidos. A aplicação é acedida através de linha de comandos, o que dificulta a sua interacção para utilizadores menos familiarizados com a linha de comandos. O CodonW é sem dúvida a aplicação mais comum na obtenção de vários índices para o estudo de sequências genéticas.

O *INteractive Codon usage Analysis* (INCA) [63] é uma aplicação que de uma forma intuitiva permite obter os índices das sequências submetidas. Calcula vários índices como as frequências de codões, *codon bias*, numero efectivo de codões (ENc), hidrofobicidade e o *codon adaptation index* (CAI). A aplicação conta também com uma rede neuronal não supervisionada para fazer *clustering* entre genes com base nos codões mais comuns. As sequências para análise são lidas através de ficheiros em formato FASTA. A aplicação tem uma ambiente gráfico rico e de fácil interacção. No global é uma aplicação muito intuitiva e útil para quem pretende analisar sequências com os índices que ela oferece.

O *Graphical Codon Usage Analyser* [27] é uma ferramenta *web* que compara as sequências submetidas com uma tabela *codon usage* definida pelo utilizador. A tabela *codon usage* é definida através de um URL, definido pelo utilizador, que terá de apontar para as tabelas de *codon usage* publicadas por Nakamura [30]. Existem três tipos de comparação: i) contabiliza todos os codões existentes na sequência, comparando os valores cumulativos para cada codão disponível com os níveis existentes na matriz, possibilitando a identificação de zonas com baixo *codon usage*; ii) compara as contagens totais dos codões presentes na sequência, dos 64 codões disponíveis no código genético, com os valores da tabela do *codon usage*; iii) compara duas tabelas de *codon usage*. Esta ferramenta é muito fraca quanto à análise de sequências mas de fácil utilização.

Para além destas aplicações existe também uma quantidade considerável de bibliotecas desenvolvidas em diferentes linguagens de programação. Para *perl* existe o BioPerl [64]; para *python* existe o BioPython [65]; para *ruby* o BioRuby [66]; para *java* o BioJava [67]; para C++ existe NCBI C++ *Toolkits* [68], e para C# não existe nenhuma biblioteca utilizada pela comunidade, embora estejam a surgir as primeiras tentativas.

Não se pode finalizar esta secção sem menciona a *toolbox* que o Matlab disponibiliza para a bioinformática, embora seja uma solução comercial. A *toolbox* oferece um conjunto de ferramentas para apoiar os investigadores nas mais diversas áreas, como seja na pesquisa de novos medicamentos, engenharia genética ou mesmo desenvolver novos algoritmos nas áreas de genómica e proteómica. A *toolbox* tem a capacidade de adquirir os dados nas mais diversas fontes e formatos tendo mesmo um módulo específico para visualizar sequências genómicas e análise de *microarrays*. Estas opções estão implementadas na linguagem do Matlab, com o código fonte disponível, podendo ajustar as funções ao problema a investigar. Inclui também a capacidade de organizar a informação consoante a ontologia e a possibilidade de construir árvores filogenéticas. Além destas funções específicas para a bioinformática o utilizador tem ao seu dispor um vasto conjunto de funções estatísticas básicas disponibilizadas pelo Matlab, ou podendo ir obter módulos mais específicos na *toolbox* de estatística do Matlab.

No conjunto das soluções apresentadas, não existe nenhuma aplicação que efectue estudos sistemáticos a nível de contexto de codões, tornando-se por isso uma questão premente a

construção de uma aplicação deste domínio. Contudo, essa aplicação não podia deixar de incluir algum do trabalho realizado anteriormente por outros grupos de investigação.

3.5 Ferramentas de alinhamento de sequências

O alinhamento de sequências será certamente um dos temas mais importante para qualquer biólogo que pretenda estudar sequências genéticas. É através destas ferramentas que se pode estudar a homologia de determinadas sequências, perceber a evolução filogenética, anotar genes, detectar as funções regulatórias ou simplesmente estudar a evolução das espécies.

Podemos dividir as ferramentas de alinhamento em duas categorias: i) procura de sequências similares numa base de dados; ii) alinhar duas ou mais sequências entre si.

A procura de sequências similares pressupõe a existência de uma sequência que pretendemos alinhar contra uma base de dados de sequências. A base de dados é previamente construída para o efeito contendo as sequências genéticas de um ou mais organismos. Estes repositórios podem conter somente um organismo ou a quase totalidade dos organismos sequenciados até ao momento. Podemos efectuar uma procura de similaridade contra a base de dados da SwissProt ou contra a base de dados do NCBI, podendo também criar a nossa própria base de dados. Por exemplo, o primeiro passo a seguir à sequenciação de um organismo é a procura de sequências similares nas bases dados existentes, para testar se existem partes comuns às sequências já conhecidas.

No caso de alinhar duas ou mais sequências pretende-se obter qual o melhor alinhamento, procurando assim, zonas conservadas presentes entre as sequências.

3.5.1 Procura de sequências similares numa base de dados

Existem várias aplicações disponíveis para procurar sequências similares em bases de dados previamente criadas, no entanto não é objectivo desta secção descrever todas em pormenor. Vamos apresentar as mais utilizadas pela comunidade de bioinformática.

O *Basic Local Alignment Search Tool* 2.0 (BLAST) do NCBI, também conhecido como *Gapped* BLAST [69] é certamente o método mais utilizado pela comunidade de bioinformática para identificar sequências similares em bases de dados de nucleótidos e

proteínas. O BLAST utiliza um algoritmo heurístico e aplica a técnica de alinhamento local. A maior parte das proteínas são de natureza modular com domínios funcionais repetidos dentro da mesma proteína, bem como repetindo esses domínios entre proteínas de espécies diferentes. O algoritmo BLAST é optimizado para encontrar esses domínios ou pequenas sequências semelhantes. Se o BLAST tentasse alinhar duas sequências em todo o seu comprimento, conhecido como alinhamento global, poucas semelhanças seriam detectadas.

O NCBI-BLAST 2.0 é um conjunto de aplicações que utilizam o algoritmo BLAST de diversas formas sobre sequências de nucleótidos e de proteínas (Tabela 2).

Tabela 2 - Várias variantes existentes do BLAST do NCBI

Aplicação	Sequência de entrada	Base de dados	Objectivo
BLASTn	Nucleótidos	Nucleótidos	Este método é usado para encontrar sequências homólogas na base de dados de nucleótidos.
BLASTx	Nucleótidos	Proteínas	Converte automaticamente a sequência de nucleótidos para aminoácidos e procura numa base de dados de proteínas.
tBLASTx	Nucleótidos	Nucleótidos	Converte a sequência para aminoácidos e procura numa base de dados que foi construída a partir de sequências de nucleótidos convertidos posteriormente para aminoácidos.
BLASTp	Proteínas	Proteínas	Usado para procurar sequências homologas em que a sequência de entrada e a base de dados são proteínas.
tBLASTn	Proteínas	Nucleótidos	Procura sequências de proteínas em bases de dados construídas a partir de sequências de nucleótidos convertidos automaticamente para aminoácidos.

FASTA [70] é um conjunto de aplicações, como o BLAST do NCBI, que permitem efectuar o teste de similaridade em bases de dados de nucleótidos e proteínas. Foi o primeiro algoritmo amplamente utilizado para a procura de similaridades. O programa efectua a procura dividindo a sequência em pequenas palavras, como o BLAST. Este programa tem melhor sensibilidade em pesquisas realizadas nas base de dados de nucleótidos porque permite diminuir a palavra de procura abaixo de sete caracteres ao contrário do BLAST [71], no caso de procura de sequências de nucleótidos. A sensibilidade e a velocidade da pesquisa são inversamente relacionados e controlados pela

variável que especifica o tamanho de uma palavra. Este programa também apresenta bons resultados quando o objectivo é identificar longas sequências com baixa similaridade, especialmente para sequências muito divergentes [72]. Esta aplicação é oferecida pelo *European Bioinformatics Institute* (EBI), dispondo assim de acesso privilegiado às bases de dados do EMBL.

WU-BLAST [73] é uma aplicação de similaridade desenvolvida na Universidade de Washington. É muito similar ao BLAST mas tem algo que o torna particular, o parâmetro da sensibilidade que permite ter um controlo sob o desempenho. O aumento da sensibilidade faz com que o algoritmo demore mais tempo a produzir os resultados. As bases de dados escolhidas variam entre as de nucleótidos e de proteínas, podendo também ser escolhida uma especificamente contendo as sequências dos organismos parasitas [72].

MPsrch [74] é uma ferramenta especificamente direccionada para proteínas implementando o algoritmo Smith-Waterman. Esta aplicação é uma das mais sensíveis na procura de proteínas, identificando algumas similaridades onde o BLAST e o FASTA não encontram, embora reporte alguns falsos positivos. Esta aplicação é disponibilizada pelo EBI e só podem ser escolhidas bases de dados de proteínas [72].

Actualmente, muitos investigadores privilegiam a rapidez na procura de sequências idênticas. Por exemplo, a procura de polimorfismos ou mutações no genoma humano em relação a uma sequência. O *Sequence Search and Alignment by Hashing* (SSAHA) foi desenvolvido para procurar sequências de forma muito rápida e próximas da sequência de pesquisa [75]. O algoritmo cria uma *hash table* com as sequências alvo, tornando a procura mais rápida e concluindo o processo com a junção dos resultados. Outra aplicação que funciona de forma idêntica é o BLAT (BLAST-*like alignment tool*) [76].

Muitas das funções das proteínas e sua evolução são encontradas através da comparação das estruturas 3D [77]. Quando não existem essas estruturas 3D poderá optar-se por estabelecer relações entre regiões conservadas. O *Position-Specific Iterated* BLAST (PSI-BLAST) [69] é uma aplicação que recorre à recursividade através do BLAST para identificar as funções de proteínas. A primeira interacção com o BLAST serve para criar um perfil que será usado nas interacções seguintes. Um novo BLAST é repetido, mas desta vez com o perfil criado anteriormente, até o algoritmo convergir, parando quando o resultado da interacção actual for inferior à interacção anterior [78].

O *Scan Protein Sequence* (SCANPS) [79] é uma aplicação para procurar similaridades em proteínas. Implementa o algoritmo Smith-Waterman e é capaz de identificar múltiplos domínios, muito similar ao PSI-BLAST [72].

Todas as aplicações anteriormente apresentadas têm a versão que se poderá utilizar na *web* e uma outra versão para correr localmente. A versão utilizada através de páginas *web* tem a vantagem dos algoritmos correrem em *clusters* de computadores, tornando mais rápida a sua execução. Regra geral, efectua-se a procura num vasto conjunto de organismos, recorrendo-se assim à versão da *web* e tirando vantagem dos servidores disponibilizados. Por vezes, em aplicações específicas, o investigador pretende procurar a homologia somente em partes de um determinado organismo, ou em organismos específicos, sendo mais complicado efectuar essa procura via *web*. Os institutos que disponibilizam estas aplicações via *web*, têm sempre versões que correm a nível local, permitindo assim um maior controlo de entrada de dados e tratamento dos resultados. Por exemplo, no NCBI, quando se submetem várias sequências ao BLAST o sistema vai aumentando progressivamente o tempo de aceitação do trabalho submetido aos seus servidores. Assim sendo, quando temos de submeter várias sequências a um conjunto de organismo é preferível correr o BLAST local.

O NCBI disponibiliza o BLAST local através a sua *toolboox* em C++ [80]. Esta ferramenta pode ser integrada e compilada em outra aplicação ou pode servir como aplicação única que corre através da linha de comando [69]. Contudo, é mais complicado para a maioria dos biólogos correr aplicações através da linha de comandos, sendo por isso usual encontrarmos algumas aplicações que disponibilizam o BLAST com interface gráfica usando como motor o pacote do NCBI.

O WinBlast, por exemplo, é uma aplicação visual que permite executar o BLAST local. No entanto, é necessário criar a base de dados através da linha de comandos e não permite submeter mais de uma sequência em simultâneo.

O bioEdit [81] é outra ferramenta disponível para a comunidade da bioinformática, mais orientada para a edição de sequências de nucleótidos e aminoácidos, alinhamentos, manipulação de sequências e sua análise. Também disponibiliza o BLAST local, mas não faz qualquer tratamento dos resultados, apresentando-os da mesma forma como são apresentados pelo BLAST.

CLC RNA *workbench* [82] é um produto comercial que oferece um vasto conjunto de ferramentas para análise sequências, disponibilizando o acesso ao BLAST local através do modo gráfico. Poderá também aceder ao BLAST através da *web* do NCBI, efectuando pesquisas de semelhança nas bases de dados públicas. Contudo, não deixa de ser uma solução comercial, e executar o BLAST para muitas sequências em simultâneo não é uma tarefa simples.

Existem também formas de executar o BLAST local em *clusters* de computadores, tornando mais rápida a procura de semelhanças nas bases de dados disponíveis. O mpiBLAST é uma das aplicações *open source* que torna possível correr o NCBI-BLAST em *clusters* de computação [83]. Embora esta aplicação não tenha sido desenvolvida pelo NCBI, utiliza o BLAST como motor, sendo o mpiBLAST responsável pela distribuição de tarefas nos diversos BLASTs existentes nos nós do *cluster*. Devido a esta arquitectura, todos os parâmetros de entrada apresentam a mesma forma do NCBI-BLAST.

O NCBI, além de disponibilizar a aplicação BLAST para utilizar localmente, também disponibiliza para *download* as suas bases de dados, tornando possível replicar localmente as experiências efectuadas através da *web*. Combinando as bases dados disponibilizadas com o mpiBLAST, pode-se diminuir em muito o tempo necessário para efectuar a procura de semelhanças efectuadas nas bases dados. Como já referido anteriormente, o NCBI-BLAST disponibilizado na *web*, vai aumentando progressivamente o tempo de aceitação dos pedidos aos seus servidores quando se submetem várias sequências em simultâneo. Embora esta técnica seja uma defesa contra ao *denial-of-service* dos servidores, cria, por vezes, muito tempo de espera para o investigador que pretende efectuar as suas pesquisas. Para laboratórios com disponibilidade para criar um *cluster* de computadores e que recorram sistematicamente aos serviços BLAST, a melhor solução passa por instalar o mpiBLAST, diminuindo assim o tempo de espera para obter os resultados na procura de semelhanças.

3.5.2 Alinhamento múltiplo

Como diferentes organismos têm ancestrais comuns é natural que diferentes espécies contenham sequências muito próximas. Devido a esta característica é possível estudar a sua evolução através do alinhamento múltiplo. As regiões conservadas são as zonas chave

para procurar semelhanças, obtendo quais as suas funções biológicas ou dando pistas para desvendar a sua estrutura.

O exemplo de um resultado de um alinhamento múltiplo entre várias sequências de nucleótidos semelhantes pode ser visualizado na Figura 17.

```
SalmoSalarMuscle     ATGGCTGAGGCATTCGCAGGCACATGGAGATGAATACAGTGA----
TetradonNigroviri    ATGGCCGAAGCTTTTGCGGGAACGTGGAGACGAGTAC---------
FundulusHeteroc      ATGGTCGAGGCTTTCGTGGGCACGTGGAGACGACTACAG-------
TaeniaSolium         ---ATGGAGCCATTCATCGGTACCTGGAGACAAAATCCGTAA----
RattusNorvegicus     ATGTGCGACGCCTTTGTGGGGACCTGGAGATGATTACNAGCCAAGG
DanioRerio           ATGGTTGACAAATTCGTAGGAACGTGGAGACGAGTACAGGCATGA-
XenopusLaevis        ATGGTGGATCAATTCGTGGGCTCCTGGAGATGAGTACAGGCATAA-
                           **    **    **  * ******  *   *
```

Figura 17 – Parte de um alinhamento múltiplo com sequências de nucleótidos. A última linha indica o grau de conservação de cada coluna com o '*' a indicar a conservação completa de uma determinada coluna.

Existem várias aplicações disponíveis para efectuar alinhamentos, sendo o mais popular o ClustalW [84] e o ClustalX. A diferença entre os dois é que o ClustalW aceita somente sequências com os caracteres correspondestes aos aminoácidos e nucleótidos, enquanto o ClustalX aceita todos os caracteres possíveis, alinhando qualquer tipo de sequência.

O ClustalW é disponibilizado como aplicação local, bem como um serviço *web* disponibilizado pelo EBI. Ao alinhar as sequências, o ClustalW apresenta as várias sequências alinhadas criando também um histograma com a percentagem de alinhamento por coluna. Uma árvore que relaciona a proximidade entre sequências é também criada. Contudo, o ClustalW tem uma versão que poderá correr em *clusters* de computadores com o nome de ClustalW-MPI [85]. Quando é necessário efectuar alinhamentos entre muitas sequências a capacidade de processamento de um simples computador não é suficiente para efectuar os alinhamentos, recorrendo, sempre que possível ao ClustalW-MPI.

Existem outras aplicações que possibilitam efectuar alinhamentos múltiplos, como o caso do bioEdit [86]. Esta aplicação oferece ao utilizador a possibilidade de fixar certas posições nas sequências, efectuando posteriormente o alinhamento sem que essas posições sejam alteradas. Assim sendo, permite o congelamento de certos domínios conhecidos, alinhando as restantes partes em torno desses domínios.

O GeneDoc [87] também é uma ferramenta de alinhamento múltiplo, muito similar ao bioEdit. Tem a vantagem de assinalar a estrutura primária das proteínas desenhando também as árvores que representam a proximidade entre sequências.

O GCG Wisconsin Package [59] agrega um conjunto muito vasto de aplicações para bioinformática. Estas aplicações estão desenvolvidas para Linux e abrangem áreas distintas. Contém aplicações para efectuar alinhamentos múltiplos, predição e visualização de sequências na estrutura primária, procura de sequências semelhantes em base de dados, construção de árvores filogenéticas, entre outras soluções. Estas aplicações estão disponíveis através da *web*, em modo gráfico através do *X-Window* e também em linha de comando.

CLC RNA workbench [82], como já referido anteriormente, é um produto comercial, oferecendo um vasto conjunto de ferramentas para análise sequências, disponibilizando também uma excelente forma de trabalhar com alinhamentos múltiplos. É possível fixar posições em determinadas colunas ou inserir espaços entre letras, produzindo seguidamente os alinhamentos múltiplos em torno dessas condicionantes.

As quatro primeiras aplicações apresentadas não são soluções comerciais, estando disponíveis na *web* para livre utilização. Tanto o ClustalW como o GeneDoc disponibilizam o código fonte para permitir alterações ou melhoramentos.

3.6 Ferramentas disponíveis para redesenho de genes

A área de sintetização de genes para posterior produção de proteínas será uma das áreas que mais ganhará com a boa compreensão da estrutura primária dos genes. Actualmente existem várias aplicações para manipular sequências genéticas, recorrendo aos parâmetros mais conhecidos para caracterizar genes, podendo posteriormente manipulá-los aproximando-os dos parâmetros do organismo onde vão ser expressos.

As aplicações comerciais existentes são normalmente muito ricas a nível gráfico mas foram tipicamente construídas para analisar sequências, não para as redesenhar, dando pouca liberdade ao utilizador para o fazer. Existem algumas aplicações não comerciais criadas para o efeito, mas permitem somente alterar dois ou três parâmetros que caracterizam o gene.

Existe também um conjunto vasto de ferramentas *web* que redesenham genes sendo, regra geral, muito pobres a nível gráfico [88, 89]. Com as possibilidades técnicas associadas à *web 2.0* pode-se dar ao utilizador grande liberdade de interacção, as aplicações actuais vão pouco além do simples HTML.

Existem vários elementos a ter em conta no redesenho de genes. O mais importante é sem dúvida o *codon usage*. Este parâmetro descreve a frequência com que cada codão ocorre em relação aos codões sinónimos (codões possíveis pertencentes a um determinado aminoácido). O *codon usage* é um factor que está implicado aos baixos níveis de expressão genética, mas também no *frameshift* (deslocamento de uma ou mais bases no processamento de tradução), sendo o factor mais importante nos níveis de expressão genética nos organismos procariotas [90]. Cada organismo tem a sua tabela de *codon usage* como foi explicado mais em detalhe no capítulo 2.

Outro ponto a ter em conta está relacionado com áreas de restrição. As áreas de restrição, são pequenas sequências de nucleótidos onde as enzimas de restrição se ligam para cortar o DNA [91]. Estas enzimas existem nas bactérias e arqueas e fornecem mecanismos de defesa contra possíveis vírus que invadam o seu DNA [92]. Para cortar o DNA, a enzima de restrição efectua duas cisões, uma em cada cadeia na dupla hélice (Figura 18). Se as sequências que são reconhecidas pelas enzimas de restrição aparecerem nas sequências do gene sintetizado, este gene irá ser cortado antes de ser expresso pelo organismo. A REBASE (*Restriction Enzyme Database)*, é a base de dados que concentra todas as sequências com as áreas de restrição [93]. Outros factores associados à redução da expressão [94] são as sequências Shine-Dalgarno [13]. Todos estes factores têm de ser tidos em conta pelas aplicações que se proponham a sintetizar genes.

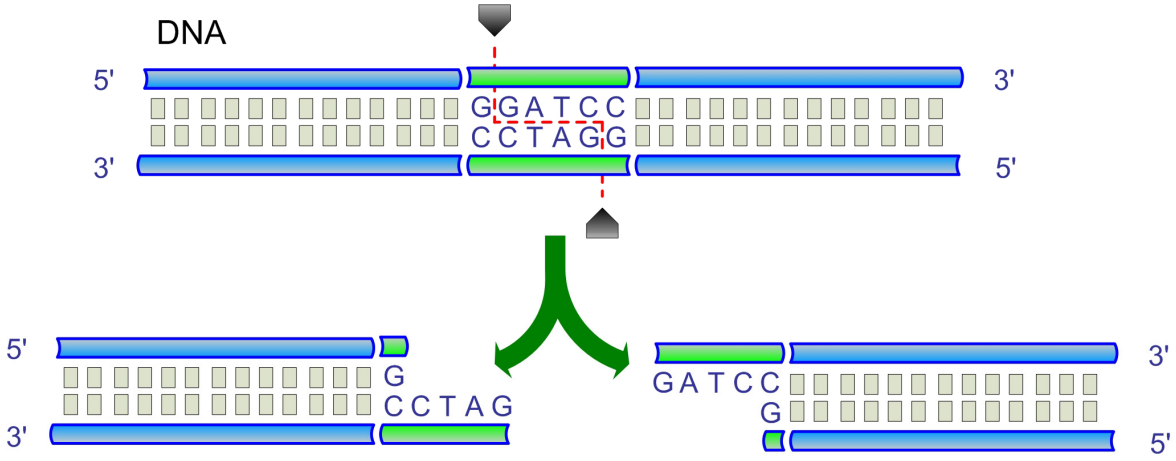

Figura 18 – Processo de corte através da enzima de restrição BamH1.

A ferramenta GeneDesigner [95] é uma aplicação comercial, desenvolvida somente para o redesenho de genes, dando grande liberdade de escolha de codões na construção da sequência final. Além de possibilitar alterar vários parâmetros no gene, tem um ambiente muito rico a nível gráfico. Consegue melhorar a sequência recorrendo à optimização de vários parâmetros. O *codon usage*, um dos mais importantes parâmetros, é melhorado através do CAI [31], tentando igualá-lo a 1 (valor máximo) - quer isto dizer que utiliza por aminoácido os codões com mais ocorrência no organismo. Tem o problema de gerar altas concentrações para codões específicos, alterando o balanceamento das concentrações de tRNA, aumentando a probabilidade de erro [96]. Está demonstrado que a parte inicial da ORF é muito importante para uma boa tradução [97-99], logo é possível separar a parte inicial do gene da restante e trabalhar com ela separadamente. Também identifica os *stem-loop* com mais de 12 pares de base e tenta eliminá-los, pois quando estes ocorrem nas sequências existe um atraso na evolução do ribossoma [100]. Também recomendam incluir dois codões stop no final de gene [95]. É também possível usar modelos probabilísticos, para implementar a opção *'optimize close to'* ou *'far away from'* em relação a uma pequena sequência de referência. Tem em conta as áreas de restrição e as sequências de Shine-Dalgarno [94]. É sem dúvida uma das melhores e mais completas soluções disponíveis para o redesenho de genes.

O DNAWorks [101] é uma aplicação web desenvolvida para ajudar na criação de genes sintéticos. O princípio de funcionamento é um pouco diferente dos restantes. A sequência de entrada é uma sequência de aminoácidos e não de nucleótidos. A partir dos aminoácidos são escolhidos os codões com base numa tabela de *codon usage* seleccionada de entre

muitas possíveis. Temperaturas de *self-annealing* [102] são também definidas pelo utilizador, assim como áreas de restrição a evitar na formação da sequência final. Tem como característica principal a facilidade de utilização, mas não permite grande interacção com a sequência a sintetizar.

OPTIMIZER [103] é uma aplicação *web* desenvolvida em PHP que optimiza o *codon usage* nas sequências submetidas, entre outros parâmetros. As tabelas de *codon usage* podem ser escolhidas entre as várias disponíveis ou podem ser introduzidas pelo utilizador. Três modos de optimização estão disponíveis: i) um aminoácido – um codão; ii) uma aproximação aleatória; iii) modo intermédio. Na primeira opção, um aminoácido por um codão, o algoritmo escolhe o codão sinónimo com o índice mais elevado na tabela de *codon usage*. Na segunda opção, na aproximação aleatória, o algoritmo escolhe aleatoriamente o codão sinónimo para cada posição, com as probabilidades baseadas na tabela de *codon usage*. A última opção permite ao utilizador escolher qual o codão a colocar em cada posição, baseado na tabela de *codon usage*. O algoritmo permite também evitar as áreas de restrição.

O GeMS [104] é uma aplicação *web*, disponibilizando também uma versão para trabalhar localmente, que permite alterar vários parâmetros no gene. Permite definir áreas de restrição, prever *stem-loop*, optimizar o *codon usage* baseado na tabela seleccionada, a separação de longas sequências para trabalhar individualmente, entre outras possibilidades. O resultado é dado como um relatório das opções tomadas para chegar à sequência final. O interface é de fácil utilização, permitindo ao utilizador menos experiente interagir facilmente com a aplicação, contendo sempre instruções sobre quais as opções a tomar.

3.7 Identificar tRNAs

A identificação de tRNAs no genoma de determinada espécie será certamente uma das primeiras tarefas a realizar depois de sequenciar um organismo. Para tal, existe uma aplicação, com o nome de tRNAScan-SE, que é amplamente usada para identificar os tRNAs [105]. Esta aplicação não é uma aplicação de procura de tRNA, mas sim uma aplicação desenvolvida em *perl* que interliga três aplicações para detecção de tRNA, obtendo assim o melhor resultado possível. Basicamente, o tRNAScan-SE tem como entrada sequências genéticas, passando-as numa primeira fase pela aplicação EufintRNA e

pela aplicação tRNAScan 1.3, devido a apresentarem uma aceitável velocidade de procura. Embora as aplicações anteriores tenham uma elevada taxa de falsos positivos, os resultados serão os dados de entrada para os passos seguintes. Posteriormente, um modelo de covariância é aplicado nos resultados anteriores. Embora de baixo desempenho, mas com uma taxa de falsos positivos reduzida e de verdadeiros positivos bastante elevada.

Tabela 3 – Resultados obtidos por diversas aplicações na procura de tRNA.

Nome das aplicações	(%) Verdadeiros positivos	Falsos positivos (por Mega bases)	Velocidade de procura (bases/s)
tRNAscan 1.3	95.1	0.37	400
EufindtRNA	88.8	0.23	373 000
tRNA covariance model search	99.8	<0.002	20
tRNAscan-SE	99.5	<0.00007	30 000

3.8 Conclusões

No presente capítulo foram apresentados diversos formatos disponíveis para a organização da informação genética assim como as bases de dados onde é possível obter a informação. Foram também apresentadas várias aplicações para extrair ou relacionar informação.

As ferramentas apresentadas cobrem diferentes áreas de estudo, como a extracção de índices, alinhamentos simples ou múltiplos, extracção de tRNAs e também ferramentas que possibilitam o redesenho de genes. Foram também identificadas variadas bases de dados genéticos passíveis de serem utilizadas no estudo das sequências genéticas.

No capítulo seguinte irão ser descritas as metodologias propostas para estudar a estrutura primária das zonas codificantes, tendo em conta as várias opções já existentes assim como as diversas aplicações possíveis de reutilização neste estudo.

Capítulo 4

---

4 Construção de um modelo de análise de estruturas primárias

4.1 Introdução

A sequenciação dos genomas abriu as portas para o estudo da estrutura primária das zonas codificantes para tentar perceber as forças que moldam a evolução das espécies. O *codon usage* foi intensivamente utilizado em vários organismos tendo algum sucesso na compreensão da estrutura primária. Contudo, outras características importantes, como o contexto de codões ou repetições do mesmo codão não estão totalmente compreendidas. O seu estudo para tentar compreender a influência na estabilidade do gene, eficiência e qualidade de tradução, torna-se imperativo.

O contexto de codões toma aqui um valor importante porque permite encontrar características até aqui não evidenciadas com os tradicionais índices associados ao *codon usage*.

Neste capítulo são apresentadas as metodologias desenvolvidas e as opções tomadas para estudar a estrutura primária das zonas codificantes dos genomas. São também apresentadas algumas ferramentas que foram necessárias a este estudo.

4.2 Análise do contexto de codões

O primeiro objectivo de estudo neste trabalho consistia em perceber se o contexto de codões, a vizinhança na posição *n-i* e *n+i*, tem influência na tradução do codão na posição *n*. Para efectuar esta primeira análise o sistema proposto contabiliza todos os pares de codões presentes nas regiões codificantes no genoma em estudo. Ao fazê-lo, constrói uma tabela de contingência de frequências de contexto passível de tratamento estatístico.

A análise de tabelas de contingência é uma metodologia estatística aplicada a dados de natureza qualitativa [106]. Os indivíduos de uma dada população podem ser classificados em categorias (ou classes) de acordo com diversos critérios. Fixados os critérios, a classificação dos indivíduos consiste em detectar as categorias nas quais cada indivíduo se identifica. As categorias a considerar são mutuamente exclusivas e a classificação é exaustiva, isto é, qualquer indivíduo pertence a uma e uma só categoria. Deste modo, os dados consistem nas frequências observadas em cada uma das categorias.

No caso concreto das tabelas a estudar, consideram-se dois critérios em linha, o codão fixo, e em coluna, o codão justaposto obtido por uma leitura 3', de acordo com o esquema da Figura 19. Uma vez que aos codões terminais não lhe sucede nenhum codão, a tabela será 61 × 64. As frequências são, naturalmente, o número de vezes que cada par de codões surge nas regiões codificantes da espécie em estudo.

As tabelas de contingência foram a metodologia escolhida para organizar e reduzir os dados relativos às sequências de símbolos (codões ou aminoácidos), já que se pretendia, numa primeira fase, uma análise global do genoma através do contexto de pares de símbolos justapostos. A informação contida na tabela permitirá realizar uma análise da associação entre pares de símbolos.

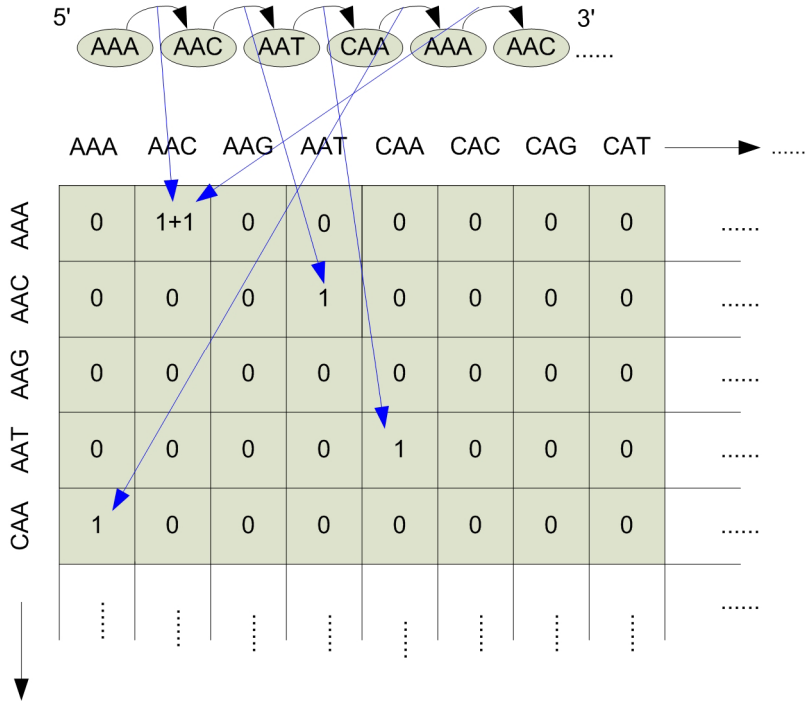

Figura 19 – Quantificação da frequência de pares de codões. Na parte superior é apresentada uma sequência hipotética. Fixando um codão faz-se corresponder o mesmo na linha da tabela e o seguinte na coluna da tabela.

Por convenção, fixando um codão qualquer o codão anterior diz-se que está a 5' (cinco linha) e o codão posterior a 3´(três linha). Assim, podemos ter tabelas de contingência 5´ ou 3´, consoante o sentido da contagem. Neste caso, estamos perante uma tabela 3' porque a contagem é feita no sentido 3'.

4.2.1 Análise residual

Para estudar as tabelas de contingência efectuamos uma análise de resíduo, face à rejeição da independência através da aplicação do teste do Qui-quadrado para tabelas de contingência, utilizando os valores residuais sugeridos por Haberman [107]. Os testes de independência e análise de resíduos em tabelas de contingência são encontrados em literatura especializada na análise de dados categóricos [108]. Se o teste realizado, no contexto das tabelas de contingência, concluir a rejeição da independência das duas variáveis, terá interesse analisar as classes que provocaram tal rejeição. Assim, no sentido de identificar as categorias responsáveis pelo valor elevado da estatística de Pearson, pode-se realizar uma análise dos resíduos ajustados, também conhecidos por resíduos de Pearson [109] dados por:

$$r_{ij} = \frac{\left(n_{ij} - e_{ij}\right)}{\sqrt{e_{ij}}} \tag{4}$$

onde e_{ij} é dado por:

$$e_{ij} = \frac{n_{i\bullet} n_{\bullet j}}{N} \tag{5}$$

Os valores $n_{i.}$, $n_{.j}$ e N correspondem ao total da linha i, total da coluna j e total de todas as linhas correspondentemente.

Neste estudo iremos utilizar os resíduos ajustados dados por:

$$d_{ij} = \frac{e_{ij}}{\sqrt{v_{ij}}} \tag{6}$$

onde v_{ij} é uma estimativa da variância de r_{ij} dada por:

$$v_{ij} = \left(1 - \frac{n_{i.}}{N}\right)\left(1 - \frac{n_{.j}}{N}\right) \tag{7}$$

Observe-se que, se os resíduos ajustados, d_{ij}, tiverem um comportamento muito diferente da distribuição normal $N(0, 1)$ será de rejeitar a independência na tabela de contingência. Uma vez que d_{ij} tem distribuição normal $N(0, 1)$ poder-se-á dizer, que para um nível de confiança de 99.73%, as categorias mais responsáveis pela rejeição da hipótese da independência correspondem às células (i, j), tais que $|d_{ij}| \geq 3$, uma vez que $P(-3 < d_{ij} < 3) = 0.9973$ [110]. Assim, na prática considera-se que um valor de resíduo ajustado é responsável pela rejeição se em módulo for igual ou superior a 3.

Algumas metodologias matemáticas foram usadas para estudar o contexto de codões [37, 111-113]. Estas metodologias são baseadas no teste *z-score* e fornecem informação sobre preferência e rejeição de independência, diferindo do modelo probabilístico assumido pelo nosso estudo. No entanto, a comparação entre a análise residual e os resultados obtidos através do teste *z-score* encontraram o mesmo comportamento entre os diferentes modelos usados, evidenciando os mesmos pares de codões com significado estatístico.

A vantagem da metodologia de análise de resíduos ajustados reside no facto que a sua teoria de inferência é facilmente interpretável e dispõe de mais ferramentas complementares de análise (por exemplo, medidas de associação). Mais ainda, o resíduo ajustado dá informação imediata sobre preferência e rejeição em relação ao que seria

esperado. Além disso, a sua probabilidade de distribuição, sob a hipótese de independência na tabela de contingência é determinada sem simulações técnicas [113], por exemplo, utilizando a técnica Monte Carlo para estimar a distribuição de parâmetros.

A Figura 20 contém um pequeno resumo de todos os processos desde a contagem dos pares de codões até à sua representação visual. Nesta representação propõe-se uma codificação cromática que faz corresponder cores a determinados níveis de resíduos ajustados. No passo um é feita a contagem dos pares de codões de todas as sequências pertencente ao genoma em estudo. No passo dois é efectuado a transformação dos valores de contagem, guardados na tabela de contingência, em valores residuais ajustados com a sua posterior representação baseada numa escala de coloração, e finalmente o passo três, a coloração das sequências genéticas baseado nos valores da tabela da contingência.

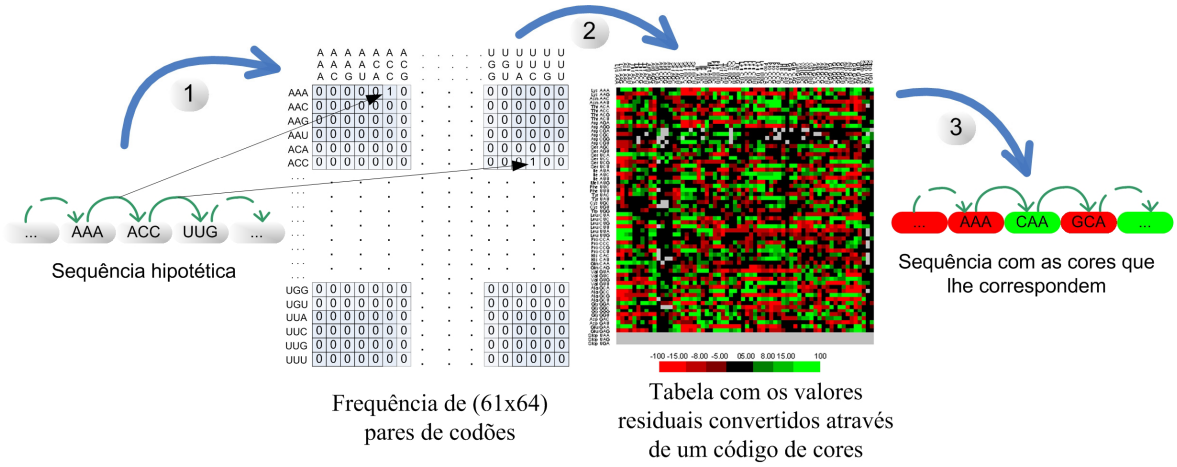

Figura 20 – Processo completo entre a contagem dos pares do codões, análise residual, atribuir limites e colorindo esses limites passando essa informação para as sequências para posterior análise.

Os valores da tabela de contingência vão-se transformar em valores de resíduos ajustados, fazendo-os variar entre os intervalos -∞ e +∞. Perante a dificuldade de visualizar tabelas contendo valores numéricos, decidimos atribuir cores para representar esses valores. Atribuímos cores com tonalidade encarnada aos valores negativos, inferiores a -3, e cores com tonalidades verdes aos valores positivos superiores a 3. Aos valores não superiores ao modulo de três, responsáveis pela não rejeição, serão coloridos a preto, não tendo significado estatístico. O cinzento identifica a ausência de contagens de pares de codões.

Partindo deste modelo pretende-se estudar correlações entre os pares de codões e provar que o seu contexto influencia os organismos no processo de tradução e sua evolução. Todas as análises centrar-se-ão no contexto de codões sendo por isso a base do nosso estudo.

4.3 Análise classificatória

De acordo com o objectivo geral deste estudo, o de averiguar a existência de leis que regem o código genético, serão aplicados métodos de análise classificatória com o objectivo de identificar grupos de indivíduos semelhantes quanto à preferência de vizinhos justapostos.

Os dados a considerar, na análise classificatória, serão tabelas de resíduos ajustados relativos aos pares de codões justapostos. A característica coluna das tabelas a estudar refere-se ao primeiro codão do par. As categorias, de um modo geral no âmbito da análise classificatória, designam-se por indivíduos. A característica linha refere-se ao segundo codão do par e as componentes (categorias) constituintes desta característica designam-se de variáveis. Nas tabelas a estudar contabiliza-se um total de 61 codões (variáveis) por um conjunto de 64 codões (indivíduos).

Portanto, para explorar os dados obtidos nas tabelas de contingência, poderemos agrupar linhas e colunas usando metodologias de classificação, como análise de *clustering* [114]. Estes padrões são obtidos através do cálculo de similaridade entre duas linhas ou colunas da tabela contingência usando, por exemplo, os coeficientes da correlação Pearson e aplicando a ligação simples (Figura 21).

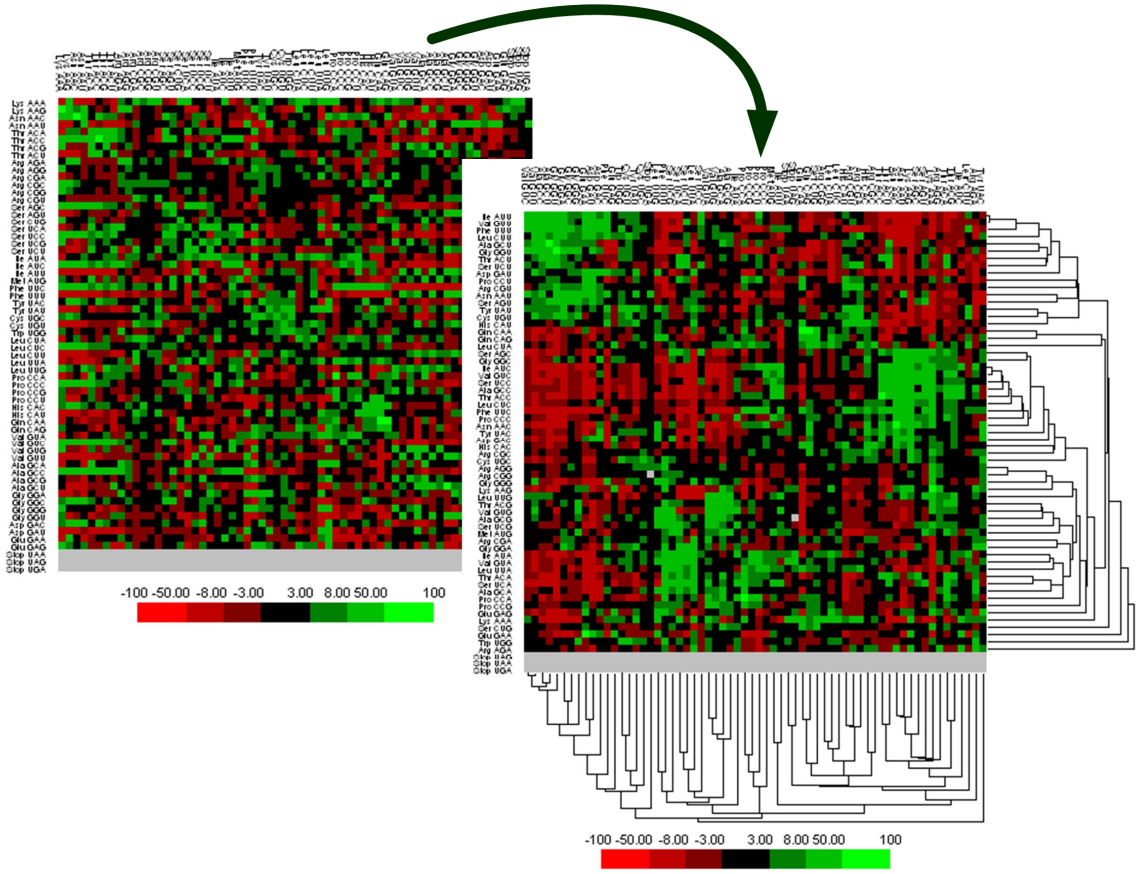

Figura 21 – Aplicação da análise classificatória a uma matriz contendo os resíduos ajustados. Poderemos ver a formação de padrões que serão objecto de estudo para detectar afinidades entre pares de codões.

4.3.1 Análise de *Clustering*

De modo geral os métodos de análise classificatória, ou *clustering*, encontram-se definidos na literatura especializada na seguinte forma [115]:

Dado um conjunto de n indivíduos para os quais existe informação sobre a forma de N variáveis, o método de análise de clustering procede ao agrupamento dos indivíduos em função da informação existente, de tal modo que os indivíduos pertencentes a um mesmo grupo sejam tão semelhantes quanto possível e sempre mais semelhantes aos elementos do mesmo grupo do que a elementos dos restantes grupos.

As técnicas a considerar efectuam uma classificação exclusiva e intrínseca, isto é, o resultado depois da aplicação do método define uma partição do conjunto inicial, ou de forma mais explícita, o resultado é obtido através de uma hierarquizada de partições. De

facto, esta análise classificatória consiste na construção de uma hierarquia de grupos de indivíduos semelhantes face a um conjunto de variáveis.

Duas escolhas são necessárias na construção da hierarquia de grupos de indivíduos: i) medida de semelhança entre indivíduos; ii) critério de agregação entre grupos. É conhecido que a utilização de diferentes medidas de semelhança e/ou diferentes critérios de agregação pode levar a resultados distintos. Assim, os resultados a obter podem depender da medida e do critério a utilizar. No entanto, o ideal na análise seria que os resultados obtidos, segundo diferentes escolhas, fossem idênticos independentemente dos métodos utilizados.

Existem variadas formas de calcular as distâncias de semelhança, apresentando-se de seguida três dessas medidas tendo características diferentes consoante a sua aplicação.

Dados dois conjuntos $X = \{x_1, x_2, ..., x_n\}$ e $Y = \{y_1, y_2, ..., y_n\}$ e considerando que todos os elementos dos vectores têm o mesmo peso, o coeficiente de correlação de Pearson centrado r é definido por:

$$r = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{X_i - \overline{X}}{\sigma_x} \right) \left(\frac{Y_i - \overline{Y}}{\sigma_y} \right) \qquad (8)$$

onde $\overline{X}$ e $\overline{Y}$ representam as médias de X e Y, e σ_x e σ_y representam os desvios padrão de X e Y respectivamente.

Os coeficientes de Pearson estão compreendidos no intervalo de [-1 .. 1]. Para r igual a zero os vectores não estão associados e para |r| igual a 1 estão perfeitamente associados, indicando o sinal o tipo de direcção da associação. A título de exemplo, tendo dois vectores múltiplos o coeficiente de correlação é 1.

Também pode ser usado como medida de semelhança, o coeficiente de correlação não centrado, dado por:

$$r = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{X_i}{\sqrt{\frac{1}{N} \sum_{i=1}^{N} (X_i)^2}} \right) \left(\frac{Y_i}{\sqrt{\frac{1}{N} \sum_{i=1}^{N} (Y_i)^2}} \right) \qquad (9)$$

Esta correlação é semelhante à centrada, assumindo que a média do vector é 0 mesmo quando não o é. Uma das diferenças surge quando temos dois vectores com a mesma forma

mas tendo “offset”; a distância de Pearson centrada seria de um enquanto a distância não centrada será inferior a um.

Cada medida tem características particulares e consoante a situação em estudo torna-se mais ou menos adequada. Contudo, no estudo de dados multivariados, é difícil identificar a melhor medida, passando por aplicar várias medidas aos dados em estudo e tentando perceber qual apresenta melhores resultados.

Apresentada a forma de cálculo da medida de semelhança entre vectores, o passo seguinte passa por construir uma árvore que agrupe todos os vectores em análise. Os métodos de agrupamento a apresentar são métodos hierárquicos que resultam em hierarquias de partições. Os métodos hierárquicos podem ainda subdividir-se em métodos aglomerativos e divisivos. A aplicação dos métodos aglomerativos determina inicialmente uma partição com tantas partes quanto o número de diferentes vectores, enquanto que nos métodos divisivos considera como ponto de partida uma partição com uma só parte, um único conjunto a que pertencem todos os vectores.

Dos métodos hierárquicos apenas se apresentam os métodos aglomerativos. Na realidade são os métodos aglomerativos os mais usados e mais divulgados na literatura. Um dos motivos para tal é o esforço computacional ser inferior face aos divisivos apresentando o mesmo tipo de resultados.

Para os métodos aglomerativos, definem-se vários critérios de agregação distinguindo-os apenas as diversas formas existentes de relacionar as distâncias entre grupos. Exemplos de critérios de agregação são: i) ligação simples; ii) ligação completa; iii) critério da média; iv) critério do centróide; v) método de Ward [114].

Não se consegue afirmar que existe um critério de agregação que seja melhor em relação a outro. Na prática, o que se faz é utilizar vários critérios e comparar os resultados. Se os resultados de diferentes critérios forem concordantes o resultado final será mais credível.

O processo de agrupamento ou de agregações pode ser representado por um diagrama bidimensional em forma de árvore conhecido por dendograma. No eixo dos xx estão representados os indivíduos e no eixo dos yy as distâncias (Figura 22). O dendograma tem a vantagem de facilitar a visualização do processo de agrupamento nas suas diversas fases,

desde os vectores separados até à inclusão num só grupo. Na realidade o dendograma é uma representação gráfica da hierarquia de partições.

Os ramos da árvore que constituem o dendograma identificam-se com as n−1 ligações entre grupos, onde cada linha horizontal representa uma ligação a que normalmente se chama de nodo. Nos métodos aglomerativos a primeira ligação identifica-se com a menor ramificação e a segunda ligação com a segunda menor ramificação e assim sucessivamente.

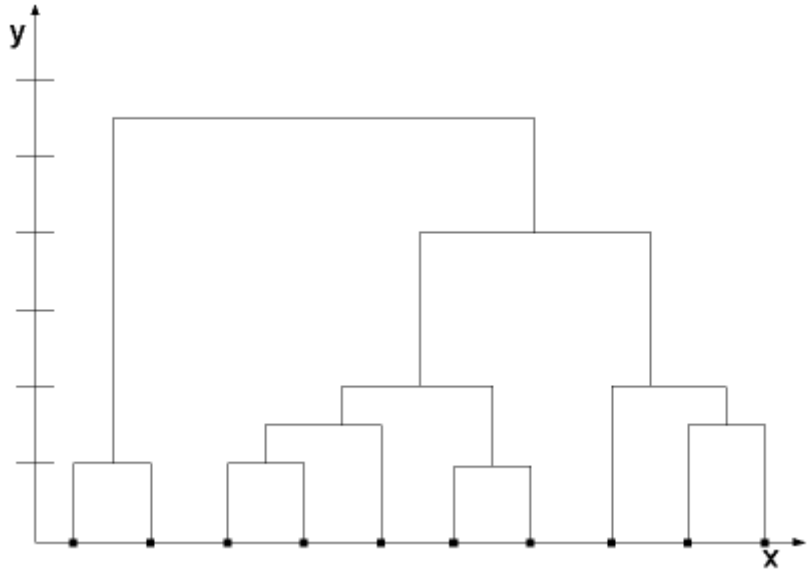

Figura 22 – Dendograma que representa uma estrutura bidimensional em forma de árvore

Se o número de grupos for conhecido à partida, a identificação dos grupos é quase imediata, à custa do dendograma. Caso contrário, a observação do dendograma pode sugerir uma estimativa para o número de grupos, mas nem sempre essa escolha é objectiva. A escolha do número de grupos poderá ser ainda menos objectiva se ao utilizar diferentes critérios de agregação forem obtidos dendogramas que sugiram diferentes partições. A determinação do número de grupos é feita cortando horizontalmente o dendograma. A título de exemplo, no dendograma da Figura 23 pode-se propor duas partições: uma correspondente à linha 1 e a outra à linha 2, com 3 e 2 grupos, respectivamente. A escolha da melhor partição, em particular do número de grupos óptimo, deverá ser feita mediante o contexto do problema, pelo que o conhecimento prévio da natureza dos dados pode auxiliar nesta decisão.

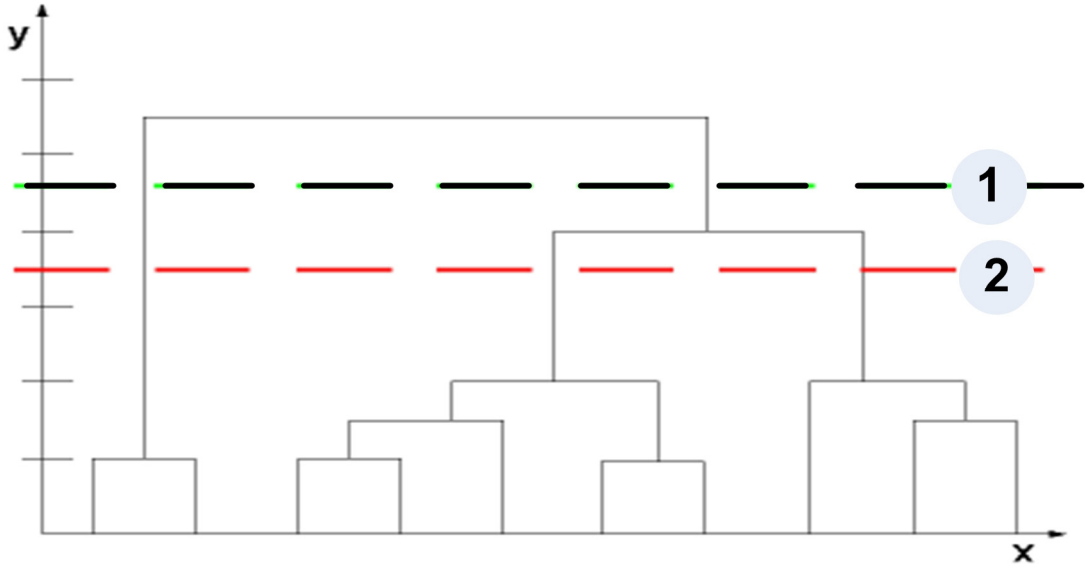

Figura 23 – Dendograma que realça duas escolhas possíveis do número de grupos

Existem outras técnicas para fazer a análise de *clustering* como por exemplo o método K-mean [116]. Neste método é necessário iniciar o agrupamento com o conhecimento do número de grupos, k, em que se pretende subdividir o conjunto de vectores. É conjuntamente necessária uma caracterização inicial de cada um dos grupos, a que se chamam posições iniciais dos centros do grupo. As posições iniciais podem ser escolhidas aleatoriamente, podendo ser escolhidas com informação à *priori* de forma a ganhar uma rápida convergência.

A crítica comum atribuída a este método é que o número de grupos tem de ser definido no início do processo. Quanto ao algoritmo não existe prova de convergência, mas na prática converge num número finito de passos. Não há garantia de convergir para a solução óptima e a escolha da configuração inicial influencia a solução obtida [117].

Actualmente está disponível uma biblioteca desenvolvida em C++, com o nome "*The C clustering library*", que disponibiliza vários métodos para implementar as diversas formas de agrupamentos [118]. Não contempla a parte da construção gráfica dos dendogramas, sendo necessário recorrer a outras aplicações para os construir. É de notar que esta biblioteca é igualmente disponibilizada em *python* e em *perl*.

Existem várias ferramentas para obtenção e/ou visualização de *clustering* como o Mev4 ou o Java TreeView, ambas escritas em Java, que tem a vantagem de ser multi-plataforma.

4.3.2 Análise de Biclustering

O método de agrupamento foi inicialmente considerado por Eisen para revelar a informação biológica [119], aplicando-o a tabelas que continham valores de expressão genética. Posteriormente, vários outros métodos de *clustering* foram desenvolvidos para aplicar ao mesmo tipo de dados [120]. Contudo, a análise de *clustering* contém algumas limitações. Numa primeira análise o *clustering* só se pode aplicar separadamente a um dos eixos, não tendo em conta a informação contida no outro eixo. Mais ainda, se um determinado vector for inserido num determinado grupo, esse vector já não será inserido em outro grupo formado posteriormente, mas que poderia ter alguma proximidade com o vector anterior. Perante estas limitações a técnica de *biclustering* pode ser usada para colmatar estas desvantagens [121]. Estes algoritmos criam grupos considerando simultaneamente, as duas dimensões da matriz, permitindo a sobreposição de vectores, pois formam vários grupos simultâneos [122]. Quer isto dizer que a análise *clustering* dá uma resposta global, enquanto a análise de *biclustering* dá uma resposta local.

O método de *biclustering* é de complexidade NP-completo [121] e não existe uma solução óptima para a procura de padrões. Existem vários algoritmos e cada um deles tem as suas vantagens e desvantagens para responder melhor a determinados dados. Teremos de ter em conta que uma só abordagem pode não identificar todos os padrões relevantes em dados de maior complexidade [123, 124].

Existem vários algoritmos disponíveis, sendo os mais comuns:

- Bimax - utiliza o método dividir para conquistar [124];
- Cheng e Church's (CC) - é baseado na média quadrada dos resíduos [121];
- *Contiguos column coherent biclusters* (CCC-Biclusters) – baseia-se numa árvore de sufixos para identificar grupos [125];
- *Iterative Signature Algorithm* (ISA) - procura sub matrizes que representam pontos fixos [126];
- *Orderpreserving Submatrix* (OPSM) - tenta identificar as maiores sub matrizes para as quais induz uma ordem linear das colunas que é idêntica a todas as linhas [127];
- *xMotif* - identifica os grupos com um método iterativo [128].

A aplicação Bicat permite efectuar *biclustering* sobre dados disponibilizados em matriz, assim como a visualização dos resultados [129]. Disponibiliza vários algoritmos de *biclustering*, como o ISA [126], xMotifs [128], OPSM [127], Bimax [124] e o CC [121].

Devido às limitações de *clustering* optou-se por introduzir no modelo um algoritmo de *biclustering*, mais concretamente o algoritmo ISA. A opção de implementar o algoritmo, em detrimento de o utilizar através de uma aplicação, deve-se ao facto de se poder optimizar ou alterar o algoritmo.

Assumirmos que X é uma matriz de dimensão n x m que contém valores reais:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ x_{31} & x_{32} & \cdots & x_{3m} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix} = [x_{ij}]$$

com n linhas R_1, R_2, …, R_n e m colunas C_1, C_2, …, C_m. O ISA inicia-se com um determinado número de linhas, podendo ser escolhidas aleatoriamente, e aplica iterativamente o algoritmo assinatura [130]. O algoritmo é processado em duas etapas, normalizando a matriz X por colunas na primeira etapa e por linhas na segunda etapa. Na primeira etapa, considerando um determinado número de linhas, as colunas que tenham uma média fora do intervalo $\zeta_x =]a_{1x}, a_{2x}[$, são escolhidas para passar à etapa seguinte. Os parâmetros a_{1x} e a_{2x} tomam os valores $(\hat{u}-t_x\sigma)$ e $(\hat{u}+t_x\sigma)$, respectivamente. Os valores t_x e t_y são definidos pelo utilizador e os valores $\hat{u}$ e σ são a média e o desvio padrão das linhas ou colunas em análise. Na segunda etapa, e de forma análoga à anterior, as médias são calculadas para as linhas, através das colunas seleccionadas anteriormente, e escolhidas de forma a estarem fora do intervalo $\zeta_y =]-\infty, a_y]$, com $a_y = (\hat{u}+t_y\sigma)$. Correndo o ISA de forma repetida obtemos vários subgrupos que poderão ou não sobrepor-se.

Devido à necessidade de identificar grupos presentes nas matrizes de valores residuais, foi realizada uma primeira abordagem com a aplicação BicAT. Devido à média ser uma medida bastante influenciada por valores extremos, foram detectados vários grupos indesejados. Se determinado grupo de colunas e linhas conter valores muito altos em

relação à média faz com que esse grupo se realce mesmo que alguns elementos dentro do grupo tenham valores muito reduzidos. Este problema já tinha sido identificado por estudo publicado anteriormente [131]. No entanto, este estudo não solucionava o problema devido a um dos intervalos rejeitar os valores negativos, mais propriamente o intervalo $\zeta_y =]-\infty, a_y]$.

Tendo em conta os constrangimentos apresentados, foi desenvolvido um novo algoritmo que identificamos por ISA-Mediana. A seguinte descrição resume os passos do novo algoritmo.

Entrada:

X: $n \times m$ matriz de valores reais;

$C = \{C_j, j = 1, \ldots, m\}$ – conjunto de m colunas;

$R = \{R_i, i = 1, \ldots, n\}$ – conjunto de n linhas;

$R^{(0)}$ = um conjunto inicial de $n_0 \leq n$ linhas seleccionadas aleatoriamente;

t_y – parâmetro que define um limite ás linhas;

t_x – parâmetro que define um limite ás colunas.

Primeira etapa:

Passo 1: inicializar $k = 0$

Passo 2: obter uma sub-matriz de X com as linhas $R^{(k)}$

Passo 3: calcular as medianas nas colunas S_{Cj}

Passo 4: calcular a média das medianas das colunas S_C

Passo 5: obter um sub grupo $C^{(k)}$ das colunas C_j que satisfaçam:

$$C^{(k)} = \{C_j \in C: g(S_{Cj}, S_C) > t_x \sigma_C\} \text{ , onde } \sigma_C = \sigma \sqrt{\frac{\pi}{2 \mid R^{(K)} \mid}}$$

Segunda etapa:

Passo 6: obter a sub-matriz de X com colunas seleccionadas anteriormente $C^{(k)}$

Passo 7: calcular as medianas nas linhas S_{Ri}

Passo 8: calcular a média das medianas das linhas S_R

Passo 9: obter um sub grupo $R^{(k+1)}$ das linhas R_j que satisfaçam:

$$R^{(k+1)} = \{R_i \in R: g(S_{Ri}, S_R) > t_y \sigma_R\} \text{ , onde } \sigma_R = \sigma\sqrt{\frac{\pi}{2 \mid C^{(K)} \mid}}$$

Passo 10: se $R^{(k+1)} \neq R^{(k)}$; $k = k+1$ e volta ao passo 2

Passo 11: bicluster = $[x_{ij}]_{\, i \in R^{(k)}, j \in C^{(k)}}$

Este algoritmo poderá nunca encontrar um sub grupo de *X*, tendo normalmente uma excepção. Se *k* for maior que determinado valor o algoritmo termina a procura sem identificar o grupo.

A desvio padrão σ é calculada através da equação $\sigma = \sqrt{\frac{\sum_{ij}\left(x_{ij} - \mu\right)^2}{n \times m - 1}}$ com $\mu = \frac{\sum_{ij} x_{ij}}{n \times m}$.

A função *g* é definida $g(x,y)= (x-y)$, $g(x,y)=-(x-y)$ ou $g(x,y)=|x-y|$ consoante a natureza dos grupos que procuramos, com valores altos, baixos ou absolutos respectivamente.

Comparando o algoritmo ISA-original com o ISA-Mediana, são apresentadas várias diferenças:

- nos passos 5 e 9 foi introduzida uma nova função onde é permitido procurar grupos com valores negativos, positivos ou em módulo, podendo efectuar-se combinações entre as três opções;
- não é necessário uniformizar as matrizes;
- nos passos 3 e 7 os valores não são multiplicados por pesos resultantes das iterações anteriores, o que acontecia com o ISA original;
- modificou-se a média pela mediana nos passos 4 e 8.

4.4 Comparar sequências semelhantes

O mapa de contexto de pares de codões de cada organismo é diferente devido às diferenças existentes nas sequências de DNA.

Durante o processo de evolução os organismos vão herdando algumas partes do código genético dos seus ancestrais. Algumas partes mantêm-se, enquanto outras se alteram devido às inserções, remoções ou arranjos de segmentos do código. Os segmentos que se alteram são responsáveis por diferentes mapas de contexto entre organismos, mesmo quando os organismos estão próximos na árvore evolutiva.

Para estudar a evolução dos organismos através do contexto de codões, mais concretamente para perceber a influência das regiões conservadas entre organismos, foram adicionadas ao modelo duas ferramentas muito usadas no universo da bioinformática. Estas ferramentas são os BLASTP [132] e o ClustalW [84]. Com a sua conjugação, será possível procurar zonas conservadas entre diferentes espécies. O BLASTP procura sequências homólogas entre organismos, baseado nos seus aminoácidos, enquanto o ClustalW faz o seu alinhamento. Existe uma primeira fase no processo, que teremos de converter as sequências de nucleótidos em sequências de aminoácidos. O BLASTP será aplicado às sequências de aminoácidos pois estamos interessados em homologia de proteínas. Posteriormente, processamos o resultado do BLASTP, retirando as coordenadas das zonas homólogas, para futuramente podermos alinhar com o ClustalW as sequências onde existe homologia entre diferentes organismos.

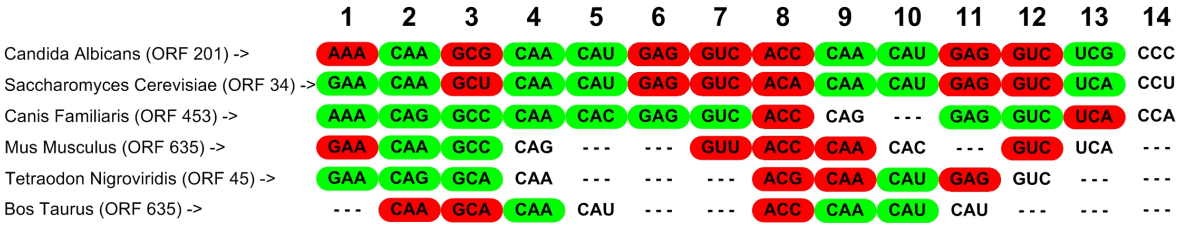

Figura 24 - Algumas sequências alinhadas provenientes de diferentes organismos.

Na Figura 24 poderemos ver o resultado da conjugação entre as duas ferramentas. Neste caso concreto o organismo *Cândida Albicans* foi o organismo usado como referência. Todos os genes pertencentes ao organismo *Cândida Albicans*, assim como os outros organismos escolhidos como alvo, já foram traduzidos para os seus aminoácidos correspondentes. O BLASTP vai percorrer cada gene pertencente à *Cândida Albicans* e procura sequências homólogas entre todos os outros genes pertencentes aos outros organismos em estudo. Depois de todo o processamento, teremos de compilar toda a informação e procurar zonas onde existe homologia entre diferentes genes na mesma posição mas pertencente a diferentes organismos. Posteriormente, alinhamos estas

sequências com o ClustalW para poderem ser visualizadas com os seus codões e cores obtidas através dos mapas de contexto. Por exemplo, na coluna 2 da Figura 24, temos diferentes codões, os quais codificam o mesmo aminoácido. A coluna 6 e 7 têm o mesmo par de codões para os três primeiros organismos e diferentes contextos. Os traços nas diversas posições indicam que o algoritmo de alinhamento teve de inserir espaços para um melhor alinhamento.

4.4.1 Algoritmo BLAST

Na procura de sequências similares existem sempre duas entradas importantes. A sequência de procura e as sequências alvo. A sequência de procura é fornecida pelo utilizador e vai ser alinhada contra todas as sequências alvo que residem em forma de base de dados previamente construída. A base de dados é o resultado de uma indexação de todas as sequências que se querem como referência, efectuada por uma aplicação associada ao algoritmo de procura.

O algoritmo BLAST tornou-se o método mais usado no campo das ferramentas de alinhamento de sequências [132]. O algoritmo ainda continua em evolução e é dos mais referenciados nos artigos na área da biologia. O algoritmo está disponível em código aberto sendo, por isso, possível alterar o seu código. Este procedimento faz com existam várias variações do BLAST, como o WU-BLAST desenvolvido pela Universidade de Washington.

A necessidade da computação no processo de alinhamento é inevitável, pois o número de símbolos possíveis pode ser elevado. Perante este facto, a procura por um alinhamento óptimo pode levar a uma explosão combinatória, tornando a sua execução extremamente demorada.

Existem duas grandes abordagens que se podem escolher para efectuar alinhamentos. O alinhamento global e local.

O alinhamento global de duas sequências efectua comparação simultânea, recorrendo, por exemplo, ao algoritmo Needleman-Wunsch [133]. O alinhamento global consiste na comparação dos símbolos por posição, e permitindo a utilização de um sistema de pontuação para avaliação do alinhamento obtido e a inserção de espaços para obter melhores resultados.

O objectivo de um alinhamento global é obter a maior pontuação possível ou óptima. O sistema mais simples consiste em dar uma pontuação por alinhar um espaço, *gap,* uma pontuação por alinhar dois símbolos diferentes, *mismatch,* e uma pontuação por alinhar dois símbolos idênticos, *match*, recorrendo às matrizes de substituição apresentadas no anexo 8.4 (Figura 25).

Neste exemplo específico, teremos um S=57, utilizando os valores de substituição da tabela BLOSUM62, que consta no anexo 8.4, e atribuindo um peso de 5 ao *gap*. Como podemos ter vários resultados para a mesma sequência, pois esta poder-se-á alinhar de forma diferente na mesma região, escolhemos o resultado mais elevado que obtivermos.

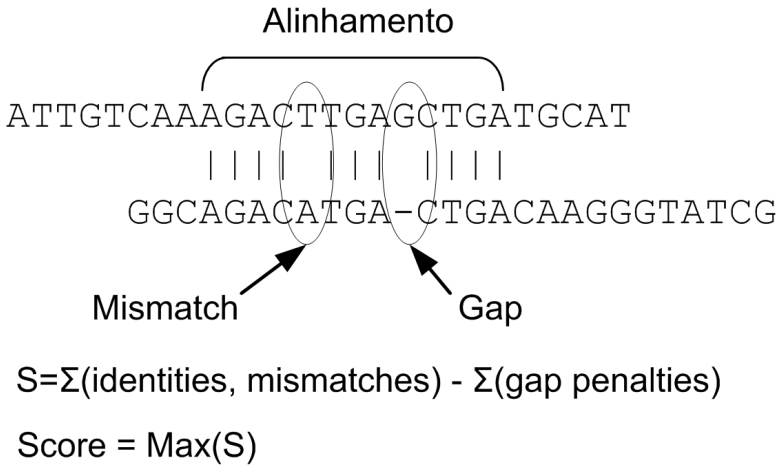

Figura 25 – Método de pontuação no alinhamento de uma sequência

O algoritmo do alinhamento global é de complexidade O(*mn*), onde *m* e *n* é o comprimento das sequências a serem alinhadas, sendo muito pesado a nível computacional. Por este motivo o algoritmo não é muito usado nas aplicações de alinhamento. No entanto, existe uma versão modificada do algoritmo que transforma a complexidade em O(*m*+*n*), reduzindo o custo computacional. Esta modificação é geralmente baseada em algoritmos dinâmicos [134].

O alinhamento local identifica as sequências através do método heurístico, sendo o método utilizado pelo BLAST, que utiliza uma versão modificada do algoritmo Smith-Waterman [135]. Inicialmente procura pequenas sequências semelhantes, isto é, não tenta alinhar a sequência por completo mas divide a sequência em pequenos pedaços e procede ao seu alinhamento.

Como se pode ver na Figura 26, o algoritmo, neste caso específico, está configurado para procurar palavras com um comprimento de três, retiradas da sequência de procura. A partir deste momento o algoritmo baseia-se nos melhores resultados para expandir a procura.

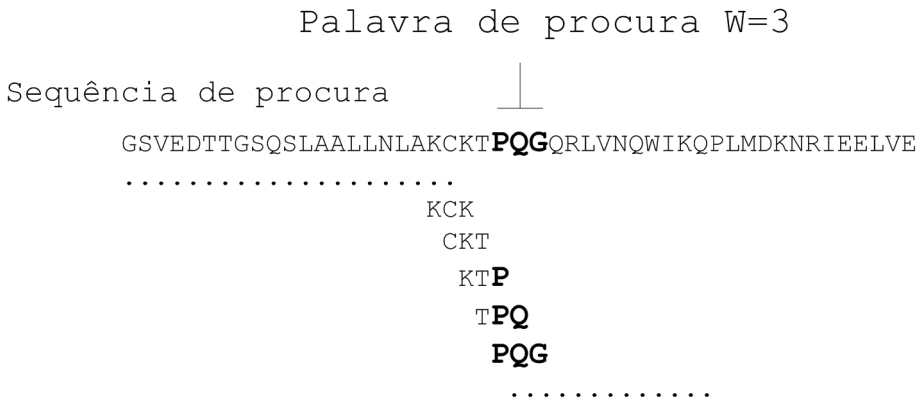

Figura 26 – Procura exacta de uma palavra com comprimento 3

Com as palavras iniciais identificadas, o algoritmo estende a procura nos dois sentidos, avançando três letras em cada iteração, como se pode ver pela Figura 27. Por cada vez que o algoritmo estende a procura em três letras o cálculo do *score* é efectuado, incrementando ou decrescendo consoante a sequência encontrada. Se o resultado for inferior a um determinado limite o alinhamento nessa região para. Este método assegura que o alinhamento não inclui regiões com baixa similaridade entre a sequência de procura e a sequência alvo. No momento em que não é possível progredir nos dois sentidos o valor esperado (*E-value*) é calculado. Se esse valor for inferior a um certo limite o alinhamento é adicionado aos resultados.

Figura 27 – Expansão do resultado de uma palavra.

Existem vários valores que quantificam a qualidade do alinhamento obtido, resultante do processo de alinhamento. Um dos mais importantes, e certamente o mais informativo, é o valor esperado (*E-value*). O *E-value* dá-nos a informação se um determinado alinhamento poderia ter ocorrido por acaso, relativamente ao tamanho da base de dados em análise, assim como em relação ao comprimento da sequência submetida. Devido a este facto, se fizermos uma procura em base de dados de diferentes tamanhos, o resultado irá ser

distinto. Por exemplo, se executarmos o BLASTP com a mesma sequência mas em duas bases de dados de proteínas distintas, como o UniProt e a PIR, verificamos que temos dois *E-values* diferentes. Quanto menor é o *E-Value*, melhor o resultado. A faixa de 0.001 a 0.0000001 é comummente utilizada para definir os alinhamentos de alta qualidade.

A identidade e similaridade são outros valores a ter em conta na análise dos resultados, como se pode ver na Figura 28. A identidade é a razão das igualdades que ocorrem no alinhamento pelo comprimento do alinhamento. A similaridade é a razão do total das igualdades mais pares de aminoácidos que partilham as mesmas propriedades pelo comprimento da sequência.

Figura 28 – Identidade versus similaridade

Outro dos valores obtidos é o *score*, sendo já referido anteriormente na Figura 25, que depende do algoritmo aplicado, pois existem várias formas de cálculo.

O método de alinhamento local não garante a melhor procura, mas tem um bom compromisso qualidade versus rapidez. O BLAST é um bom método para um primeiro estudo, pois dá uma boa indicação sobre os possíveis alinhamentos, quando o tempo é uma questão importante. No entanto, se o objectivo for obter resultados mais precisos e o tempo não for um factor importante, é melhor usar o algoritmo Smith-Waterman, ou uma aplicação que o inclua, como a aplicação SSEARCH [136] ou o MPsrch [74].

4.4.2 Algoritmos para efectuar alinhamentos múltiplos

As sequências de aminoácidos e de nucleótidos conseguem-se agrupar em famílias e as suas semelhanças podem ser detectadas através de alinhamento múltiplo. Este procedimento fornece informação sobre a conservação entre sequências, o que não se consegue fazer com o simples alinhamento de apenas duas sequências. Por exemplo, se um determinado aminoácido se mantém entre duas sequências podem ser somente um acaso. No entanto, se um determinado aminoácido, ou conjunto de aminoácidos, estiver

conservado num conjunto de sequências isto pode indicar que esse grupo de aminoácidos tem um papel significante na estrutura ou na função enzimática.

A Figura 29 contém várias sequências de aminoácidos alinhadas, conseguindo-se identificar as zonas conservadas. Os asteriscos indicam conservação total, os dois pontos indicam conservação de aminoácidos que pertencem a um determinado grupo que partilha determinadas características (Anexo 8.4), e o ponto indica a conservação de aminoácidos que pertencem a um outro grupo.

```
SalmoSalarMuscle      -----RGAALKGAGGAATTKRRAGGLACATGGGAWGTGRRA----
TetradonNigroviri     --RQALAAALKGAAGAPTTTRRGGGAACGTGGACWC---------
FundulusHeteroc       -TTGTLAAALKGAGGAPTTKRTGGGLACGTGGAGWRTP-------
TaeniaSolium          -----SRAFLRGAGCSATTKRTCGGTACCTGGAGWTYTTTYYYYY
RattusNorvegicus      -LPPRGEQFLNGACSSATTTGTGGGGACCTGGAGWCNAGCRPPTL
DanioRerio            LRRRTGREFLRGACAAATTKGTAGGAACGTGGAGWQRT-------
XenopusLaevis         -----AQRLLQGATAAATTKGTGGGLTCCTGGAGWPRRTTATAA-
                                .*:**  : **:   **  * ******
```

Figura 29 – Resultado de várias sequências alinhadas. Os asteriscos indicam conservação total, os dois pontos indicam conservação de aminoácidos que partilham determinadas características e o ponto indica a conservação de aminoácidos partilhando outras características.

O problema de alinhar múltiplas sequências é muito mais complicado do que alinhar somente duas sequências. Para diminuir a complexidade do problema, assim como o tempo de execução, métodos heurísticos são aplicados para a construção do alinhamento múltiplo.

O primeiro grande desafio no procedimento do alinhamento múltiplo é como classificar os diferentes alinhamentos, ou seja, qual a função a usar para calcular o peso dos diferentes alinhamentos. Partindo do princípio que as sequências têm uma história em comum, que estão relacionados através da sua filogenia, a função de atribuição de pesos deveria ter isso em conta. No entanto, não é simples, dado que aumenta o número de parâmetros consideravelmente. Por isso, é comum ignorar essa complexidade e assumir que as sequências não partilham relações, ou utilizam-se correcções heurísticas para compartilhar a ancestralidade.

O segundo desafio é encontrar o melhor alinhamento através de um peso dado por uma função. Para um par de sequências isso pode ser feito através de algoritmos dinâmicos, identificando o melhor alinhamento global. No entanto, este método não é viável, sendo somente utilizado quando se tem um número reduzido de sequências para alinhar. Para trabalhar com n sequências, a programação dinâmica requer a construção de uma matriz de

n dimensões, requerendo ainda a necessidade de pesquisa espacial, aumentando exponencialmente com o aumento de n, consumindo muito poder computacional, assim como memória.

Os métodos heurísticos começam por agrupar as sequências mais próximas, acrescentando progressivamente as sequências menos relacionadas ao alinhamento inicial. Devido a uma inferior necessidade de requisitos computacionais em comparação com métodos de alinhamento global, os métodos heurísticos são amplamente utilizados em vários programas que efectuam o alinhamento múltiplo. Devido a este facto a abordagem mais comum é, portanto, aplicar o método de alinhamento progressivo [137]. Este método faz uma primeira triagem das sequências alinhando-as mutuamente, construindo assim uma árvore onde dispõe as sequências baseada nas distâncias. Baseado na árvore começa por alinhar as sequencias mais próximas fechando o processo alinhando a sequências mais distantes. Esse processo é mostrado na Figura 30. O alinhamento progressivo é usualmente muito eficaz, mas sofre de um problema quando erros de alinhamento são efectuados no início não mais serão rectificados. Essas imperfeições permanecerão até ao fim. Contudo, no exemplo da Figura 30, pode haver informação importante nas sequências C e D que poderiam melhorar o alinhamento A e B , mas isto não poderá ser útil porque o alinhamento de A e B é efectuado de forma independente de C e D. Por outro lado, o método progressivo tem um bom compromisso entre o tempo e qualidade do alinhamento.

O algoritmo de alinhamento múltiplo mais comummente usado é do ClustalW [84], sendo construído sobre o método de alinhamento progressivo. É menos rigoroso no resultado, mas é o algoritmo que consome menos memória [138]. O método progressivo também é usado para optimizar, por exemplo o T-Coffee [139] tendo mais rigor no resultado que o ClustalW mas apresenta problemas quando se tenta alinhar mais de 100 sequências [138].

Existem variados parâmetros que se podem definir no algoritmo de alinhamento progressivo. Por exemplo, no ClustalW, poderá definir-se os valores dos *gap penalties* e escolher as matrizes de substituição para efectuar os alinhamentos, consoante o grau de relação que se quer impor ao alinhamento das sequências.

O método iterativo é outra das formas de efectuar alinhamentos múltiplos, executando de forma similar ao método progressivo, tendo a vantagem de continuamente considerar o alinhamento existente aquando da adição de novas sequências. Isso melhora a precisão dos

métodos progressivos. O algoritmo de alinhamento MUSCLE baseia-se em métodos iterativos [140]. Tem um bom compromisso entre rigor e tempo de computação, mas terá de ser diminuir esse rigor quando se pretende alinhar para cima de 1000 pares de bases, devido a custos de computação [138].

Actualmente, o mais promissor algoritmo de alinhamento múltiplo é a metodologia de algoritmos estatísticos [141, 142]. Estes algoritmos incorporam na função de cálculo de pesos, a informação subjacente da filogenia a que a sequência pertence e o uso de um modelo estocástico da evolução molecular, que permite comparar as diferentes soluções obtidas e assim tomar uma decisão com mais rigor.

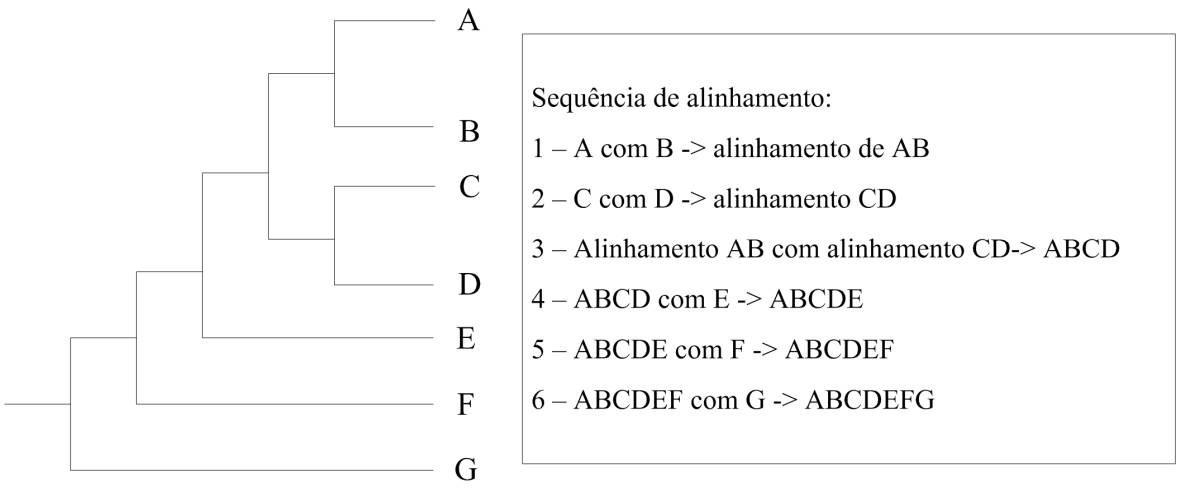

Figura 30 – Processo de alinhamento de múltiplas sequências com o método progressivo.

4.5 Optimização de genes

Um dos principais objectivos do estudo consistia na manipulação dos genes, sem alterar a sequência de aminoácidos, baseado nos dados obtidos através das tabelas de valores residuais e interagindo com outros índices já existentes. Basicamente, podemos substituir os diferentes codões pelos seus sinónimos, preservando os aminoácidos, sem alterar o produto final, a proteína. Pata tal, iremos usar parâmetros e técnicas, tais como: i) valores residuais; ii) valores RSCU; iii) *codon usage;* iv) remover codões raros; v) incrementar níveis de GC. Com esta aproximação, é possível adaptar um gene ao ribossoma anfitrião, melhorando assim a sua expressão genética.

Baseado nos requisitos anteriores, foi necessário desenvolver um algoritmo que permitisse alterar a sequência consoante as necessidades dos diferentes utilizadores.

Os vários passos a serem seguidos são:

Variáveis do algoritmo:

$G = \{ G_i, i=1,...,m\}$ – gene, m representa o comprimento do gene em codões;

t_{CAI} = peso do CAI, $0 \leq t_{CAI} \leq 1$;

t_{RSCU} = peso do RSCU, $0 \leq t_{RSCU} \leq 1$;

$t_{Residual}$ = peso dos valores residuais, $0 \leq t_{Residual} \leq 1$;

I_{GC} = aumentar níveis de GC;

I_{CR} = diminuir o número de codões raros;

K = número de pares alterados;

(A_i, A_{i+1}) = par de codões na posição i.

Etapas:

Passo 0: inicializar $k= 0$;

Passo 1: inicializar $i = -1$, $k_{last}= k$;

Passo 2: Incrementar i em uma posição;

Passo 3: voltar ao passo 2 se esta posição estiver bloqueada pelo utilizador;

Passo 4: analisar o par de codões (G_i,G_{i+1});

Passo 5: obter o peso $W_i = f((G_i,G_{i+1}), t_{CAI}, t_{RSCU}, t_{Residual})$;

Passo 6: obter o melhor par de codões alternativo $(A_i, A_{i+1}) = g(i, t_{CAI}, t_{RSCU}, t_{Residual}, I_{GC}, I_{CR})$;

Passo 7: obter o peso $WA_i = f((A_i,A_{i+1}), t_{CAI}, t_{RSCU}, t_{Residual})$;

Passo 8: se $WA_i > W_i$ possibilitar a alteração de (G_i,G_{i+1}) por (A_i,A_{i+1});

Passo 9: o utilizador pode concordar com esta alteração, bloquear a posição ou escolher outra com um peso inferior ao peso máximo;

Passo 10: se $i < m$ voltar ao passo 2;

Passo 11: Se $k < k_{last}$, o algoritmo pára e apresenta os resultados, caso contrário volta ao passo 1.

Os vários pesos, t_{CAI}, t_{RSCU} e $t_{Residual}$ podem assumir valores entre 0 e 1. Zero significa que o índice não irá entrar para o cálculo, e 1 significa que o índice tem peso máximo na escolha dos pares a serem alterados. Estes valores são definidos pelo utilizador no momento do redesenho do gene.

A diferença entre t_{CAI} e t_{RSCU} está no facto que o t_{CAI} é obtido através dos genes mais expressos do genoma enquanto o t_{RSCU} é obtido com a contabilização de todos os genes presentes no genoma.

A função f((G_i,G_{i+1}), $t_{CAI,}$ $t_{RSCU,}$ $t_{Residual}$) devolve um valor numérico que representa o peso que determinado par de codões tem. Este valor tem em conta os diferentes pesos definidos pelo utilizador, nomeadamente, t_{CAI}, t_{RSCU} e $t_{Residual}$. Para o CAI e o RSCU só se analisa o codão G_i, enquanto para o $t_{Residual}$ é analisado o par (G_i,G_{i+1}). Esta opção foi escolhida devido à análise 3' que se está a efectuar.

A função g(i, $t_{CAI,}$ $t_{RSCU,}$ $t_{Residual,}$ $I_{GC,}$ I_{CR}) devolve um par de codões que corresponde à melhor opção para a posição i. O I_{GC} e I_{CR} indica que se podem evitar codões raros ou dá-se preferência a codões com elevados níveis de GC.

Este algoritmo está desenhado para encontrar a melhor solução de forma automática, sem recorrer à intervenção do utilizador, nomeadamente a confirmação a cada alteração proposta pelo algoritmo. Pode-se ainda concentrar somente em zonas específicas, por exemplo modificar um número limitado de zonas de interesse.

4.6 Conclusões

No presente capítulo foram apontadas as opções tomadas para estudar as estruturas primárias dos genomas. Foi descrito o modelo desenvolvido baseado no contexto de codões, assim como a estatística necessária para extrair informação.

Foram ainda apresentados e discutidos diversos algoritmos que justificam a inclusão no modelo bem como as funcionalidades que se pretendem obter com a sua utilização

conjunta. Para além das ferramentas apresentadas, foram também explicadas as razões da escolha de determinados algoritmos para a extracção dos resultados pretendidos.

No capítulo seguinte será apresentada a aplicação desenvolvida para extrair informação com relevância biológica aplicando os algoritmos anteriormente apresentados. Será efectuada uma análise de requisitos e seguidamente será descrita a aplicação desenvolvida.

Capítulo 5

---

5 Anaconda

5.1 Introdução

A importância do erro associado à descodificação da informação genética, levou-nos a planear e a desenvolver um sistema informático que ajudasse a identificar possíveis leis gerais que governem a fidelidade de tradução dos genes. Para tal, foram desenvolvidas algumas metodologias matemáticas e algoritmos para análise de grandes volumes de informação genética bem como uma interface visual que ajudasse a interpretação dos resultados. A aplicação aqui apresentada foi desenvolvida para facilitar a compreensão do contexto de codões assim como facilitar a comparação com outros dados já existentes relacionados com a tradução genética.

No presente capítulo expõem-se os requisitos que foram surgindo ao longo do desenvolvimento da aplicação, assim como uma breve explicação da sua arquitectura, implementação e descrição funcional.

5.2 Requisitos funcionais

Os requisitos funcionais do sistema foram definidos em colaboração com os utilizadores. No entanto, devido ao carácter experimental deste projecto, muitos requisitos foram sendo redefinidos ao longo do desenvolvimento da aplicação. Podemos subdividir os requisitos funcionais em 3 categorias:

- Armazenamento
    - importar ficheiros contendo os genes dos organismos em dois formatos possíveis (FASTA e GenBank);
    - importar as tabelas que contêm o número de tRNAs disponíveis para cada organismo;
    - possibilidade de gravar e ler áreas de trabalho (*workspaces*), de modo a poder armazenar todos os procedimentos e resultados de vários estudos;
    - possibilidade de importação de tabelas de RSCUs, podendo ser atribuídas posteriormente aos organismos;
    - incluir as diversas tabelas de correspondência entre codões e aminoácidos;
    - leitura e gravação de matrizes de formato (genomas x 3904 pares de codões);
    - permitir gravar os valores residuais presentes nas tabelas em ficheiros;
    - oferecer a possibilidade de gravação dos dados em visualização em ficheiro.
- Processamento e parametrização
    - possibilidade de associar padrões de qualidade aos genes no momento da sua leitura, permitindo a criação de relatórios finais assim como a sua visualização em dois grupos distintos: i) genes válidos; ii) genes não válidos;
    - permitir o cálculo dos RSCUs para genes ou genomas completos;
    - criação de matrizes de quantificação sobre os genes válidos para posterior cálculo dos valores residuais;
    - possibilidade de comparação das matrizes através do módulo das diferenças dos valores residuais entre duas matrizes;
    - possibilitar a procura de genes homólogos entre diferentes organismos para posterior alinhamento;
    - possibilidade de comparar todos os organismos presentes na aplicação através da análise de *clustering*, construindo um árvore de distâncias;

- permitir transformar matrizes 61x64 em 1x3904, permitindo aplicar análise de *clustering* e de *biclustering* aos novos mapas;
- obter os histogramas dos codões presentes em determinado organismo;
- permitir a contagem do número de codões e de aminoácidos repetidos nas sequências;
- possibilidade de definir codões raros;
- permitir a procura diferentes padrões nos genes, assim como: i) codões raros; ii) padrões de valores residuais; iii) sequências de codões; iv) sequências de aminoácidos; v) percentagens de GC presente no gene; vi) valores de CAI. Os filtros criados terão de ser guardados para posterior consulta ou reutilização;
- possibilitar a construção de novas matrizes de valores residuais para os genes que passem nos filtros, podendo ser comparadas com qualquer outra matriz presente no sistema;
- permitir calcular os RSCU para o gene ou conjunto de genes, com possibilidade de gravação dos valores em ficheiro;
- possibilidade de gravar a informação presente no gene em ficheiro.

- Visualização
  - visualização das matrizes com os valores residuais com possibilidade de aplicação de análise de *clustering* e de *biclustering*;
  - visualização dos genes, com os codões representados com as cores que correspondem aos intervalos dos valores residuais. A visualização tem de conter diversa informação: i) sequência de aminoácidos; ii) assinalar codões repetidos; iii) assinalar nucleótidos repetidos; iv) assinalar metilações; v) incluir o tRNA que faz correspondência ao codão; vi) percentagem de valores residuais presentes nos genes agrupados ao código de cores; vii) número de codões raros e sua razão com o comprimento do gene; viii) *codon usage*.
  - redesenho de genes reflectindo os valores obtidos pela aplicação;

- possibilidade de redireccionamento dos vários valores obtidos num organismo, por exemplo, valores residuais, para outro organismo.

5.3 Requisitos não funcionais

Um dos requisitos principais prendia-se com a rapidez de processamento. Como os genomas mais pequenos contêm normalmente um número superior a 6000 genes e um dos objectivos era comparar vários genomas em simultâneo, teremos de optimizar ao máximo os algoritmos de forma a minimizar o tempo de processamento.

A aplicação não poderá ler o genoma para a memória, pois os genomas de maiores dimensões têm perto de uma centena de mega *bytes*. Para tal, a aplicação construirá um mapa em memória com a localização dos genes no ficheiro para posterior acesso.

Embora não existindo nenhum impedimento teórico para limitar a aplicação à abertura de genomas em simultâneo iremos impor o limite de 200.

A aplicação será construída para sistemas operativos Windows. A linguagem utilizada será o C++ devido ao desempenho que é necessário para processamento de grandes quantidades de dados.

5.4 Casos de uso

Em qualquer aplicação desenvolvida em investigação fundamental não podemos prever onde o próximo passo nos vai levar, ocorrendo muitas vezes a eliminação de opções tomadas ou a sua continuação e seu aprofundamento. Portanto, os primeiros requisitos centraram-se mais em definir uma arquitectura geral que suportasse a longo prazo a inserção de novos componentes ditados pela necessidade progressiva.

Para simplificar o planeamento da aplicação recorreu-se a um diagrama de casos de utilização, em notação *Unified Modeling Language* (UML) [143], que define os requisitos básicos da aplicação (Figura 31).

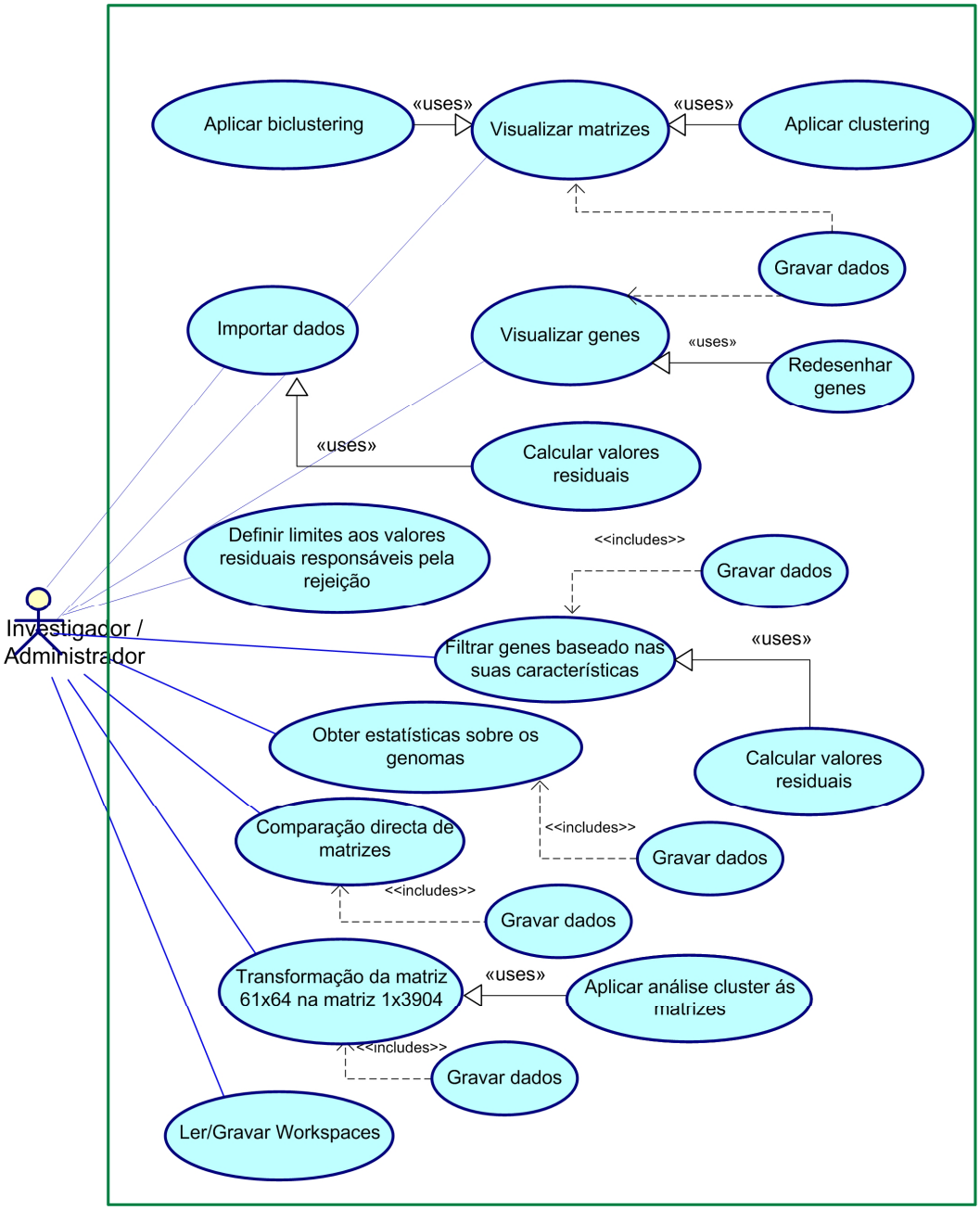

Figura 31 - Diagrama de casos de utilização para o sistema Anaconda.

O investigador é o actor único tendo poderes para configurar e usar todo o sistema. Esta simplificação foi possível por não ter sido identificado requisitos de segurança que conduzissem à criação de perfis diferenciados por utilizador.

O utilizador poderá importar as sequências existentes em ficheiros para a aplicação, para posterior análise, em dois formatos possíveis: FASTA e GenBank. A aplicação terá de estar preparada para possibilitar futuramente a leitura de outros formatos. O utilizador poderá aplicar filtros às sequências lidas, aumentando assim a qualidade dos genes que

serão processados. Terá de ter a possibilidade de definir qual a tabela que faz a correspondência entre codão/aminoácido, pois consoante a ordem a que pertence o organismo terá uma tabela específica, não muito distante da tabela que define as correspondências gerais. Na ausência dessa tabela o sistema assume a tabela geral. Estas tabelas são fornecidas pelo NCBI podendo ser descarregadas a qualquer momento da página do NCBI. No entanto, no momento da instalação o sistema já copia essas tabelas libertando o utilizador dessa tarefa.

Ao ler os genes para o sistema, terá de criar as tabelas de contingência e calcular os valores residuais ajustados, ficando assim disponíveis para análise e sua posterior visualização.

No diagrama de caso "visualizar matrizes" o utilizador poderá visualizar as matrizes obtidas aquando da leitura dos genomas. Poderá também aplicar análise estatística (*clustering* ou *biclustering*) à matriz em análise, evidenciando assim a formação de grupos entre pares de codões.

A definição dos limites para a rejeição dos valores residuais é da responsabilidade do utilizador. Por defeito, o módulo da rejeição dos resíduos ajustados é de 3. No entanto o utilizador poderá querer diminuir ou aumentar esse limite.

No diagrama de caso "visualizar genes" o utilizador poderá visualizar os genes podendo seleccionar quais características do gene que pretende observar, tais como os aminoácidos correspondentes, *codon usage*, o anti-codão, entre outras. Poderá também alterar o gene recorrendo a parâmetros fornecidos pelo sistema.

A inclusão do diagrama de caso "gravar dados" fornece ao utilizador a possibilidade de gravação em ficheiro dos resultados visíveis, estando englobado neste diagrama de caso os dados visuais assim como a possibilidade de gravar em ficheiro texto os resultados obtidos. Com esta última possibilidade proporciona-se a capacidade de poder efectuar posteriores análises através dos dados obtidos.

O diagrama de caso "Filtrar genes baseado nas suas características" contém a opção de isolar genes que possuem determinadas características, como um determinado número de codões raros, certa combinação de valores residuais ajustados, determinada sequência de aminoácidos ou codões, determinados níveis de GC, entre outros. Estes filtros poderão ser aplicados em simultâneo. Os genes seleccionados poderão ser processados novamente

obtendo assim novos valores residuais ajustados. Estas novas tabelas poderão ser comparadas com qualquer das tabelas existentes na aplicação.

O diagrama de caso "Obter estatísticas sobre os genomas" estabelece quais as estatísticas possíveis de obter para os genomas carregados na aplicação. Por exemplo, o número de codões e aminoácidos presentes no genoma, a distribuição de codões nos genes, vários histogramas sobre a percentagem de GC, a distribuição dos valores residuais, a distribuição do tRNA, a distribuição do CAI, entre outros.

A "comparação directa de matrizes" fica responsável por construir os mapas de diferença de contexto, entre mapas que contêm a mesma ordem de codões nas linhas e colunas. O resultado é um mapa de diferença de contexto que contém o módulo das diferenças dos valores residuais entre os dois mapas em análise. Esta operação poderá ser aplicada a qualquer tipo de mapa, desde que tenha a mesma dimensão e a mesma sequência de codões nas linhas e colunas.

A transformação das matrizes 61x64 em matrizes 1x3904 é necessária para efectuar comparação de valores residuais entre genomas. Também é possível aplicar análise de *clustering* e *biclustering* a estes novos mapas assim como obter mapas de diferença de contexto.

A possibilidade de gravar e ler áreas de trabalho foi um dos requisitos apresentados inicialmente, pois quando se trabalha com vários genomas em simultâneo torna-se mais simples gravar todo o trabalho num só ficheiro, com a possibilidade de efectuar a sua leitura posteriormente. Esta opção reduz em muito o tempo inicial de leitura, particularmente quando se trabalha com um grande número de organismos.

5.5 Arquitectura e desenvolvimento do sistema

Na definição da arquitectura do sistema teve-se o cuidado de separar as várias funcionalidades por blocos. Para tal, foram definidos quatro blocos principais onde estão enquadrados todas as ferramentas disponíveis (Figura 32).

Os blocos definidos são:

- Aquisição - responsável pela leitura dos ficheiros para o Anaconda e também pela imposição de qualidade nos genes lidos pela aplicação;

- Processamento - responsável pelo cálculo dos valores residuais, análise *clustering*, processamento de sequências homólogas e filtragem de genes;
- Visualização - responsável pela visualização de toda informação, assim como algumas caixas de diálogo que também têm essa função;
- Dados presentes em memória – área onde é guardada a informação processada, disponível para ser visualizada ou comparada com outros dados.

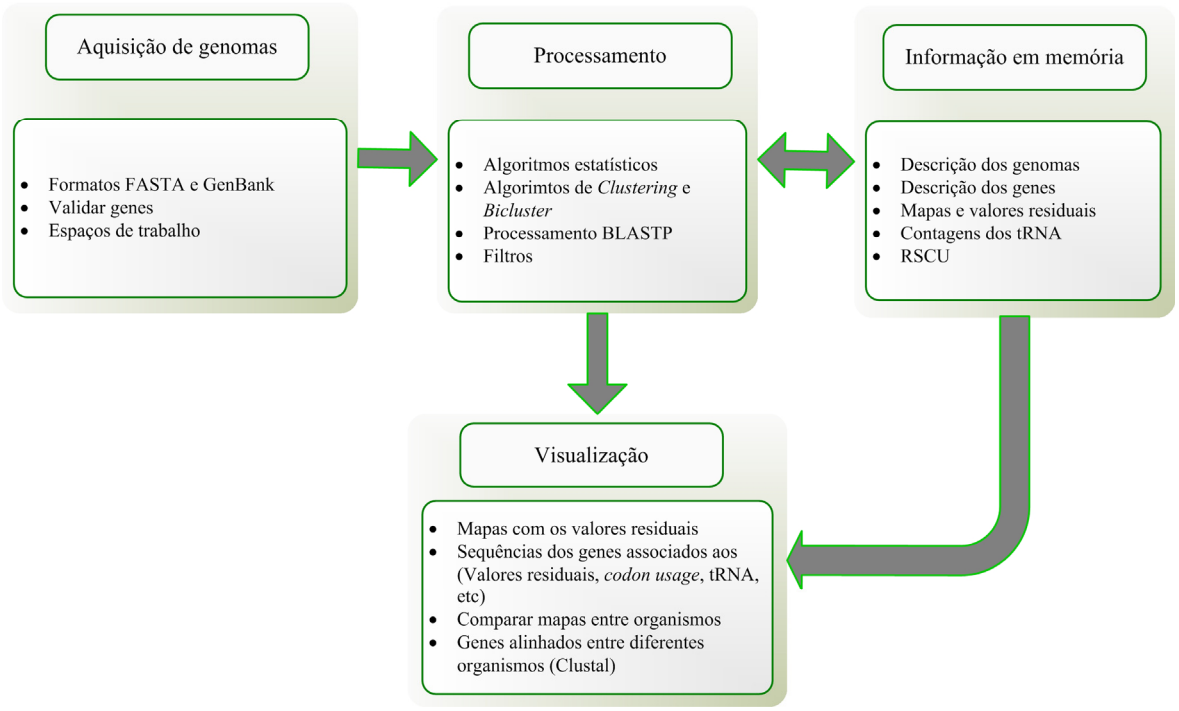

Figura 32 - Arquitectura do sistema.

Através desta arquitectura, composta por vários módulos separados, torna-se possível encaixar ou substituir diferentes módulos quando necessário. Esta particularidade simplifica alterações que foram sendo introduzidas ao longo do tempo ou que serão necessárias futuramente.

Para descrever a implementação do sistema, vamos recorrer a pequenos diagramas de forma a simplificar a descrição das várias classes desenvolvidas.

O sistema foi construído com o IDE *Visual Studio* 6.0, recorrendo ao C++ apoiado nas bibliotecas *Microsoft Foundation Class* (MFC). Estas bibliotecas impõem a arquitectura predefinida na Figura 33, que permite aos programadores, com uma certa experiência em MFC interpretar com facilidade qualquer aplicação desenvolvida em MFC. Fornece

também um conjunto de classes que permite uma diminuição significativa do tempo de desenvolvimento de aplicações. A linguagem de programação C++ foi escolhida devido a ter um óptimo desempenho a nível de processamento, requisito imposto à partida por esta aplicação.

A arquitectura MFC fornece automaticamente quatro classes base, na construção de uma aplicação do tipo *Single Document Interface* (SDI), impondo desde a sua criação um conjunto de regras bem definidas.

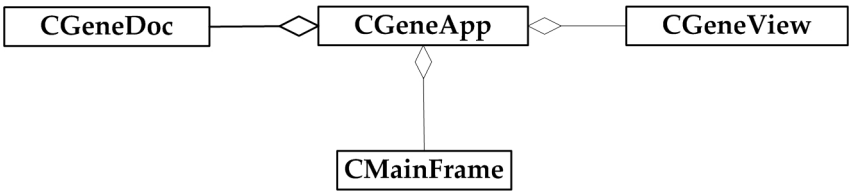

Figura 33 - Grupo de classes impostas pela arquitectura MFC.

As quatro classes base são:

- *CGeneApp* – classe a ser chamada no momento da execução da aplicação. Esta classe fica responsável por gerir todas as mensagens entre o sistema operativo e da aplicação;
- *CMainFrame* – gere a parte visual respeitante ao menu, barra de atalhos, barra de estado e as classes responsáveis pela visualização dos dados obtidos;
- *CGeneView* – neste caso específico, as funcionalidades desta classe foram incorporadas na classe *CMainFrame*;
- *CGeneDoc* – contém as estruturas de dados da aplicação, ficando responsável por responder aos comandos de activação das caixas de diálogo gerais.

A classe *CGeneApp*, sendo a classe principal, tem agregadas as três restantes classes impostas pela arquitectura MFC: i) *CGeneDoc*; ii) *CMainFrame*; iii) *CGeneView*. Como já foi descrito anteriormente, a classe *CGeneView* não tem utilidade neste caso específico devido à opção tomada para desenvolver a aplicação. Optou-se por construir várias classes, consoante os dados a visualizar, estando essas classes dependentes da classe *CMainFrame*. A classe *CGeneView* é normalmente utilizada quando se opte por uma só classe para a visualização dos dados no modelo *Single-document interface* (SDI), não sendo o caso na presente situação.

A classe *CGeneApp*, além das classes impostas pela arquitectura MFC, tem agregada a classe *CDadosIniciais* (Figura 34) que contém as propriedades de todos os parâmetros de configuração que um determinado utilizador possui quando está a trabalhar no sistema. Esses parâmetros estão gravados na sua área pública no sistema operativo, não indo interferir com os parâmetros de outro utilizador que partilhe o programa no mesmo sistema computacional. Esta concepção é implementada pela classe *CSaveIniFiles*, possibilitando a existência de parâmetros distintos para vários utilizadores no computador, herdando a classe *CIniFile*, responsável pela gravação e leitura desses parâmetros em ficheiro. Os parâmetros de configuração foram agregados à classe base para proporcionar um fácil acesso em qualquer parte do código.

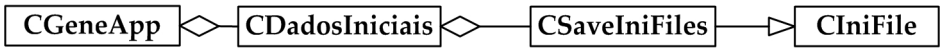

Figura 34 - Agregação de classes à classe *CGeneApp*.

5.5.1 Visualização de dados

O bloco responsável pela visualização dos dados no sistema está presente na Figura 35. No entanto, essas classes estão agregadas na classe *CMainFrame* devido à necessidade de fazer a troca de objectos consoante os dados e visualizar.

As classes estão divididas em dois grupos principais, *CGeneTab* e *CGeneViewPrincipal*. A *CGeneViewPrincipal* é responsável pela visualização da informação, contendo quatro classes agregadas que fornecem a visualização dos dados do sistema, sendo elas a *CViewMatrix*, *CViewGene*, *CViewMaps* e *CViewAlignment*.

A classe *CViewMatrix* constrói os mapas para visualização, ficando também responsável pela construção das árvores obtidas através da análise de *clustering*. Tem também agregada a classe *CDlgCluster* que fornece ao utilizador as diversas possibilidades de ordenar a matriz: i) ordenar alfabeticamente os codões pela primeira, segunda ou terceira base; ii) ordenar por aminoácidos; iii) *clustering*.

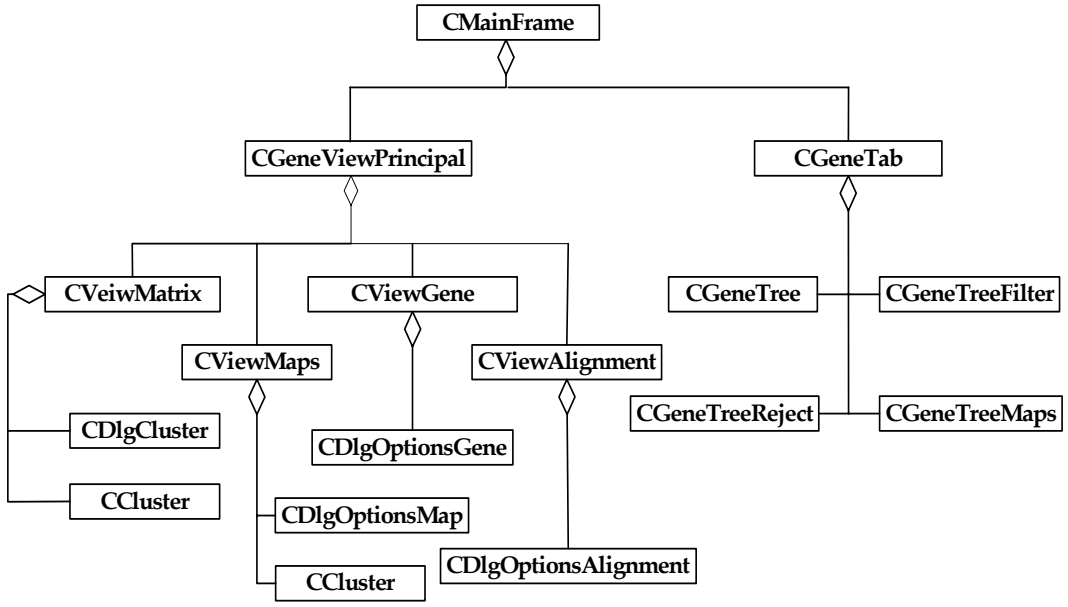

Figura 35 – Agregação das classes responsáveis pela visualização dos dados.

A classe *CViewGene* possibilita a visualização dos genes com várias opções disponíveis acedidas através da classe *CDlgOptionsGene*, desde a sequência de aminoácidos, anti-codões, entre outros valores possíveis de obter através do gene em visualização.

Por vezes, o objectivo de investigação é tentar detectar relações entre os mapas de pares de contexto de várias espécies. Para tal, opta-se por transformar as matrizes obtendo-se vectores de 3904 posições. Alinhando os vários vectores obtidos de diferentes organismos construímos um mapa comparativo de contextos (organismos) x (3094 pares de codões). Uma explicação com mais pormenor encontra-se no anexo 8.6. Estes mapas podem demorar algum tempo a construir, nomeadamente, se trabalhar com um número muito elevado de organismos, por exemplo superior a cem. O Anaconda fornece a possibilidade de gravar os mapas, podendo posteriormente serem lidos sem a necessidade de processar novamente todos os genomas. Estes mapas são visualizados através da classe *CViewMaps* que oferece várias opções de visualização através da classe *CDlgOptionsMap*, podendo também ser aplicada a análise de *clustering*.

A classe *CViewAlignment* é responsável pela representação visual das sequências dos genes homólogos alinhados. A classe *CDlgOptionsAlignment* permite várias opções de visualização os alinhamentos.

5.5.2 Estrutura de dados

A classe *CGeneTab* está responsável por agregar as quatro classes responsáveis pelas árvores que suportam toda a informação acessível no Anaconda. A classe *CGeneTree* contém os genes que passaram pelo filtro de qualidade, sendo assim, considerados válidos para análise. A *CGeneTreeReject* contém os genes que não passaram nos filtros de qualidade impostos aos genes no momento da leitura para o sistema. A classe *CGeneMaps* contém todos os mapas lidos pelo Anaconda e, finalmente, a classe *CGeneFilter* contém os genes filtrados por uma ferramenta específica, mediante várias características impostas pelo utilizador. No entanto, esta última classe também fica responsável por guardar todos os genes homólogos obtidos pela ferramenta BLAST, para posterior análise. Todos os genes referidos anteriormente estão agrupados por genoma e cromossoma.

A classe *CGeneDoc* agrega dois tipos distintos de classes. As classes que contêm a estrutura da dados e as classes que permitem alterar, processar ou visualizar dados através de caixas de diálogo.

Na Figura 36 está representada a estrutura de dados que serve de suporte ao sistema, agregada na classe *CGeneDoc*.

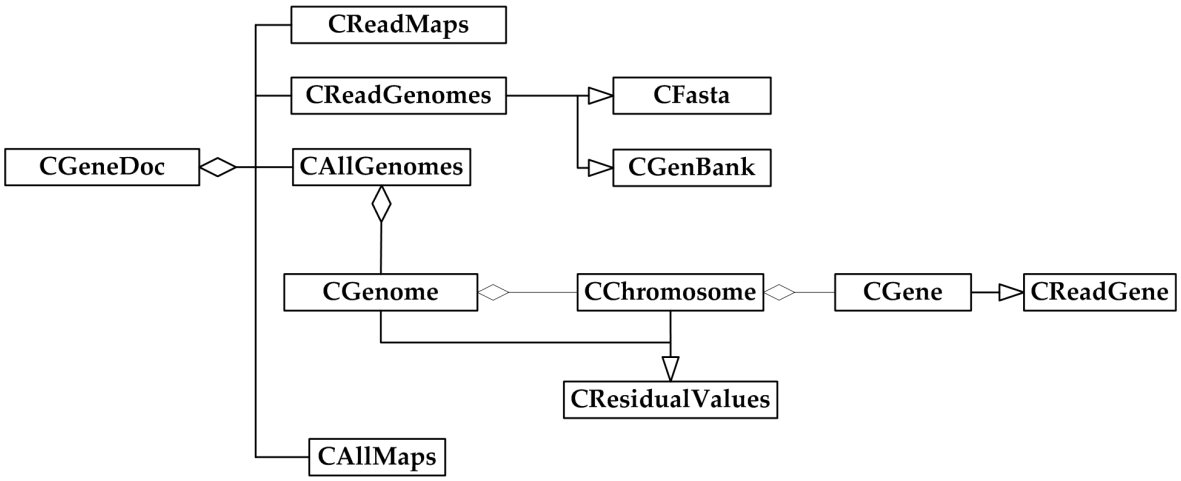

Figura 36 - Estrutura de dados descrita através de classes

As classes *CReadMaps* e *CReadGenomes*, como o próprio nome indica, são responsáveis pela leitura dos mapas e genomas. A classe *CReadGenomes* agrega as classes *CFasta* e *CGeneBank*, sendo os dois formatos de leitura possíveis no sistema. Para se respeitar os requisitos associados ao desempenho, tivemos que definir uma estrutura de dados que

conseguisse manter em memória as matrizes, os nomes do genomas, cromossomas, genes e sua posição no ficheiro, para que, numa eventual leitura dos mesmos, se possa aceder rapidamente ao início do gene sem ter que passar pelos anteriores. As sequências dos genes nunca são guardadas em memória, pois limitava muito a aplicação no momento em que se comparam, por exemplo, mais de 100 espécies em simultâneo. A classe *CAllGenomes* é responsável por gerir todos os genomas processados e agrega a classe *CGenomes* que contém o genoma em si. A classe *CChromosome* contém os cromossomas pertencentes ao genoma, que por sua vez agrega a classe *CGene* que não é mais do que conjunto de todos os genes pertencentes ao cromossoma. As duas últimas classes herdam a classe *CResidualValues* responsável por guardar e calcular os valores residuais nas tabelas de contingência obtidas no momento da leitura dos genomas. A classe *CGene* herda da classe *CReadGene* a possibilidade de ler a sequência do gene quando necessária, porque como referido anteriormente não se encontra guardada em memória. A classe *CReadGenomes* fica responsável por ir construindo a estrutura de dados em memória.

A classe *CAllMaps* contém todos os mapas lidos pelo sistema e disponíveis para processamento.

5.5.3 Configuração e processamento

Por último, apresentam-se as classes responsáveis pela configuração e processamento de dados. As classes utilizadas para este fim são as classes que recorrem às caixas de diálogo, normalmente com o prefixo "*CDlg*", estando representados no diagrama de classes da Figura 37. No entanto, nem todas as classes estão presentes neste diagrama, optando-se por representar somente as mais importantes.

A classe *CDlgColors* é responsável por configurar os limites dos valores residuais fazendo-os corresponder às cores a visualizar nas matrizes e genes. É também representado um histograma contendo a distribuição dos valores residuais de uma determinada matriz. Tem agregada a classe *CDlgReport* que possibilita gerar um relatório sobre a distribuição dos valores residuais.

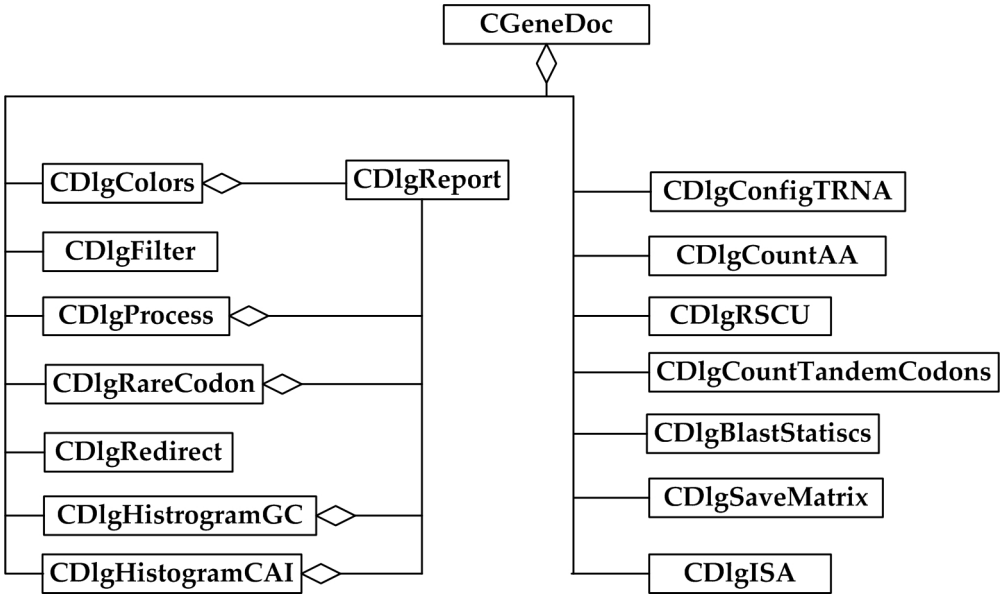

Figura 37 - Classes responsáveis pela configuração de parâmetros e processamento de informação.

A possibilidade de procurar genes que obedeçam a certos parâmetros é conseguida através da classe *CDlgFilter*. As definições dos parâmetros são guardadas para posteriormente se poderem aplicar.

A classe *CDlgProcess* é responsável pela parte visual da leitura dos genomas, recorrendo à classe *CReadGenomes* para efectuar o processamento. No final de cada leitura é gerado um relatório com algumas estatísticas sobre o genoma lido.

A classe *CDlgRedirect* permite redireccionar as matrizes de contexto de um organismo para outro organismo, possibilitando a análise de comportamento de um genoma com as tabelas de outro genoma. Este ponto é muito importante, quando o objectivo é estudar qual o comportamento de um determinado gene inserido noutro genoma.

As classes *CDlgHistogramGC* e *CDlgHistogramCAI* permitem a construção de histogramas contendo os níveis de GC nas três posições possíveis, a percentagem de GC total e a distribuição do CAI de um determinado genoma.

A classe *CDlgConfigTRNA* permite a configuração e definição dos tRNAs para um determinado genoma. A classe *CDlgCountAA* e *CDlgCountTandemCodons* permitem a contagem de aminoácidos e codões em tandem nos genomas. A classe *CDlgRSCU* permite a configuração e definição dos RSCUs, valores necessários ao cálculo do CAI.

Para encontrar genes homólogos entre os genomas presentes no Anaconda recorre-se à ferramenta BLAST, disponível pela classe *CDlgBlastStatistics*, podendo posteriormente obter variada informação sobre a relação entre contexto de codões nas sequências homólogas pertencentes a diferentes organismos.

A classe *CDlgSaveMatrix* permite a gravação em ficheiro das matrizes que contêm os valores residuais.

A classe *CDlgISA* permite a aplicação de um algoritmo de *biclustering* para a procura de grupos que partilhem características, tanto nas matrizes 61x64 codões como nos mapas.

5.6 Descrição de aplicação

A aplicação Anaconda, nome atribuído ao sistema desenvolvido, foi construída com base no paradigma do explorador de ficheiros, de forma a facilitar a interacção com o utilizador, apresentando uma árvore de navegação no lado esquerdo, onde residem os vários genomas admitidos, e do lado direito a apresentação dos resultados. Na Figura 38 é apresentada a janela principal do Anaconda, contendo vários genomas admitidos e com a visualização de uma matriz 61x64 codões com a análise de *clustering* aplicada.

Os ficheiros que correspondem aos cromossomas têm que obedecer ao formato FASTA ou GenBank, estando disponíveis nas bases de dados internacionais. Para agrupá-los num só grupo, os ficheiros terão que ser abertos em simultâneo. Aquando da sua abertura é dada a possibilidade de impor certos requisitos aos genes: i) começar com o codão de iniciação: ii) múltiplo de três nucleótidos; iii) não conter codões de finalização nas regiões codificantes; iv) conter um número mínimo de codões, entre outros. Com estas possibilidades conseguem-se impor às sequências admitidas os requisitos mínimos que definem a constituição de um gene.

No momento de leitura do ficheiros existe a possibilidade da não quantificação das sequências admitidas, permitindo posteriormente redireccionar as tabelas de valores residuais pertencentes a outros genomas. Esta técnica permite estudar como se comportam os contextos de determinadas sequências genómicas quando inseridas noutros genomas.

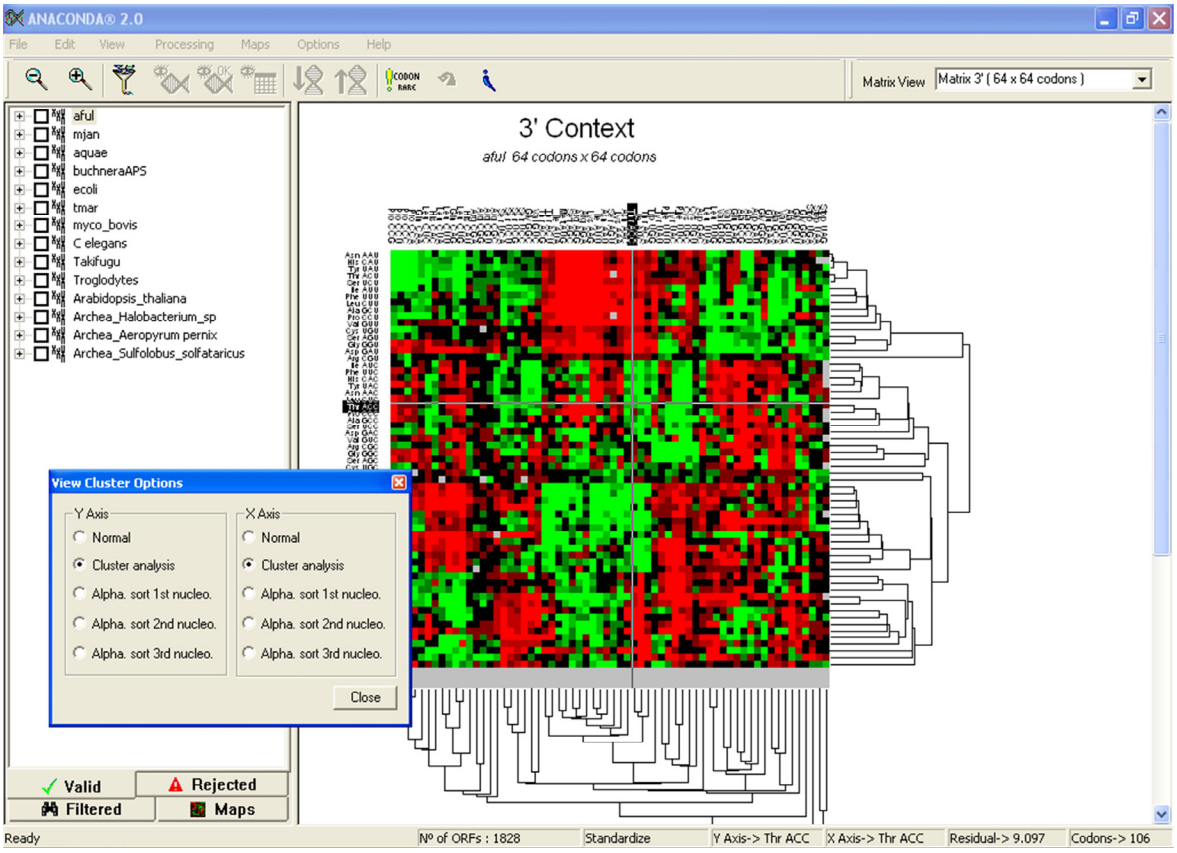

Figura 38 – Janela principal da aplicação Anaconda.

Na parte esquerda da janela principal (Figura 38) temos um *Tab Control*, no qual estão incluídos quatro árvores, as quais contém:

- Sequências Válidas: árvore onde ficam representados os genes, que preenchem os requisitos de entrada, ficando agrupados nos cromossomas a que pertencem;
- Sequências Rejeitadas: genes que não preenchem os requisitos de entrada impostos pelo utilizador;
- Sequências filtradas: genes que foram filtrados de acordo com pré-condições, que impõem certos requisitos aos genes. Esta ferramenta pode ser configurada pelo utilizador, baseada nas tabelas residuais;
- Mapas: os vários mapas de pares de contexto que poderão ser admitidos ao Anaconda.

Na Figura 39 A) pode-se ver a sequência de um gene reflectindo os valores residuais. Por exemplo, se a posição AAA com a posição CAT na matriz, corresponder o valor -30.05, significa que o codão AAA aparece com o fundo vermelho. Esta situação indica que o

codão AAA tem pouca afinidade pelo codão CAT. Se a posição CAT com a posição CCA corresponder o valor 10.0, o codão CAT aparece com o fundo em verde. Esta situação indica que esta sequência é preferida em relação às combinações com valores residuais negativos. Os codões raros são apresentados com uma elipse azul à sua volta.

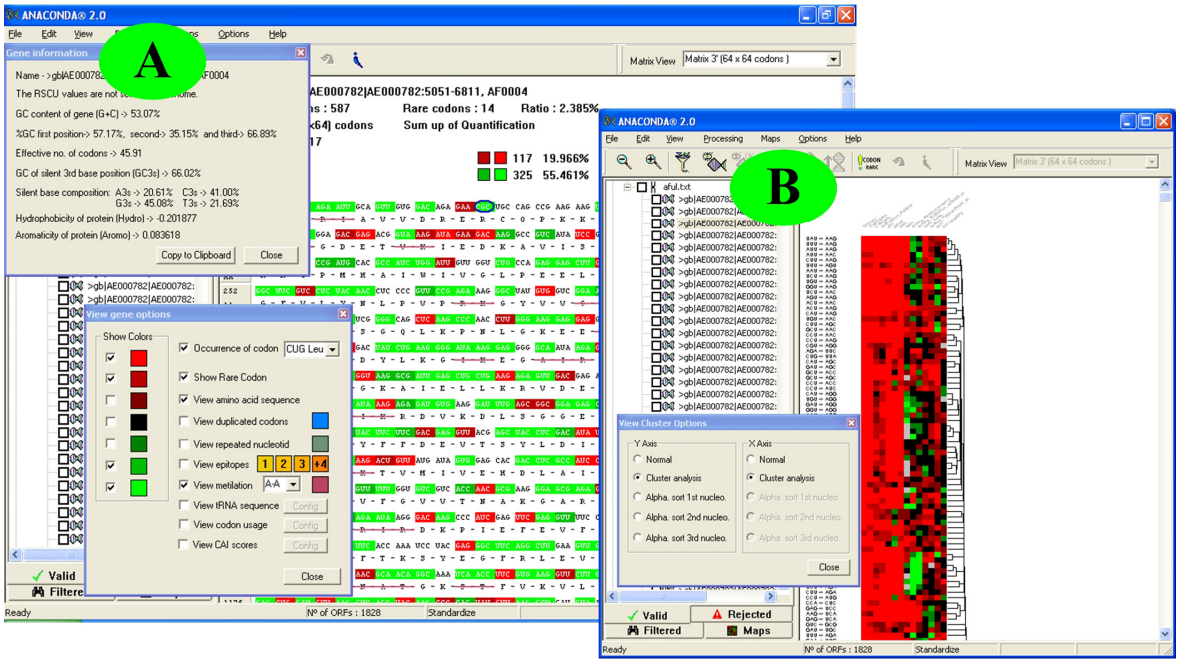

Figura 39 – Visualização de dados com a aplicação Anaconda. Figura A) - visualização de um gene colorido com os valores residuais correspondentes à sua matriz. Figura B) - visualização de um mapa de pares de contexto para várias espécies.

Temos também a possibilidade de ver diversa informação sobre o gene em análise: i) número de codões raros; ii) rácio relativo ao número total de codões; iii) percentagem de codões "vermelhos"; iv) a percentagem de codões "verdes"; v) número de ocorrências de um determinado codão; vi) zonas de metilação; vii) zonas onde ocorrem repetições de um mesmo codão. Estas possibilidades de visualização são activadas através de uma caixa de diálogo também presente na imagem. Na outra caixa de diálogo visível na imagem, pode-se obter informação, tal como, percentagem de codões com G e C na terceira posição, o *Codon Adaptation Index* (CAI), número efectivo de codões, hidrofobicidade, entre outras informações. Estes resultados podem ser armazenados em ficheiro de texto para posteriores análises.

Na Figura 39 B) pode-se observar ainda um mapa comparativo de pares de contexto no qual foi aplicado a análise *clustering* para evidenciar a existência de possíveis padrões.

A aplicação disponibiliza ainda várias ferramentas para analisar as sequências ao pormenor, não estando todas as possibilidades descritas na presente secção. Na (Figura 40) a caixa de diálogo A) contém o número de tRNAs disponíveis por codão, estando organizados por organismos para poderem ser atribuídos aos genomas em estudo. Assim, é possível estudar o comportamento dos tRNAs relativamente ao contexto de codões, pois a abundância ou a ausência de um determinado tRNA pode influenciar o processo de tradução. O Anaconda contém um conjunto significativo de tabelas de tRNAs, por organismo, aproximadamente cem. O conjunto de dados foi obtido aplicando o tRNAScan, descrito anteriormente na secção 3.7, às sequências de um conjunto vasto de organismos.

A caixa de diálogo B) apresenta uma ferramenta que permite encontrar diversos padrões entre as sequências admitidas. A ferramenta disponibiliza um conjunto vasto de filtros a aplicar às sequências, podendo ser usados em simultâneo. Entre os vários filtros disponíveis, estão: i) padrões de cores nos genes de um determinado genoma; ii) padrões de codões raros; iii) sequências de nucleótidos ou aminoácidos; iv) percentagens de codões raros; v) percentagem de valores de GC, entre outros. Existe também a possibilidade de gravar os filtros para sua posterior reutilização. Os genes que corresponderem ao filtro aplicado serão carregados na árvore dos genes filtrados para posterior visualização. Pode-se construir posteriormente uma matriz de valores residuais contendo somente os genes filtrados, possibilitando assim a sua comparação com outras tabelas existentes na aplicação.

Na caixa de diálogo C) O investigador tem à sua disposição uma ferramenta com a qual poderá obter informação variada: número de codões, número de aminoácidos, valores *Relative Synonymous Codon Usage* (RSCU), entre outras informações. Esta ferramenta poderá ser aplicada a genes, cromossomas e genomas.

Figura 40 – Várias caixas de diálogo acessíveis no Anaconda. A) contém o número de tRNAs disponíveis por codão, organizados por organismos, podendo-se aplicar aos organismos em estudo. B) ferramenta que permite encontrar diversos padrões entre as sequências admitidas. C) quantificação do número de codões e aminoácidos e cálculo do *codon usage*. D) histogramas que contêm a distribuição do GC nos genes. E) definição do nível de codão raro.

A caixa de diálogo D) contém os histogramas da distribuição do GC, para as várias posições, nos genes. Uma das grandes potencialidades do sistema é a possibilidade de gerar histogramas com os CAI de um determinado genoma ou cromossoma.

A caixa de diálogo E) mostra uma ferramenta que calcula histogramas, contendo a ocorrência dos codões num determinado genoma ou cromossoma. Permite também redefinir o nível que indica se um determinado codão é raro ou não.

Através da ferramenta de análise de *clustering* pode-se construir árvores filogenéticas de todos os genomas carregados na aplicação.

5.6.1 Análise de alinhamentos

Para estudar a associação do contexto de codões nas sequências homólogas entre organismos, o algoritmo BLASTP foi introduzido no Anaconda. A opção tomada foi interagir com o algoritmo BLASTP que corre localmente em detrimento da versão *web*, ficando assim com mais controlo do processamento da informação. Na Figura 41 estão descritos os passos que são necessários para o executar de modo a obter as homologias num determinado número de sequências.

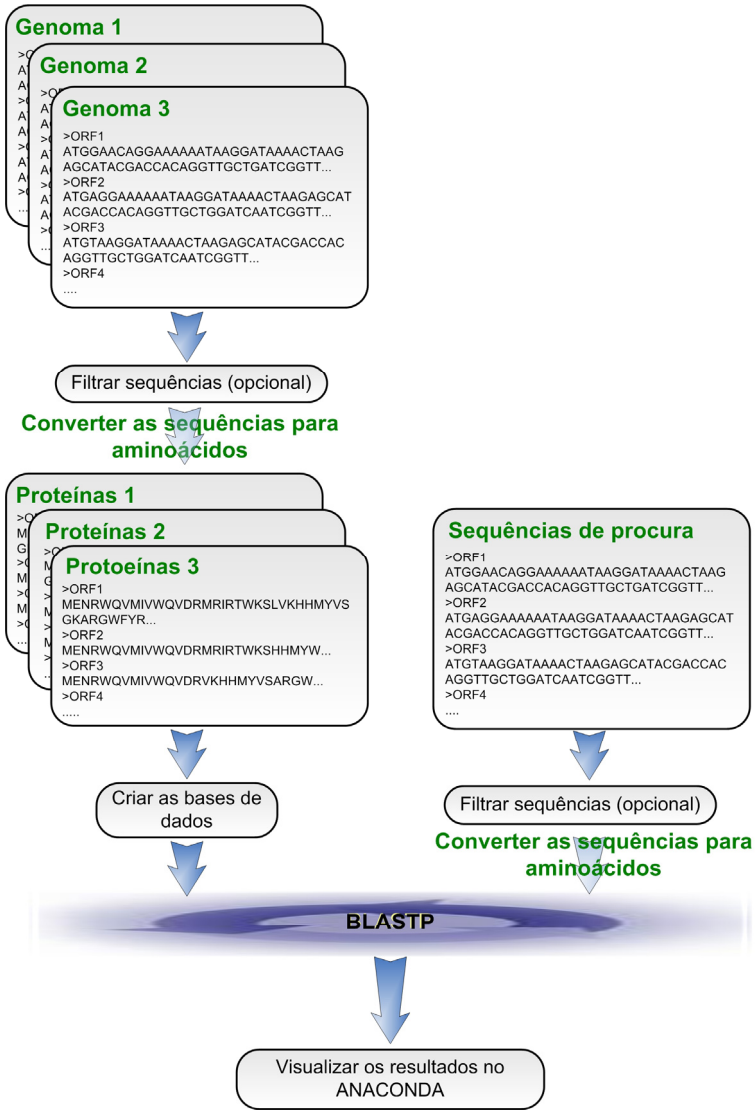

Figura 41 – Os vários passos que o Anaconda perfaz para correr o BLASTP.

O BLASTP está dividido em dois módulos principais: i) criação de uma base de dados local; ii) o processo de procura de sequências homólogas na base de dados criada anteriormente.

O processo da criação da base de dados engloba três passos (Figura 41):

- escolha dos organismos que vão integrar a base de dados, tornando-se assim nas sequências alvo. Neste passo é também possível aplicar filtros aos genomas, podendo o utilizador estar interessado em analisar sequências que obedeçam a um determinado padrão. Os filtros disponíveis são os mesmos apresentados anteriormente;
- transformação das sequências de codões em aminoácidos, pois o objectivo é a procura de sequências homólogas;
- criação da base de dados com as sequências que resultaram dos passos anteriores.

Na Figura 42 A) é apresentada a caixa de diálogo que permite despoletar o BLASTP no Anaconda. O utilizador terá de escolher quais os organismos que vão constituir a base de dados e qual o organismo de procura, definindo ainda outros parâmetros possíveis de aplicar ao algoritmo BLASTP.

Estes passos são todos executados de forma automática sem necessidade de intervenção. O Anaconda guarda o registo das sequências já processadas, no momento da criação da base de dados, para futuramente serem confrontados com novos processos de criação de uma nova base de dados. Se determinada base de dados já existir o processo passa automaticamente para o BLASTP, poupando assim tempo de processamento.

A procura das sequências homólogas é então efectuada pelo BLASTP, na base de dados anteriormente criada, e os resultados são lidos pelo Anaconda para posterior análise.

Com esta aproximação vários processos de conversão são executados, codões para aminoácidos, de forma transparente para o utilizador e sem impacto no algoritmo. Outra vantagem é a não necessidade de criação de enormes base de dados locais, construindo-se apenas o que o utilizador necessita.

Os resultados obtidos com o BLASTP são introduzidos na árvore dos genes filtrados para posterior análise. O utilizador poderá navegar pela árvore e visualizar quais os genes homólogos existentes nos outros organismos para o gene em questão, obtendo o seu alinhamento múltiplo através da ferramenta ClustalW inserida no Anaconda. Obtém também, um conjunto vasto de informação sobre o alinhamento que determinada sequência

obteve. É igualmente apresentado um histograma com os *scores* do alinhamento obtidos pelo ClustalW.

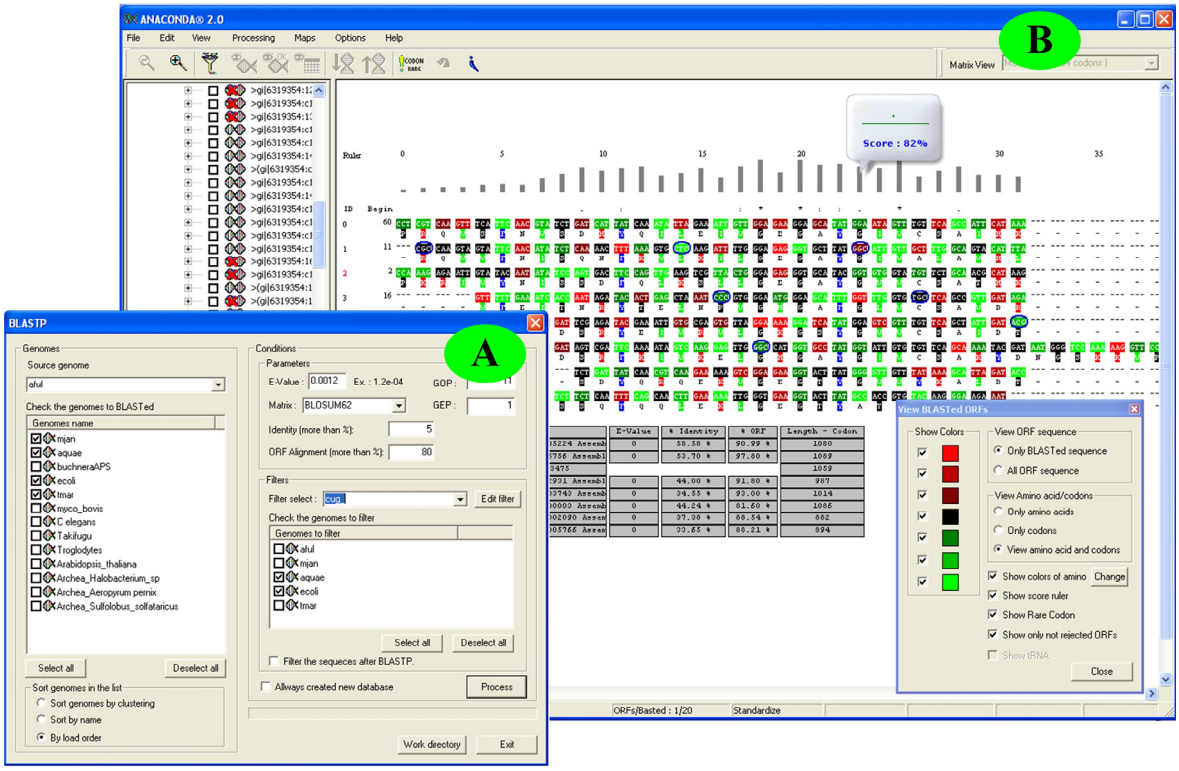

Figura 42 – Processo BLASTP no Anaconda. A) caixa de diálogo onde se define quais os organismos envolvidos no BLASTP entre outros parâmetros. B) Resultados obtidos após executar o BLASTP e proceder ao alinhamento através da ferramenta ClustalW.

5.6.2 Optimização de sequências de genes

Um dos principais objectivos do Anaconda consistia na modificação da sequência dos genes in silico, sem alterar a sequência de aminoácidos, para reduzir as áreas mais vulneráveis à ocorrência de erros de tradução. Basicamente, o objectivo é alterar os codões, preservando o seu aminoácido, de forma a reflectir os parâmetros obtidos com o Anaconda.

Com esse objectivo foi desenvolvida uma ferramenta que permite a alteração dos genes baseado em parâmetros ajustados pelo utilizador (Figura 43). A ferramenta é disponibilizada através da caixa de diálogo que se encontra na parte esquerda da imagem.

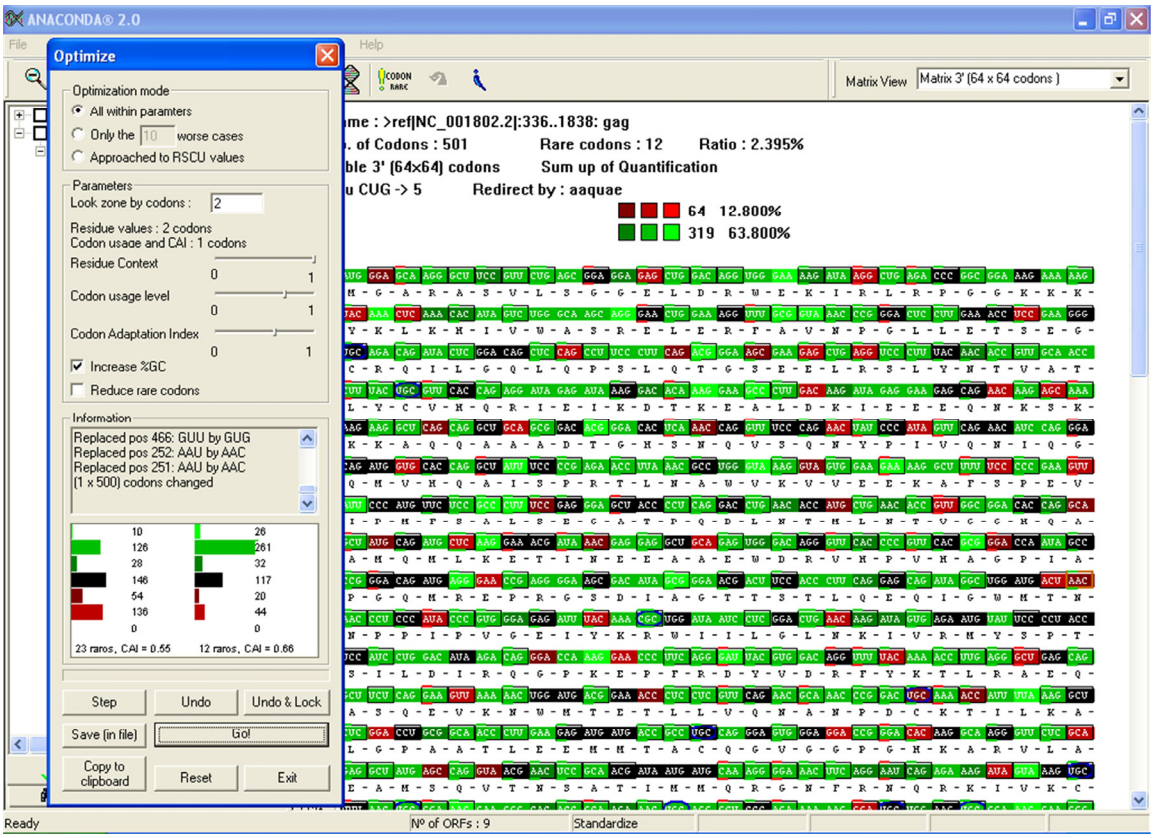

Figura 43 – Ferramenta que possibilita o redesenho de genes reduzindo as zonas mais vulneráveis ao erro no processo de tradução.

A ferramenta possibilidade a alteração do gene em três modos:

- Optimização total – o algoritmo de optimização é aplicado até se encontrar a melhor solução;
- Optimizar os piores casos – o algoritmo faz uma ordenação dos piores casos e aplica a melhor solução ao número de casos definidos. Os piores casos são obtidos pelo *score* calculado para cada posição, como referido na secção 4.5.
- Aproximar aos valores RSCU – o algoritmo percorre todos os codões e substitui pelos que apresentam melhor valor de RSCU para o aminoácido em análise. Nesta aproximação o utilizador não pode definir qualquer parâmetro de optimização.

Na imagem encontra-se um gene já optimizado, com as caixas formadas à volta dos codões a indicar que esse codão foi alterado para melhorar o contexto nessa zona. Neste caso concreto, o utilizador atribuiu um peso mais elevado ao contexto, como se pode ver pelas barras deslizantes presentes na caixa de diálogo. Pretendeu também aumentar a

percentagem de GC no gene. Os histogramas visualizados na caixa de diálogo mostram o número de valores residuais presentes antes e depois da aplicação do algoritmo. O histograma do lado esquerdo contém os valores antes das alterações e o do lado direito contém os valores depois das alterações. Como podemos ver, ouve um incremento dos valores residuais positivos, nomeadamente as três primeiras barras dos histogramas. Em contrapartida, ouve um decréscimo dos valores residuais negativos, as três últimas barras dos histogramas.

A ferramenta também possibilita ao utilizador optimizar o gene passo a passo. Esta opção está disponível nos três modos de alteração, podendo prosseguir passo a passo, recuar ou simplesmente bloquear determinado codão numa posição para não permitir a sua alteração.

5.6.3 Análise de biclustering

Devido às limitações impostas pela análise de *clustering* optou-se por introduzir no Anaconda um algoritmo de *biclustering*. O algoritmo implementado foi o ISA acessível através da caixa de diálogo presente na Figura 44.

A versão implementada contém algumas modificações, já descritas na secção 4.3.2. No entanto, mediante a escolha correcta de parâmetros é possível aplicar a versão original do ISA. Os resultados são mostrados no lado esquerdo da caixa de diálogo, permitindo a sua rápida visualização, sendo possível aplicar a análise de clustering ao grupo obtido.

Esta implementação tem a vantagem de ter um tempo de processamento muito inferior quando comparado com a implementação na aplicação BicAT [129]. Quando se aplica em tabelas de dados de dimensões elevadas, por exemplo (100 x 4000), essa diferença é muito significativa.

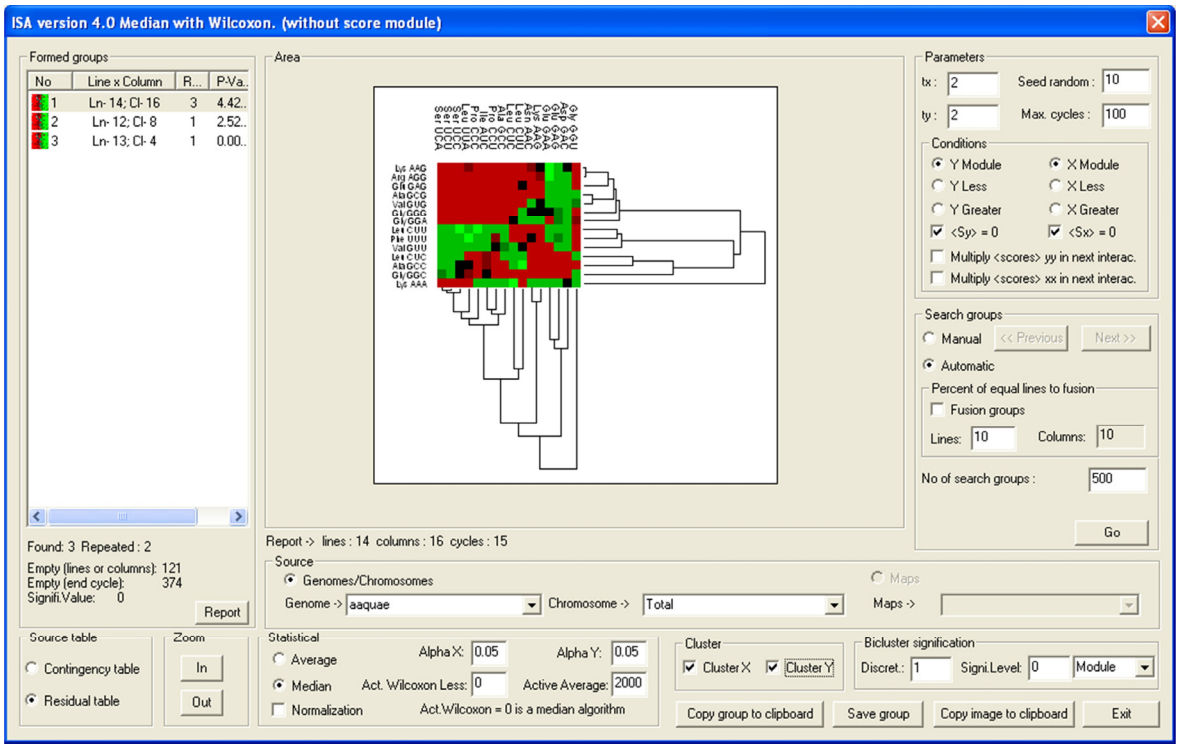

Figura 44 – Caixa de diálogo que permite aplicar o algoritmo de biclustering ISA às matrizes com os valores residuais.

5.7 Conclusões

Neste capítulo foi descrito o processo de desenvolvimento do Anaconda, aplicação desenvolvida para estudar o contexto de codões nos mais variados organismos.

Os principais requisitos, a rapidez de processamento e a facilidade de utilização, foram preocupações constantes no desenvolvimento da aplicação. Privilegiou-se a interface do utilizador procurando construir-se um ambiente rico e de fácil utilização. Como resultado, a aplicação contém 180 classes, 81 caixas de diálogo perfazendo um total de aproximadamente 75.000 linhas de código. A aplicação encontra-se disponível para *download* desde a versão 1.0, em 2006 em http://bioinformatics.ua.pt/applications/anaconda, tendo sido efectuadas uma média de 70 *downloads* por ano.

Apesar do tempo de desenvolvimento já investido no Anaconda a aplicação está aberta à integração de ferramentas adicionais.

No capítulo seguinte, apresentamos alguns resultados obtidos com o Anaconda no estudo da estrutura primária nas zonas codificantes.

Capítulo 6

---

6 Análise de genomas baseado no modelo desenvolvido

6.1 Introdução

Para obter mais conhecimento sobre a estrutura primária do genoma, possibilitando assim identificar as características estruturais que influenciam o erro no processo de tradução, utilizámos o software Anaconda, aplicando-o em vários cenários e efectuando múltiplas abordagens que deram origem a cinco publicações [144-148].

Neste capítulo iremos apresentar um estudo nas zonas codificantes de três leveduras eucariotas (*Saccharomyces cerevisiae*, *Candida albicans, Schizosaccharomyces pombe*) e de uma bactéria, *Escherichia coli*. Seguidamente, iremos efectuar um estudo comparativo entre as espécies que estão presentes nos três reinos da vida, eucariota, bactérias e arqueas, identificando pontos em comum aos três reinos. Por fim, iremos apresentar alguns dados obtidos com o algoritmo ISA-mediana.

A aplicação desenvolvida faculta variados meios necessários para efectuar os estudos que irão ser apresentados, possibilitando assim a identificação de padrões entre espécies. Fornece, também, novas perspectivas sobre o papel do contexto de codões na descodificação dos genomas e, em última instância, a pressão imposta pela maquinaria ribossomal sobre a evolução dos mesmos.

6.2 Análise comparativa do contexto entre pares de codões

A aplicação Anaconda, desenvolvida neste trabalho, permite a leitura das regiões codificantes de um genoma. Ao fazê-lo, identifica o codão iniciação em todas as sequências, movendo a janela de descodificação progressivamente de três em três nucleótidos até encontrar o codão de terminação. Este processo é realizado para todas as sequências codificantes que compõe o genoma em estudo. No processo de deslocamento da janela, vai contabilizando todos os pares de codões criando uma tabela de contingência de 61 por 64 codões. Esta tabela irá conter todos os pares de codões, do genoma que foi previamente lido, e servirá de base à aplicação dos cálculos estatísticos para obter os valores residuais ajustados.

6.2.1 Análise global do contexto de codões em *S.cerevisiae*

Com o objectivo de ter uma visão geral do mapa dos valores residuais ajustados a metodologia da análise residual foi aplicada à levedura *S.cerevisiae*, sendo o genoma obtido em repositórios especializados. Neste caso específico, o genoma foi obtido no repositório do NCBI (ftp://ftp.ncbi.nih.gov/genomes), contendo 6267 genes.

O genoma está em formato FASTA e é lido pelo Anaconda, gene a gene, contabilizando todos pares de codões numa matriz de contingência. É possível impor parâmetros de qualidade aos genes no momento do seu processamento. É também possível excluir certas posições do gene, como o seu início ou a parte final. Posteriormente, os valores residuais ajustados são obtidos através da tabela de contingência obtida sendo convertidos para um código de cores (Figura 45), com a tonalidade verde a corresponder aos valores superiores a 3, representando os pares de codões preferidos, e a tonalidade vermelha a corresponder aos valores que são inferiores a -3 que representam os pares de codões preteridos.

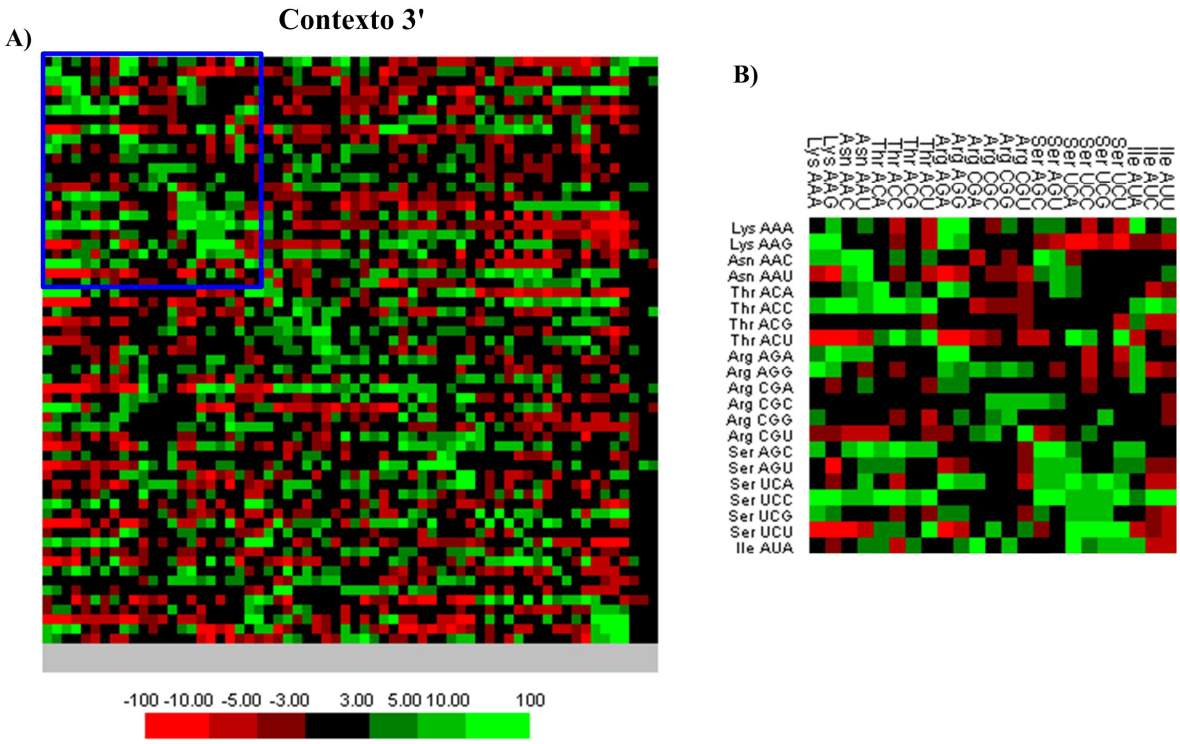

Figura 45 – Mapas representando os valores residuais, convertidos através de uma palete de cores. As diferentes intensidades indicam a dimensão dos valores residuais.

Na Figura 45 pode-se ver claramente que existe um conjunto de codões preferidos, representados em verde, e um conjunto de codões que são preteridos, representados a vermelho. No entanto, existem um grande número de casos onde os pares de codões que não são preferidos nem rejeitados, estando representados com a cor preta. Estes pares de codões estão compreendidos entre os limites [-3, 3] nos valores residuais ajustados. Portanto, os valores residuais ajustados dão-nos a informação sobre os pares de codões que estão acima ou abaixo do valor esperado. Os valores residuais positivos representam os pares de codões que estão em número superior ao esperado e os valores residuais negativos representam os pares de codões que estão em número inferior ao valor esperado. Os valores residuais que estão compreendidos entre os limites [-3, 3] estão em linha com o esperado.

A Tabela 4 mostra vários valores residuais que um determinado mapa poderá conter. Neste caso específico, a tabela contém os valores residuais dos pares de codões CUG-yyy, representando o sentido 3' pois o codão está ao lado direito do codão CUG.

Tabela 4 – Vários valores residuais correspondentes a vários pares de codões CUG-yyy

Codão 3'	Residual	Codão 3'	Residual	Codão 3'	Residual	Codão 3'	Residual
AAA	7,436	ACG	0,644	UCU	-10,007	CCA	-2,438
AAG	1,927	CGU	-1,809	CUU	1,167	CCG	2,895
AAU	0,397	CGC	2,981	CUC	2,18	CAU	2,026
AAC	2,037	CGA	8,258	CUA	5,258	CAC	2,642
ACU	-6,947	CGG	5,404	CUG	6,774	CAA	4,049
ACC	-5,239	ACG	-4,726	CCU	-1,769	CAG	7,105
ACA	-5,12	AGG	-0,666	CCC	8,894	UAA	0,22

6.2.2 Aplicar análise de *clustering* à matriz de *S.cerevisiae*

Os mapas que contêm informação sobre os pares de codões são de difícil interpretação, pois a informação parece estar distribuída de forma aleatória. Para tentar encontrar relações entre pares de codões aplicou-se a análise de *clustering* (Figura 46). Embora exista a possibilidade de ordenar as linhas e colunas de forma a agrupar os codões por aminoácidos, ou por ordem alfabética, não se obtiveram padrões relevantes ou interessantes em vários mapas em diversos genomas analisados. Estas opções estão sempre disponíveis no momento da visualização dos mapas, sendo possível identificar regras de associação entre pares de codões.

Com a análise de *clustering* foi possível evidenciar grupos com os valores residuais próximos. Aplicando simultaneamente o *clustering* às linhas e colunas vários grupos foram identificados com manchas de verde e vermelho, realçando as preferências dos codões (Figura 46).

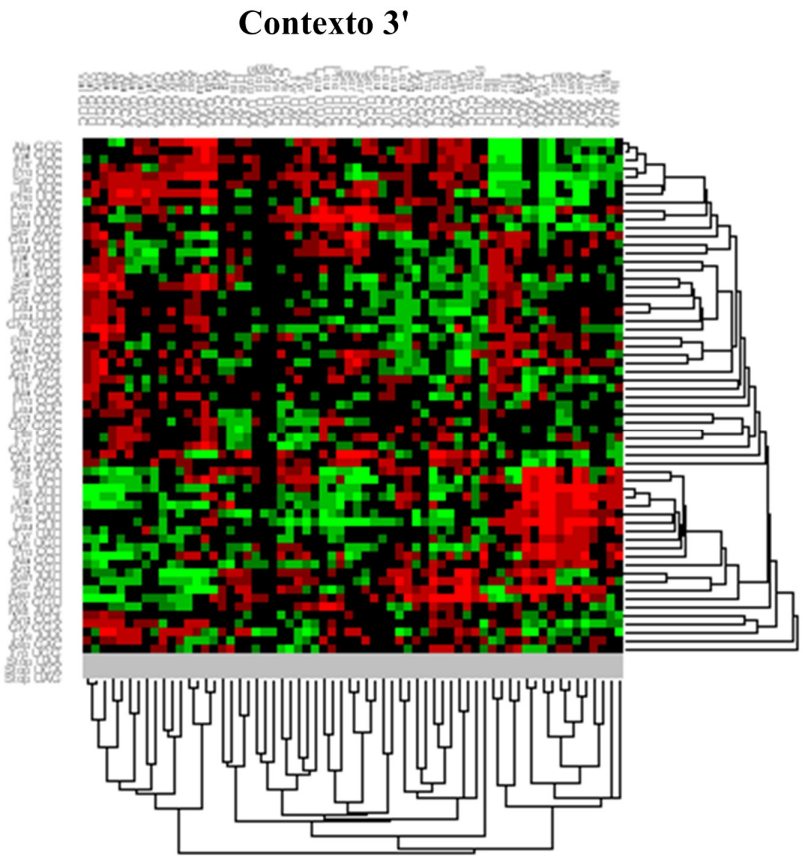

Figura 46 – Análise de *clustering* aplicada em ambas as dimensões num mapa de valores residuais, fazendo evidenciar as relações mais próximas entre pares de codões.

Para identificar os codões envolvidos na formação de grupos, sendo os responsáveis por criar regras entre pares de codões, foram seleccionados três grupos para uma análise mais profunda (Figura 47).

A Figura 47 a) é formada por pares de codões em que o último nucleótido do primeiro codão é o uracilo (U) e o primeiro nucleótido do codão seguinte é a adenina (A). Não se pode expandir a regra para as outras posições dos nucleótidos pois não existe a formação de padrões. Portanto, este grupo permite a criação da regra xxU-Ayy definindo a rejeição deste conjunto de pares, pois a cor predominante é o vermelho. A intensidade de rejeição não é idêntica para todas as combinações, no entanto, com a excepção do codão AAU, AGU e outros codões onde o valor residual está compreendido entre o limite [-3, 3], os codões terminados em U rejeitam os codões seguintes com o nucleótido A na primeira posição. Por outras palavras, e assumindo que o contexto influencia a correcta descodificação dos genes, os pares de codões com o forma xxU-Ayy da espécie *S.cerevisiae* têm mais dificuldade de processamento por parte do ribossoma.

Ao analisar os dois outros grupos verificamos a presença de dois padrões com predominância do verde indicando preferência nestes pares de codões. Assim sendo, duas regras adicionais são definidas para o organismo *S.cerevisiae*, nomeadamente a regra xxC-Ayy e a regra xxU-Gyy. Igualmente ocorrem excepções, como no caso anterior, aparecendo alguns valores residuais negativos ou com o valor entre o limite [-3, 3]. Contudo, existe uma predominância de bom contexto suficiente para a construção das referidas regras.

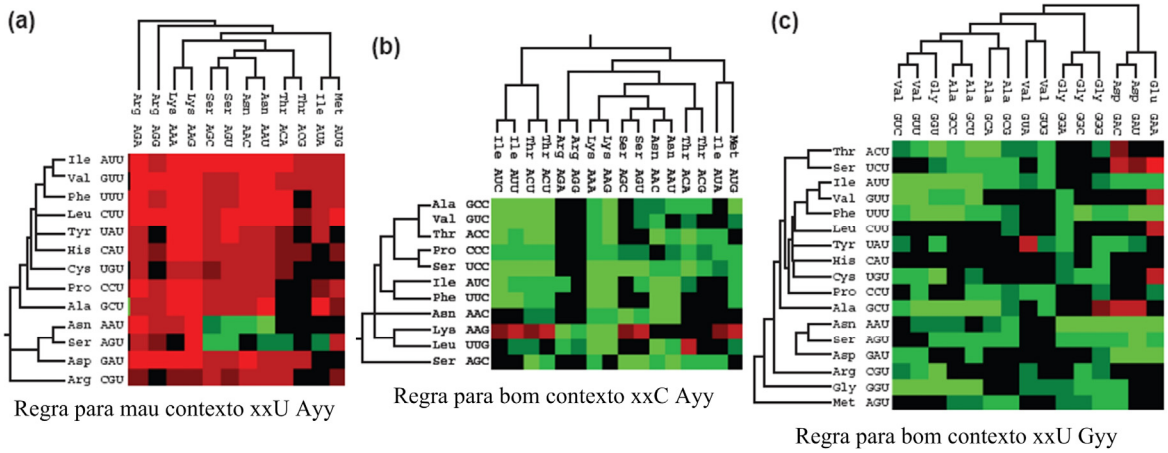

Figura 47 – Identificação de vários grupos possibilitando a definição de regras para a *S.cerevisiae*.

6.2.3 Comparação do contexto entre organismos

Quatro mapas de contexto estão presentes na Figura 48 que representam os valores residuais dos quatro organismos distintos. Fazendo uma análise mais detalhada destes mapas, pode-se observar que existe uma diagonal verde que sobressai nas três leveduras, *S.pombe*, *S.cerevisae* e *C.albicans* mas não na *E.coli* que pertence ao reino das bactérias. Este facto indica que geralmente os codões têm preferência pelo próprio, não querendo dizer que não exista preferência por outros codões. Isto fica-se a dever a zonas comummente repetidas, com o mesmo codão, nas sequências de organismos eucariotas. Repetições de três nucleótidos são comuns nos organismos eucariotas e deve-se ao processo de replicação do genoma [149].

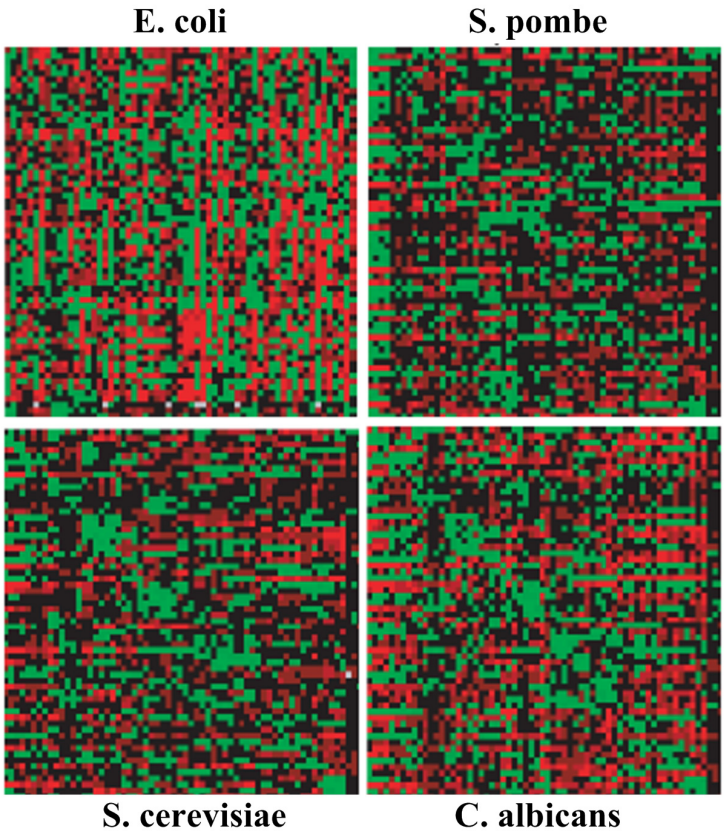

Figura 48 – Os mapas correspondem aos quatro organismos em análise. Como se pode ver a diagonal verde é evidente nos organismos eucariotas, *S.pombe*, *S.cerevisiae* e *C.albicans*.

No entanto, existem importantes diferenças que ficam evidentes quando os valores residuais das três leveduras são ordenados por ordem decrescente (Tabela 5). Estes valores representam os mais negativos e mais positivos e consequentemente proporciona uma boa forma para relacionar o contexto presente nas três leveduras. Nos dez mais positivos valores residuais, só três são comuns às três espécies, nomeadamente GAA-GAA, GGU-GGU e GCU-GCU. Um resultado similar é obtido quando os valores residuais mais negativos são ordenados, sendo comuns às três espécies dois pares de codões, o UUU-AAG e AUU-AAG.

Várias diferenças ficam evidentes, quando os valores residuais das leveduras em estudo são ordenados por ordem decrescente (Tabela 5). Estes valores representam os mais negativos e mais positivos valores residuais e consequentemente proporciona uma forma de relacionar o contexto presente nas três leveduras. Nos dez mais positivos valores residuais, só três são comuns às três espécies, nomeadamente GAA-GAA, GGU-GGU e GCU-GCU. Um resultado similar é obtido quando os valores residuais mais negativos são

ordenados, sendo comuns às três espécies dois pares de codões, o UUU-AAG e AUU-AAG.

Tabela 5 – Ordenados por ordem decrescente os dez mais negativos e positivos valores residuais para os organismos *S.cerevisiae*, *S.pombe* e *C.albicans*.

S.cerevisiae		**S.pombe**		**C.albicans**	
Contexto	**Residual**	**Contexto**	**Residual**	**Contexto**	**Residual**
Os valores mais negativos					
UUU->AAG	**-24,58**	GAA->CCU	-24,159	UUU->CCA	-32,691
GAU->AAG	-22,487	GAU->AAG	-24,124	UUC->GAA	-31,586
AUU->AAA	-21,546	**UUU->AAG**	**-23,899**	UCA->GAU	-28,317
AUU->AAG	**-21,285**	AUU->AAA	-22,923	**AUU->AAG**	**-28,284**
CUU->AAA	-20,656	UCU->AAG	-22,334	GGU->UUU	-27,198
UUU->AAA	-20,563	CUU->AAA	-21,25	AAC->UUA	-26,198
UCC->GAA	-20,069	GUU->AAA	-21,218	GAC->UUA	-25,795
AAG->UCU	-19,706	**AUU->AAG**	**-21,08**	**UUU->AAG**	**-25,316**
GAU->CAA	-19,274	UUU->AAA	-20,704	GGA->AAA	-25,26
GAA->CCA	-19,155	GAA->UCU	-20,698	UUC->GAU	-24,822
Os valores mais positivos					
GAU->GAU	29,839	CAG->CAA	25,279	ACA->ACA	49,476
AAG->AAG	29,937	GAA->GAG	25,644	CAC->CAC	49,511
UUG->AAA	30,459	AAG->AAG	26,901	CCA->CCA	52,889
GAA->GAA	**30,573**	CUU->CGU	27,013	**GAA->GAA**	**57,356**
AAG->AAA	31,427	**GAA->GAA**	**28,051**	AAG->AAA	58,605
CAG->CAA	33,445	AGA->AGA	29,623	**GCU->GCU**	**62,611**
AGA->AGA	33,798	AAA->AAG	30,358	ACC->ACC	70,117
GGU->GGU	**35,979**	**GCU->GCU**	**32,158**	**GGU->GGU**	**72,48**
GCU->GCU	**36,231**	**GGU->GGU**	**33,681**	AAC->AAC	87,115
CAG->CAG	45,422	UCU->UCU	35,086	CAA->CAA	105,216

A comparação de genomas era um dos objectivos pertinentes para a análise da estrutura primária dos genes. Com esse objectivo traçado, o Anaconda disponibiliza uma ferramenta que permite obter um mapa que resulta do módulo da diferença entre dois mapas em análise. Assim, é possível visualizar onde residem as diferenças mais significantes entre mapas e apontar diferenças entre genomas distintos.

Esta abordagem foi seguida para identificar diferenças entre os valores residuais dos pares de codões nas espécies *S.cerevisiae*, *S.pombe* e *C.albicans*. O resultado é um mapa de diferença de contexto (MDC) que contém o módulo das diferenças dos valores residuais

entre dois mapas (Figura 49). Uma nova coloração em tons de azul, com novos limites, foi definida para assinalar as diferenças numéricas.

Usando esta metodologia as diferenças entre mapas ficaram evidentes, nas três leveduras em análise, indicando que o contexto dos codões é especifico de cada espécie, tal como acontece com o *codon usage* (Figura 49).

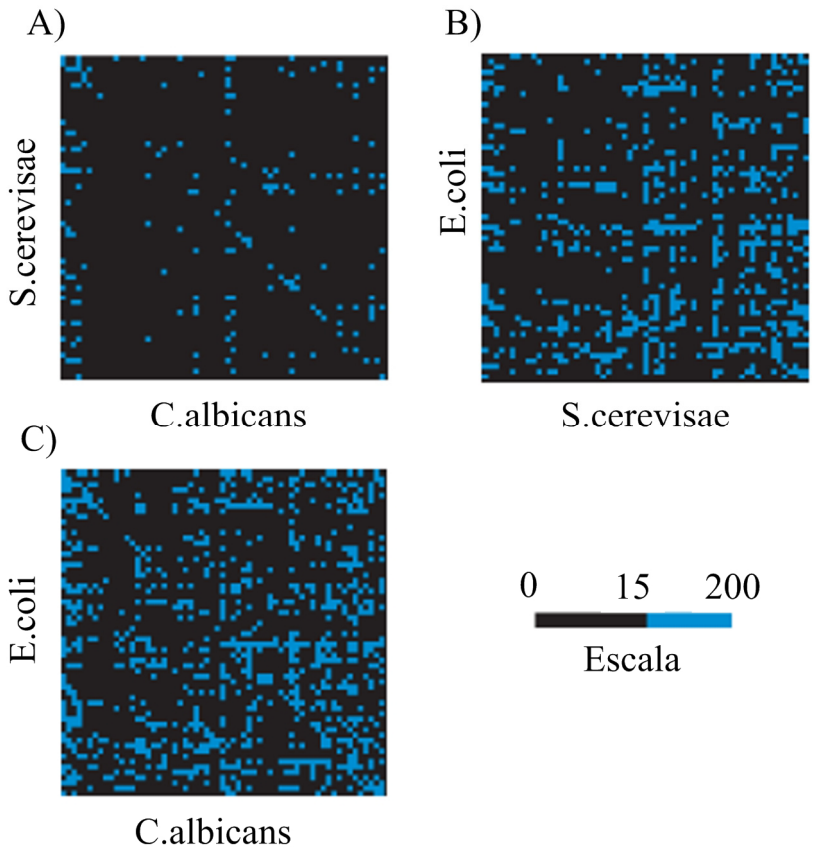

Figura 49 – Mapas contendo as diferenças entre valores residuais, possibilitando a comparação simultânea entre dois organismos. Estes valores reflectem o módulo da diferença dos valores residuais dos dois mapas em análise.

Nos três MDC visíveis na Figura 49, os contextos comuns entre os mapas estão coloridos a preto e as diferenças estão realçadas a azul. Como esperado, baseado nas distâncias da árvore filogenética, os MDC *E.coli*-*S.cerevisiae* e *E.coli*-*C.albicans* mostram mais diferenças do que o MDC do par *S.cerevisiae*-*C.albicans*. O MDC mostra também que o contexto é mais similar para o par *S.pombe*-*S.cerevisiae* do que para os outros dois pares formados pelas restantes leveduras, indicando assim que existem menos diferenças entre *S.pombe* e *S.cerevisiae* do que entre *C.albicans* e *S.cerevisiae*. Esta relação é

surpreendente, considerando que *S.pombe* divergiu da *S.cerevisiae* à 420 milhões de anos enquanto a *C.albicans* divergiu somente à 170 milhões de anos [150].

O efeito da diagonal verde do mapa de *C.albicans* influência o MDC do par *C.albicans*-*S.cerevisiae* e do par *C.albicans*-*E.Coli*, como se pode ver na Figura 49 (mapas a e c).

6.2.4 Influência dos níveis de GC na influência no contexto

Como a percentagem de GC tem um papel importante no *codon usage* [151] um estudo pormenorizado foi desenvolvido focando somente este aspecto. As diferenças entre níveis de GC são mais evidentes na terceira posição devido à degeneração do código genético pelo que limitamos o estudo a esta posição (GC_3). O objectivo era provar se as tendências de mutação contribuem para o contexto de pares de codões nos organismos de *S.cerevisiae* e *E.coli*.

O Anaconda está preparado para separar os genes de acordo com a percentagem de GC total, a percentagem de GC na primeira posição (GC_1), na segunda posição (GC_2) e na terceira posição (GC_3) dos codões. No decorrer da separação é possível criar grupos de genes com elevadas ou baixas percentagens de GC em qualquer posição ou no total, (Figura 50 A, B). Permite também quantificar os genes consoante o seu nível de GC nas três posições permitindo construir histogramas para facilitar a sua análise.

A distribuição dos genes no organismo *S.cerevisiae*, consoante os seus níveis de GC_3, varia entre os valores de 12% a 77%, no entanto a maioria dos genes está entre os 35-40% de GC_3 (Figura 50 A). No caso da *E.coli*, a distribuição do GC_3 está mais dispersa, variando de 20% a 90%, mas a maioria dos genes estão entre os 50 a 60% (Figura 50 B). Com esta distribuição é possível construir mapas de contexto contendo somente os genes com baixo ou elevado percentagem de GC3 para ambos organismos. Esses mapas poderão servir de base para a construção de mapas de diferença de contexto (MDC), permitindo identificar as diferenças de contexto consoante os valores de GC_3.

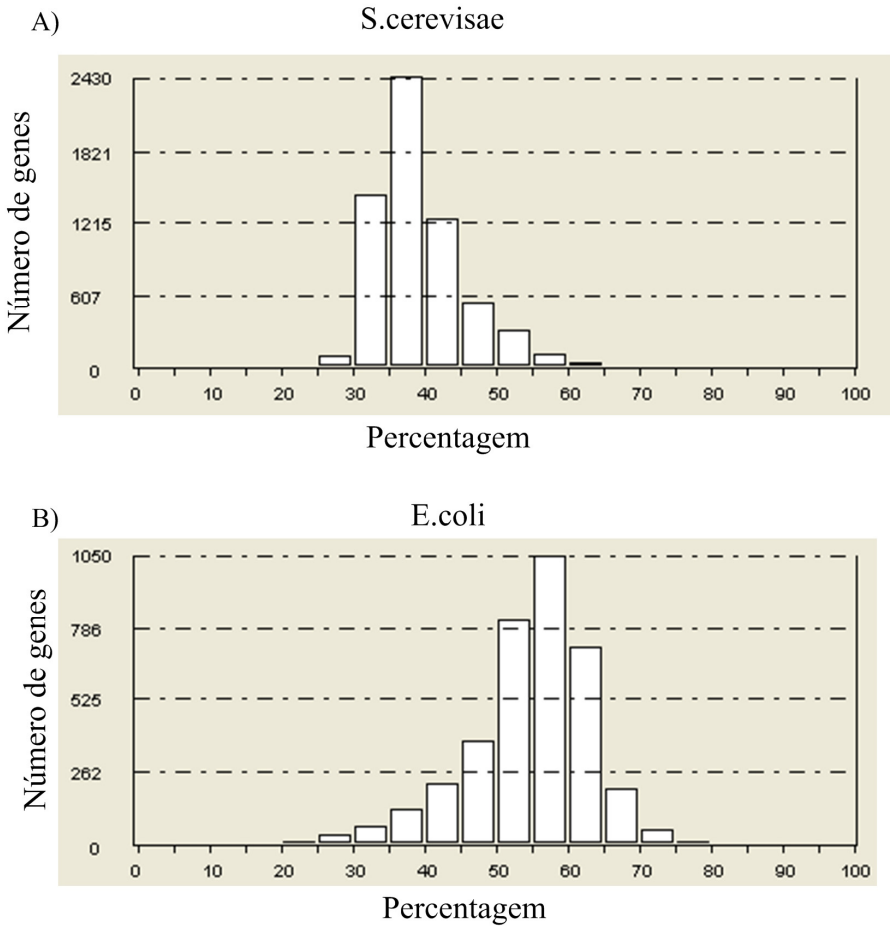

Figura 50 – Distribuição da percentagem de GC_3 nos genes pertencentes aos genomas *S.cerevisiae* e *E.coli*, visíveis na figura a) e b).

Os mapas de diferença de contexto (MDC) são necessários para compreender a importância das tendências introduzidas pelas mutações genéticas nos mapas de contexto. Esses mapas resultam do módulo da diferença entre dois mapas sendo criados a partir dos genes que contêm altas e baixas percentagem de GC_3. Serão coloridos com tonalidade de azul, variando somente de intensidade, como se pode ver na Figura 51.

Se as mutações no código genético forem aleatórias não contribuirão para a tendência do contexto, produzindo um MDC maioritariamente preto, pois os mapas contendo somente os genes com altos e baixos GC_3 serão idênticos a nível contextual, produzindo um resultado aproximado de zero para a maioria das células.

O mapa de diferença de contexto do organismo *S.cerevisiae* evidencia algumas diferenças, indicando que a tendência de GC_3 contribui para o contexto dos pares de codões. Contudo, algumas destas diferenças correspondem a pequenos desvios da preferência ou rejeição do

resíduo ajustado. Por outras palavras, os valores residuais têm o mesmo sinal, positivo ou negativo, mas o valor é mais elevado num dos mapas de GC_3. Em alguns casos, uma inversão de sinal nos resíduos ajustados, por exemplo de positivo para negativo, é detectado, indicando que o resíduo é positivo no mapa que foi construído com os genes com baixa percentagem de GC_3 e negativo no mapa que foi construído com os genes com elevada percentagem de GC_3. Esta inversão de sinal comprova a influência do GC nos mapas de contexto, como foi comprovado pelos valores da Tabela 5. Resultados similares foram obtidos para a E.coli, mas existe um maior número de inversão de sinal, indicando que a tendência GC é mais forte em E.coli do que em S.cerevisiae (Figura 51).

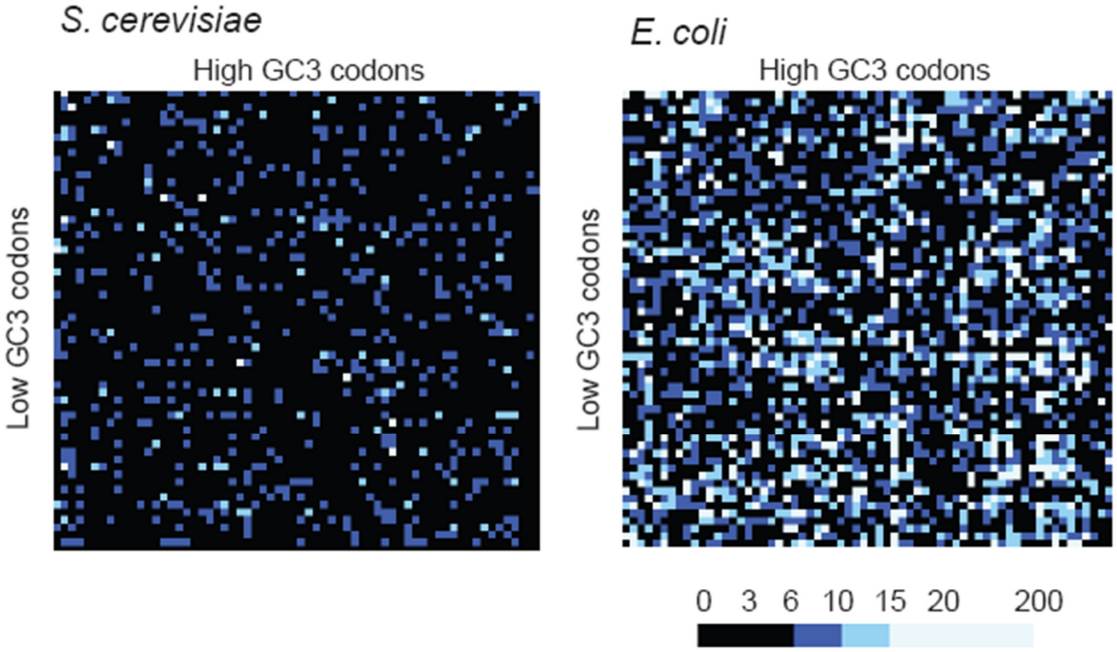

Figura 51 – Mapas de diferença de contexto resultantes da comparação entre dois mapas criados a partir de genes com elevada e baixa percentagem de GC_3.

6.3 Comparação do contexto de codões entre os três reinos

Usando o Anaconda, é possível comparar várias espécies em simultâneo. Como o objectivo era alargar o estudo do contexto de codões aos três reinos da vida foi efectuada uma análise comparativa entre 81 bactérias, 18 arqueas e 20 eucariotas. Para tal, os genomas dos 119 organismos foram lidos simultaneamente pelo Anaconda e os mapas de 61x64 codões para cada organismo foram convertidos num vector de 3904 elementos. Esses vectores foram

colocados lado a lado e aplicada a análise de *clustering*, evidenciando assim os padrões formados (Figura 52).

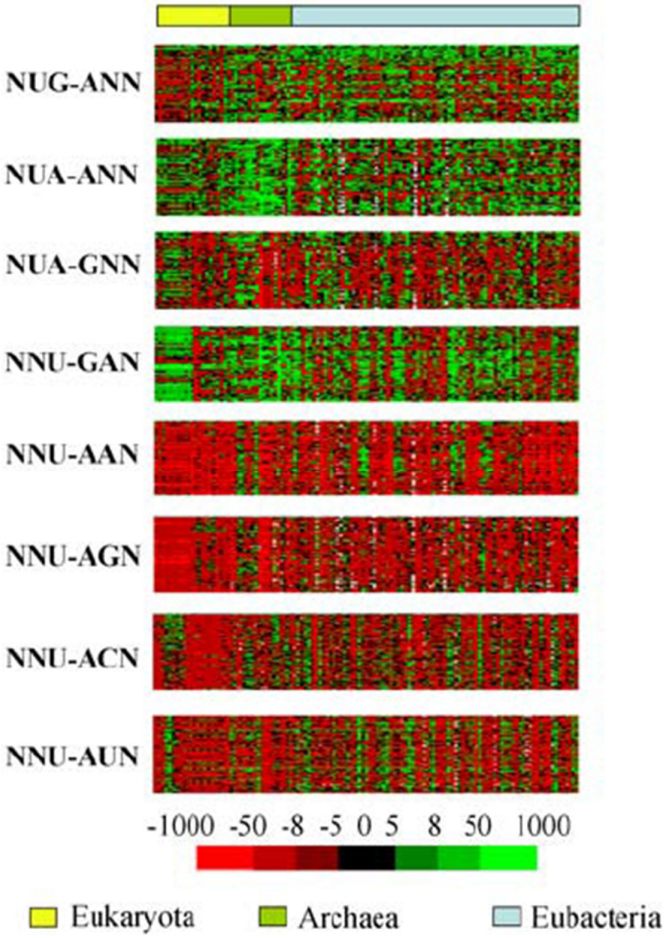

Figura 52 – Vários pares de dinucleótidos evidenciados para os 119 organismos em análise

A única regra universal detectada no mapa que abrange a totalidade dos 119 organismos foi xxU-Ayy. Esta regra inclui os codões de finalização quando ocorre uma falha de sincronismo, nomeadamente os codões UAA e UAG para as combinações yyU-AAy e xxU-AGy, estando relacionados com o fim prematuro da tradução. Para aprofundar esta característica, formaram-se subconjuntos retirados dos mapas de contexto que contêm os pares que representam codões de finalização fora de sincronismo (Figura 52). Esta estratégia revelou que as combinações xxU-AAy e xxU-AGy eram de facto as mais negativas em quase todos os organismos. No entanto, as combinações xxU-ACy e xxU-AUy que não contêm codões de finalização fora de sincronismo têm maioritariamente resíduos negativos.

As combinações xxU-GAy, xUA-Ayy e xUG-Ayy, as quais contêm codões de finalização fora de sincronismo, têm uma predominância de resíduos positivos. No entanto, as duas últimas combinações terão de ter uma falha de sincronismo de dois nucleótidos para ocorrer uma paragem prematura.

Visto que algumas regras com contexto positivo incluíam o dinucleótido U-A, é evidente que a influência do dinucleótido não é a única causa para a rejeição do contexto nestes pares de codões. Contudo, a finalização prematura da tradução, não é o único problema, porque a combinação xxU-ACy e xxU-AUy não contém nenhum codão de finalização e também é fortemente rejeitado.

6.4 ISA-Mediana

Face às limitações dos métodos de *clustering* outros métodos de formação de grupos foram aplicados às matrizes de valores residuais, nomeadamente o *biclustering*. Com o objectivo de comparar a eficiência dos algoritmos ISA e ISA-median foram efectuados testes nas matrizes de valores residuais da *S.cerevisiae*.

Quando os algoritmos de *biclustering* são aplicados a dados de *microarrays* o seu significado biológico é obtido através da análise do Gene Ontology (GO) ou através de redes de interacções proteína-proteína [131]. Mas como é possível obter significado estatístico numa matriz de contexto de codões contendo valores reais?

Para obter significado estatístico nos grupos encontrados através de *biclustering*, que não depende de nenhum dado biológico relevante, foi aplicado o conceito de matriz "muito densa" [152] sendo redefinida num teste de hipótese unilateral à direita. Dada uma matriz binária $m \times n$ com k uns, a sub-matriz B é densa se contém mais uns do que a matriz original. Então, a sub-matriz B pode ser considerada potencialmente significante, se o número observado de uns conduzir à rejeição da hipótese nula $H_{0,B}$: $p_B = p_0$ contra a hipótese alternativa $H_{1,B}$: $p_B > p_0$ onde p_B é a probabilidade de encontrar uns na sub matriz B com $p_0 = k/(mn)$. Para testar $H_{0,b}$ é calculado *p-value*:

$$p - value = 1 - \phi\left(\frac{\frac{|1_B|}{|B|} - p_0}{\sqrt{\frac{p_0(1-p_0)}{|B|}}}\right) \quad (10)$$

onde $|1_B|$ representa o número de uns da sub-matriz B e $|B|$ é o número de elementos em B.

Os algoritmos de biclustering identificam mais do que um grupo em cada corrida (conjunto de testes aplicados a uma matriz de dados). Dada uma matriz $X=[x_{ij}]$ é obtida a qualidade estatística de cada grupo testando a seguinte hipótese nula; H_0: "O algoritmo não identificou grupos densos".

Para tal, foi discretizada a matriz X numa matriz binária $B = [b_{ij}]$:

$$b_{ij} = \begin{cases} 1 \quad , \quad se \quad \left|\frac{x_{ij} - \hat{\mu}}{\hat{\sigma}}\right| > \lambda \\ 0 \quad , \quad outros \end{cases}$$

Se o algoritmo identificar n grupos, $B_1, B_2, \ldots, B_n$ a hipótese nula global H_0 é equivalente à intersecção de n hipóteses nulas; $H_{0,i}$:"O grupo B_i não é denso", i=1,2, … ,b. Neste contexto, o nível de significância ρ para cada grupo B_i encontrado é potencialmente significante se $p\text{-}value_{Bi} < \rho/n$.

Aplicando os algoritmos com diferentes parâmetros t_x e t_y à mesma matriz de valores residuais, foram obtidos diferentes resultados. Três cenários foram testados em separado procurando grupos com elementos positivos no eixo do x e y (X-positivo, Y-positivo), negativos no eixo dos x e y (X-negativo, Y-negativo) e compostos (X-módulo, Y-positivo). As funções aplicadas aos eixos correspondem às funções g definidas da seguinte forma: i) g(x,y)= (x-y); ii) g(x,y)=-(x-y); iii) g(x,y)=|x-y| consoante a procura de valores positivos, negativos ou em módulo respectivamente, conforme a descrição do algoritmo que consta na secção 4.3.2.

Para cada algoritmo foi aplicada uma corrida de 500 testes à matriz de 61x64 codões, do organismo *S.cerevisiae*, para vários valores de t_x e t_y com $t_x=t_y$. O algoritmo ISA-original encontra um maior número de grupos, para todas as condições, como se pode ver nos gráficos da Figura 53. No entanto, um maior número de grupos não significa que sejam

mais significativos estatisticamente nem proporcionam um meio para obter regras de contexto.

Quando mais afastada estiver a linha de significância da linha da quantidade menos significado têm os grupos para o ponto t_x=t_y. No caso (X-negativo, Y-negativo) para o ISA-original pode-se constatar que todos os grupos identificados têm pouco significado. O ISA-mediana tem melhor comportamento a nível geral, relevando um melhor comportamento na relação quantidade de grupos/significado estatístico. Somente no caso (X-positivo, Y-positivo) o ISA-original tem um melhor comportamento.

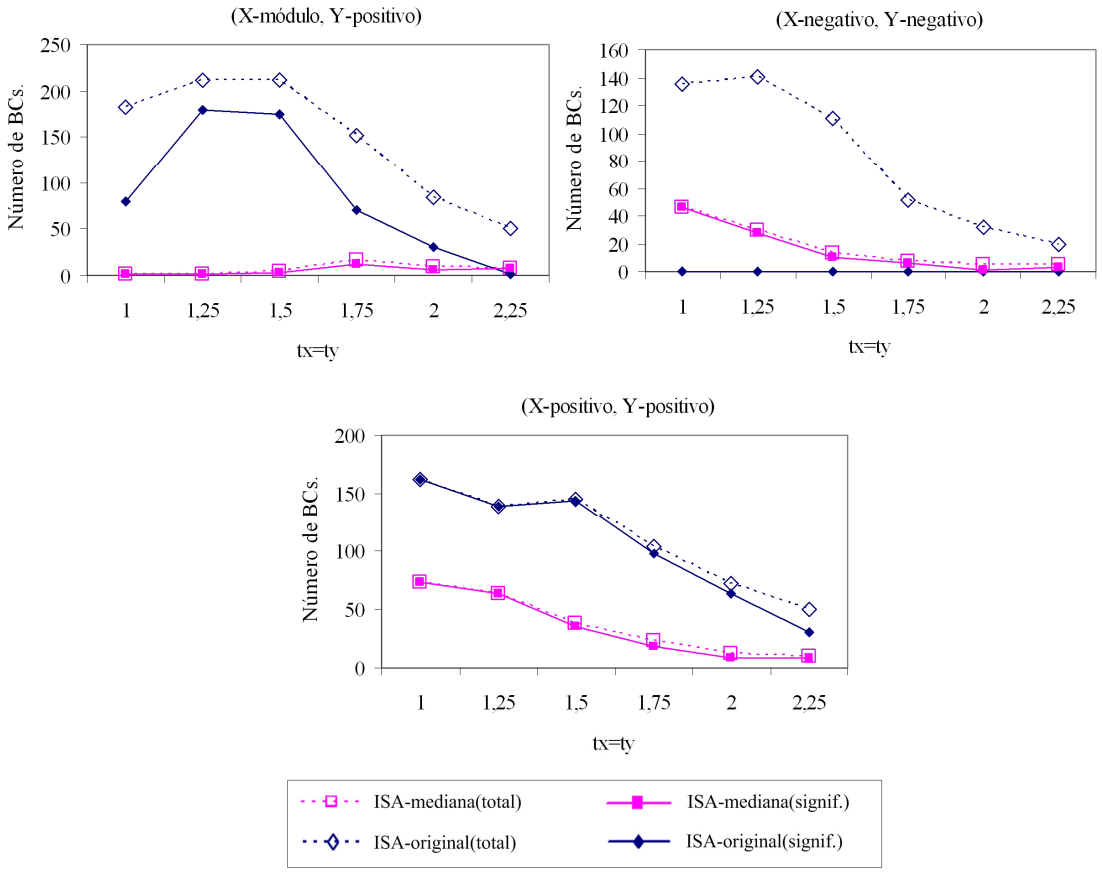

Figura 53 - Os gráficos representam os valores obtidos com a aplicação dos algoritmos ISA-original e ISA-mediana. As linhas a tracejado representam o número de grupos para vários T_x e T_y, e as linhas sem tracejado representa o seu significado estatístico.

Devido à necessidade de identificar quais os grupos que foram encontrados em ambos algoritmos, foi calculado um *score*, com o intuito de comparar a capacidade de um algoritmo encontrar grupos já identificados pelo outro.

Para tal, foi aplicada a seguinte equação:

$$S(M_1, M_2) = \frac{1}{|M_1|} \sum_{(R_1,C_1)\in M_1} \max_{(R_2,C_2)\in M_2} \frac{|R_1 \cap R_2| + |C_1 \cap C_2|}{|R_1| + |C_1|} \qquad (11)$$

onde M_1 e M_2 correspondem aos grupos obtidos através dos algoritmos $M_{\overline{X}}$ e $M_{Q\frac{1}{2}}$ que equivalem ao ISA-original e ISA-mediana respectivamente. O *score* obtido quantifica se o grupo identificado pelo algoritmo A está contido em algum grupo identificado pelo grupo B. Este *score* também contempla o caso de sobreposição de grupos de diferentes dimensões.

A Figura 54 contém os *scores* obtidos para os dois algoritmos em análise. No entanto, foram criados dois grupos distintos, um com a totalidade dos grupos encontrados e outro contendo somente os grupos com significado estatístico.

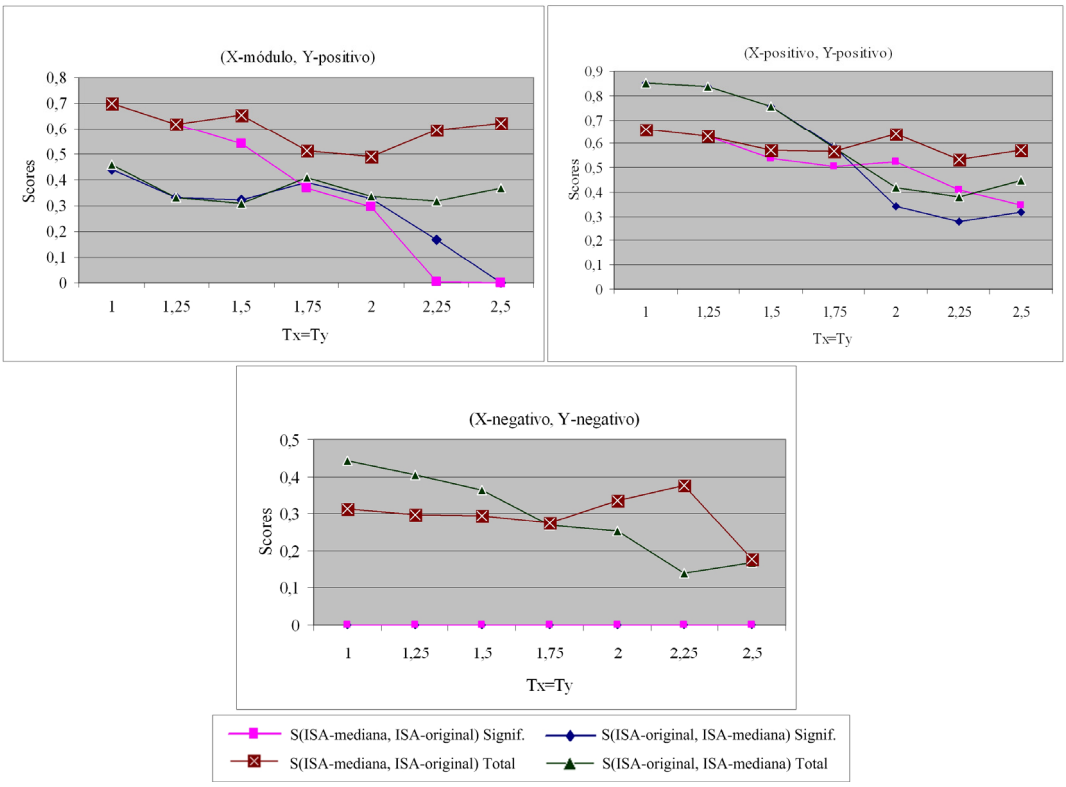

Figura 54 – Vários scores obtidos para os dois algoritmos em análise. Dois conjuntos distintos foram formados, um contendo todos os grupos encontrados pelos algoritmos e outro somente com os grupos com significado estatístico.

Os resultados reflectem que o ISA-mediana consegue recuperar os grupos identificados pelo ISA-original mesmo obtendo um número significativamente menor de grupos para cada corrida quando comparado com o ISA-original.

A Figura 55 contém dois grupos formados pelo algoritmo ISA-mediana aplicado à matriz de valores residuais da *S.cerevisiae*. O primeiro grupo tinha já sido identificado recorrendo ao *clustering* hierárquico. No entanto, o segundo grupo não tinha sido identificado através do *clustering* hierárquico. Este novo grupo, permite a possibilidade de criação de uma regra de contexto para este organismo. Contudo, estes grupos presentes na figura não foram identificados através do ISA-original. No entanto, o ISA-original identificou outros grupos mas sem a possibilidade de criação de regras de contexto.

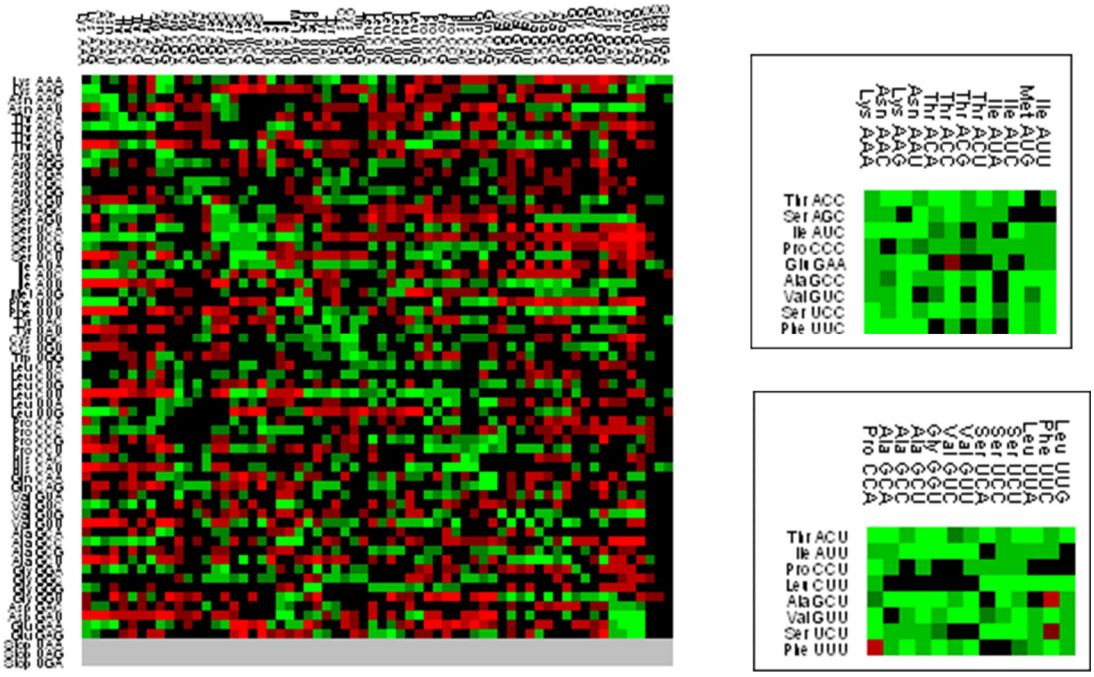

Figura 55 – Mapa de contexto do organismo *S.cerevisae* com dois grupos formados através do *biclustering* ISA-mediana. O primeiro grupo já tinha sido identificado pelo *clustering* hierárquico.

O algoritmo do ISA-mediana foi também aplicado a dados de *microarrays*, tendo obtido piores resultados em relação ao ISA-original [153]. Isto deve-se ao facto de o número de linhas ser muito superior às matrizes de valores residuais, tendo raramente atingido o critério de paragem do algoritmo. Quando o critério de paragem é conseguido, para o ISA-mediana, são obtidos poucos grupos com um elevado número de linhas e colunas, não tendo qualquer significado. Em oposição quando o algoritmo ISA-original é aplicado a este tipo de dados vários grupos são detectados.

6.5 Conclusões

No presente capítulo foram apresentados alguns resultados obtidos através da aplicação Anaconda e que, de certa forma validam o seu desempenho.

Foi inicialmente efectuado um estudo comparativo sobre três leveduras eucariotas *Saccharomyces cerevisiae*, *Candida albicans* e *Schizosaccharomyces pombe* e uma bactéria *Escherichia coli*. Seguidamente foi efectuado um estudo comparativo entre 119 espécies pertencentes aos três reinos da vida, eucariota, bactérias e arqueas, identificando pontos comuns aos três reinos. Posteriormente, foi apresentado como é possível efectuar estudos recorrendo a genes homólogos entre espécies e, para finalizar, foram apresentados alguns resultados obtidos comparando o ISA original com o ISA mediana.

No capítulo seguinte, apresentamos as conclusões do trabalho desenvolvido e também algumas linhas de intervenção que poderão merecer atenção em trabalho futuro.

Capítulo 7

7 Conclusões e Trabalho Futuro

Com a evolução da tecnologia foi possível sequenciar organismos em larga escala proporcionando oportunidades únicas para estudar o funcionamento e a evolução dos seres vivos. No entanto, enormes quantidades de informação são geradas, criando novos desafios cuja solução passa pelo desenvolvimento de metodologias matemáticas e ferramentas bioinformáticas capazes de lidar com grandes volumes de informação.

Os organismos são compostos por uma ou mais células sendo o seu funcionamento regulado por informação que está contida dentro da própria célula. Devido aos mecanismos que ocorrem dentro das células serem de tal forma complexos é necessária subdividir as diversas áreas para compreender o seu funcionamento. Vários processos ocorrem dentro da célula, sendo o processo de tradução bastante importante para o seu bom funcionamento. Este mecanismo está presente em todos os organismos vivos pois possibilita a passagem da informação presente nas sequências genéticas para estruturas tridimensionais funcionais, conhecidas por proteínas. O ribossoma, entidade responsável pela tradução, faz corresponder a cada combinação de três letras, conhecido como codão, um anti-codão associado a um aminoácido. Como resultado do processo, os aminoácidos vão-se ligando entre si formado uma cadeia polipeptídica. O produto final irá corresponder a uma proteína útil para a célula.

Este trabalho centrou-se no estudo do contexto de codões, característica que influencia a tradução dos genes para proteínas. Tendo como facto que a frequência dos codões nos

genes não é uniformemente distribuída, e diferindo de organismo para organismo, levanta-se a hipótese de que o contexto dos codões influencia a velocidade e fidelidade de descodificação. Em *E.coli* os pares de codões com menos ocorrências são traduzidos com menor velocidade do que os pares de codões mais representados, indicando que o contexto influencia a velocidade de tradução [4]. Este facto sugere que o contexto em *E.coli* está sobre forte pressão pela maquinaria de tradução.

Para estudar estas e outras propriedades do genoma desenvolveu-se um modelo apoiado em metodologias e ferramentas informáticas, para o estudo do contexto dos codões à escala genómica. O sistema bioinformático desenvolvido lê todos os genes presentes num determinado genoma, independentemente do seu número total, e simula a leitura dos codões pelos anti-codões no ribossoma. Ao fazê-lo memoriza os codões vizinhos e constrói uma tabela de frequências de contexto que pode ser tratada estatisticamente. O sistema constrói automaticamente várias tabelas de contingência, calculando à posteriori os valores residuais das mesmas.

7.1 Contribuições

Até ao momento nenhum trabalho exaustivo tinha sido desenvolvido para estudar o contexto dos codões nas zonas codificantes. Com este trabalho foi possível efectuar vários estudos a nível contextual e relacionar essa informação com outros índices já disponíveis na comunidade científica.

A aplicação encontra-se em exploração nos laboratórios do departamento de biologia da Universidade de Aveiro tendo sido obtidos resultados promissores sobre as leis gerais que governam a fidelidade de descodificação do código genético. Actualmente a informação obtida está a ser validada *in vivo* no laboratório de genómica funcional.

A arquitectura do Anaconda foi desenhada de forma a suportar novas funcionalidades que iriam sendo adicionadas consoante as necessidades. No entanto, permite ainda adicionar outras funcionalidades que poderão tornar-se necessárias.

O Anaconda foi desenvolvido baseado em paradigmas visuais conhecidos, tornando a interacção com os utilizadores acessível.

Vários métodos estatísticos foram incluídos na aplicação, como *clustering*, *biclustering*, valores residuais, entre outros.

O Anaconda é uma aplicação que permite efectuar caracterizações de genes, obtendo vários índices, podendo também efectuar comparações entre ambos. É possível também redesenhar genes, com base nos vários valores estatísticos obtidos através da aplicação.

Várias publicações e participações em congressos foram conseguidas com base nos resultados obtidos com o Anaconda.

Com este modelo foi possível obter várias associações entre codões, quer nas regiões onde o DNA codifica proteínas como nas regiões não codificantes. Disponibiliza também variada informação sobre a estrutura primária dos genes, nomeadamente índices CAI, codões raros, distribuição de codões nas zonas codificantes ou aminoácidos sequencialmente repetidos, entre outros resultados. Os resultados são visualizados através de um paradigma bastante familiar, facilitando a interacção com o utilizador.

Várias regras foram obtidas em resultado de análises efectuadas em diversos organismos. Uma regra que é realçada com os grupos formados através de análise de *clustering* no mapa do organismo *S.cerevisiae* é a xxU-Ayy, contendo algumas excepções. Por exemplo, com o padrão xxU-Ayy que evidencia a rejeição, os pares de codões AAU-AGC, AAU-AGU, AAU-AAU, AAU-AAC e o conjunto AGU-AGC, AGU-AGU, AGU-AAU, AGU-ACA, AGU-AUA tem resíduos positivos indicando que estes pares são preferidos pelo genoma.

A análise de *clustering* nos mapas de contexto mostra também que alguns dos grupos formados são criados com base em regras de dinucleótidos, nomeadamente xxU-Ayy, xxC-Ayy e xxU-Gyy.

Uma das importantes características que evidencia tendências de mutação no contexto de codões são as percentagens de GC, em particular o GC3. O GC tem uma forte influência no *codon usage* e em casos extremos consegue reduzir o número, ou mesmo eliminar, certos codões presentes nas zonas codificantes. Os resultados mostram claramente que o GC3 afecta o contexto, no entanto este efeito é mais visível para pares de codões que têm baixos valores residuais.

Além dos casos mencionados anteriormente outras características contribuem para o contexto, tendo sido evidenciadas pela aplicação Anaconda para diferentes organismos. Por exemplo, o contexto nas leveduras evidenciou características nos organismos eucariotas que não está relacionado com a tradução dos genes, mais concretamente, a preferência dele mesmo para seu vizinho, originando uma diagonal verde nos mapas de contexto. Ainda está por comprovar em laboratório experimentalmente se estas repetições melhoram a eficiência ou precisão no processo de tradução nas leveduras, pois não existem até ao momento testes específicos que indiquem o aumento de eficácia nestas regiões.

Nas análises comparativas demonstrou-se que o processo de replicação assim como o processo de tradução influenciam o contexto dos codões. Com esta aproximação ficou comprovada a importância da influência das zonas codificantes para o contexto, mas comprovou-se também que o processo de tradução também influência o contexto.

Os estudos realizados, tanto para as regiões codificantes como para as regiões não codificantes, produziram semelhantes padrões estando em conformidade com estudos anteriormente apresentados [154]. Estes resultados implicam que muitos dos constrangimentos detectados nas regiões codificantes não são impostos pela maquinaria de tradução, mas resultam da pressão selectiva da replicação do DNA.

A tendência do *codon usage* é principalmente mantida pela pressão imposta pela mutação e em segundo plano está a influência imposta pelo processo de tradução, confirmando assim a relevância da replicação do DNA na influência do contexto de codões [155].

A análise efectuada aos 119 organismos em relação à influência dos dinucleótidos confirma uma clara rejeição da metilação nas sequências C-G codificantes nos eucariotas. A metilação do DNA torna-o indisponível para a transcrição e consequentemente para a tradução [156]. Por outro lado, os dinucleótidos U-A são extremamente reprimidos no conjunto dos 119 organismos.

Como anteriormente foi afirmado, somente uma única regra geral foi detectada para os 119 organismos analisados, rejeitando os pares de codões com a combinação xxU-Ayy. Claramente, esta tendência é o resultado da repressão do dinucleótido U-A nas sequências dos organismos. No entanto, outros contextos que têm o dinucleótido U-A não são fortemente rejeitados. Por exemplo, o contexto xUA-Ayy é geralmente preferido nas sequências, indicando que existem diferenças entre contextos que contêm o dinucleótido

U-A. Este facto sugere que o processo de tradução influencia a escolha dos pares de codões.

O estudo da evolução é uma das áreas onde a sequenciação proporcionou grandes avanços, pois recorre às sequências genéticas para efectuar essas análises. Com esse objectivo em mente, foi desenvolvido um método para efectuar estudos de evolução recorrendo ao contexto de codões. Através do Anaconda já foi possível efectuar um estudo de evolução entre organismos, recorrendo aos seus genes homólogos, pertencendo aos três reinos da vida e estando em fase de submissão para publicação.

Na parte de alteração das regiões codificantes uma solução foi proposta, tendo sido desenhada de forma a proporcionar a aplicação da informação obtida. No entanto, os resultados ainda terão de ser testados em laboratório para poderem ser posteriormente publicados.

No campo dos algoritmos de *biclustering*, o algoritmo ISA foi analisado em detalhe, encontrando algumas desvantagens e propondo algumas alterações de forma a melhorar o seu desempenho em matrizes de menor dimensão. O novo algoritmo foi implementado com sucesso estando disponível para ser utilizado pela comunidade científica através da aplicação desenvolvida. Efectuou-se também um estudo exaustivo de comparação entre o algoritmo proposto e o algoritmo original tendo o trabalho sido submetido a uma revista científica.

Vários índices já disponíveis para caracterizar as zonas codificantes foram também inseridos no modelo, como os valores de tRNA e *codon usage*, possibilitando relacioná-los com os valores residuais.

Com este novo modelo, suportado pela aplicação Anaconda, é possível caracterizar exaustivamente vários genomas possibilitando ainda correlacioná-los entre si, algo que não era possível até à presente aplicação. A informação é ainda passível de ser exportada podendo ser utilizada por outras aplicações em posteriores análises.

7.2 Perspectivas de Trabalho Futuro

Uma das direcções mais promissoras para desenvolvimentos futuros passa pelo redesenho de genes. Trata-se de um aspecto muito complexo que merecia, por si só, uma abordagem

noutra dissertação. Embora tenha sido apresentada uma solução para redesenho de genes é possível melhorar significativamente a ferramenta proposta. Esse trabalho passaria também por testar as sequências alteradas, baseadas nas informações fornecidas pelo Anaconda, em testes de laboratório para identificar o seu comportamento.

Outro dos caminhos possíveis de investigação será prever o comportamento dos valores residuais nas estruturas tridimensionais das proteínas. Durante o processo de tradução a fase final é composta pelo enrolamento, isto é, quando a proteína adquire a sua forma final. Actualmente, várias proteínas já têm a sua representação em três dimensões, podendo ser obtida em base de dados especializadas em estruturas tridimensionais. Durante o processo de enrolamento, certas zonas necessitam de mais tempo para adquirir a forma. Nestes pontos específicos o ribossoma necessita de abrandar o processamento para a sequência peptídica ganhar a forma correcta [157]. Logo, estas zonas terão de ser desfavoráveis no ponto de vista ribossomal. Uma das proposta seria identificar as zonas de enrolamento das proteínas e compreender quais as implicações que essas zonas têm nos valores de contexto. Estas regiões certamente terão uma maior incidência de valores contextuais negativos, reflectindo o abrandamento do ribossoma nestas posições específicas.

8 Anexos

8.1 Códigos IUPAC para os aminoácidos

Abreviação a uma letra	Abreviação a três letras	Descrição
A	Ala	Alanina
R	Arg	Arginina
N	Asn	Asparagina
D	Asp	Ácido aspártico
C	Cys	Cisteína
Q	Gln	Glutamina
E	Glu	Ácido glutâmico
G	Gly	Glicina
H	His	Histidina
I	Ile	Isoleucina
L	Leu	Leucina
K	Lys	Lisina
M	Met	Metionina
F	Phe	Fenilalanina
P	Pro	Prolina
U	Sec	Selenocisteína
S	Ser	Serina
T	Thr	Treonina
W	Trp	Triptofano
Y	Tyr	Tirosina
V	Val	Valina
B	Asx	Ácido aspártico ou asparagina
Z	Glx	Glutamina ou acido glutâmico
X	Xaa	Qualquer aminoácido

8.2 Código IUPAC para os nucleótidos

Código	Descrição
A	Adenina
C	Citosina
G	Guanina
T	Timina
U	Uracilo
R	Purina (A ou G)
Y	Pirimidina (C, T, ou U)
M	C ou A
K	T, U, ou G
W	T, U, ou A
S	C ou G
B	C, T, U, ou G (não A)
D	A, T, U, ou G (não C)
H	A, T, U, ou C (não G)
V	A, C, ou G (não T, não U)
X	A, C, G, T, or U

8.3 Lista dos ficheiros que contêm dados biológicos

Origem	Extensão	Observações
ACE files	.ace	sequências consenso
Phylip Alignment	.phy	alinhamentos
Clustal Alignment	.aln	alinhamentos
FASTA	.fsa/.fasta/.faa/.fna/.ffn	sequências
GenBank	.gbk/.gb/.gp/.gbs	sequências anotadas
PIR (NBRF)	.pir	sequências
DNAstrider	.str/.strider	sequências
Swiss-Prot	.swp	sequências de proteínas
Embl	.embl	sequências de nucleótidos
mmCIF	.cif	estruturas
PDB	.pdb	estruturas
BLAST Database	.phr/.nhr	base de dados BLAST
RNA Structure	.ct/.col/.rnaml	estruturas de RNA

8.4 Matrizes de substituição

As sequências biológicas evoluíram através dos tempos e o seu estudo demonstrou que nem todas as transformações ocorrem com a mesma probabilidade. Certas substituições, alteração de um aminoácido por outro, são mais comuns, enquanto outras ocorrem com menos frequência. Por essa razão existe diferentes pesos para cada alteração.

As matrizes mais comuns são a *BLOcks SUbstitution Matrix* (BLOSUM) [158] e a *Point Accepted Mutation* (PAM) [159].

Tabela 6 – Tabela com os valores de substituição que correspondem à BLOSUM62

Amino Acids	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	**4**	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0
R	-1	**5**	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3
N	-2	0	**6**	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3
D	-2	-2	1	**6**	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
C	0	-3	-3	-3	**9**	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1
Q	-1	1	0	0	-3	**5**	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2
E	-1	0	0	2	-4	2	**5**	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2
G	0	-2	0	-1	-3	-2	-2	**6**	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3
H	-2	0	1	-1	-3	0	0	-2	**8**	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	**4**	2	-3	1	0	-3	-2	-1	-3	-1	3
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	**4**	-2	2	0	-3	-2	-1	-2	-1	1
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	**5**	-1	-3	-1	0	-1	-3	-2	-2
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	**5**	0	-2	-1	-1	-1	-1	1
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	**6**	-4	-2	-2	1	3	-1
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	**7**	-1	-1	-4	-3	-2
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	**4**	1	-3	-2	-2
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	**5**	-2	-2	0
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	**11**	2	-3
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	**7**	-1
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	**4**

Baseado na evolução das proteínas ficou patente que estas alterações podem ser quantificadas por uma matriz de pesos, também chamada matriz de substituição. A Tabela 6 contém os valores de substituição para todos os aminoácidos para a BLOSUM62. Por

exemplo, como o aminoácido Triptofano (W) é um aminoácido raro, e como só em raras ocasiões se altera para uma Leucina (L) tem o peso -2. Quanto mais negativo é o peso mais penalização existe nessa substituição.

A matriz PAM85 foi publicada em 1978 e foi construída através do alinhamento global de sequências que tinham 85% de similaridade [159].

Há algumas limitações quanto às matrizes PAM que tornam as matrizes BLOSUM mais atractivas. As sequências com que foram construídas as matrizes PAM já são muito antigas e o método de cálculo da PAM assume que todos os aminoácidos têm o mesmo rácio de mutação, o que não acontece com as matizes BLOSUM. Em 1992, catorze anos depois das matrizes PAM serem publicadas, as matrizes BLOSUM foram desenvolvidas e publicadas. As BLOSUM foram desenvolvidas baseadas num modelo onde se prevêem sequências mais divergentes e utiliza alinhamentos locais. Por exemplo, a BLOSUM62 foi construída com sequências que não contenham menos de 62% de identidade [160].

Decidir qual das matrizes a utilizar para obter o melhor alinhamento não é uma tarefa fácil. No entanto, existem algumas regras básicas para a escolha da matriz mais apropriada. Se não existir qualquer conhecimento das sequências envolvidas na procura, certamente a BLOSUM62 é a melhor escolha. Na procura de sequências próximas com a sequência de procura poderá escolher-se a BLOSUM80 ou a PAM1. Se a procura estiver centrada em sequências mais distantes em relação à sequência de procura, a BLOSUM45 ou a PAM250 será a escolha mais correcta

As matrizes BLOSUM com baixos valores correspondem às matrizes PAM com altos valores como se pode ver pela Figura 56.

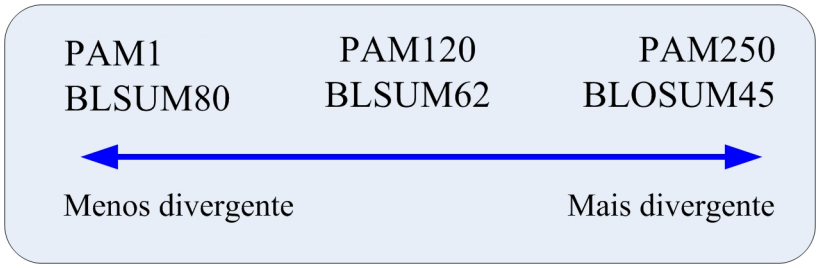

Figura 56 – Relação entre matrizes de substituição. A BLOSUM62 estabeleceu-se como matriz standard para a maioria das aplicações de alinhamento. É a matriz por defeito no BLAST.

As matrizes apresentadas anteriormente são utilizadas nas sequências de aminoácidos. Quando a procura é efectuada em sequências de nucleótidos, geralmente utiliza-se a matriz da Tabela 7. Por cada conjugação é somado 5 ao *score* e por cada falha é somado -4 ao *score*.

Tabela 7 – Matriz de substituição quando envolve nucleótidos

Nucleótidos	A	T	G	C
A	**5**	-4	-4	-4
T	-4	**5**	-4	-4
G	-4	-4	**5**	-4
C	-4	-4	-4	**5**

8.5 Aminoácidos agrupados pelas suas propriedades

A Figura 57 mostra um diagrama de Venn agrupando os aminoácidos de acordo com as suas propriedades. Este diagrama foi adaptado a partir de uma proposta feita por Livingstone em 1993 [161] e é apenas uma das classificações possíveis.

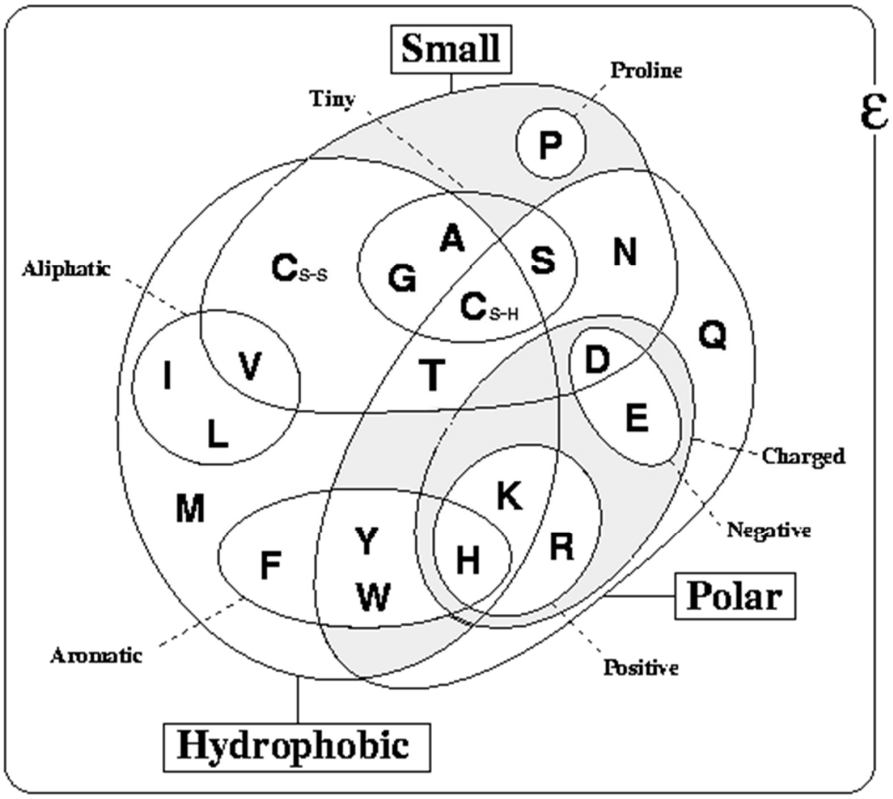

Figura 57 – Os vários aminoácidos agrupados pelas suas propriedades físicas

8.6 Formação de mapas de pares de contexto

Devido à necessidade de procura de similaridades de contexto entre organismos foi construído um mapa que agregasse todas as matrizes de contexto, utilizando a matriz de cada organismo. A Figura 58 contém os passos detalhados dessa transformação: i) a matriz de contexto de cada organismo é transformada num vector de 3904 posições, correspondendo cada posição a um par de codões; ii) as colunas são colocadas lado a lado criando assim o mapa de pares de contexto. Aplicando análise *clustering* e *biclustering* é possível evidenciar possíveis padrões que são transversais aos organismos em estudo.

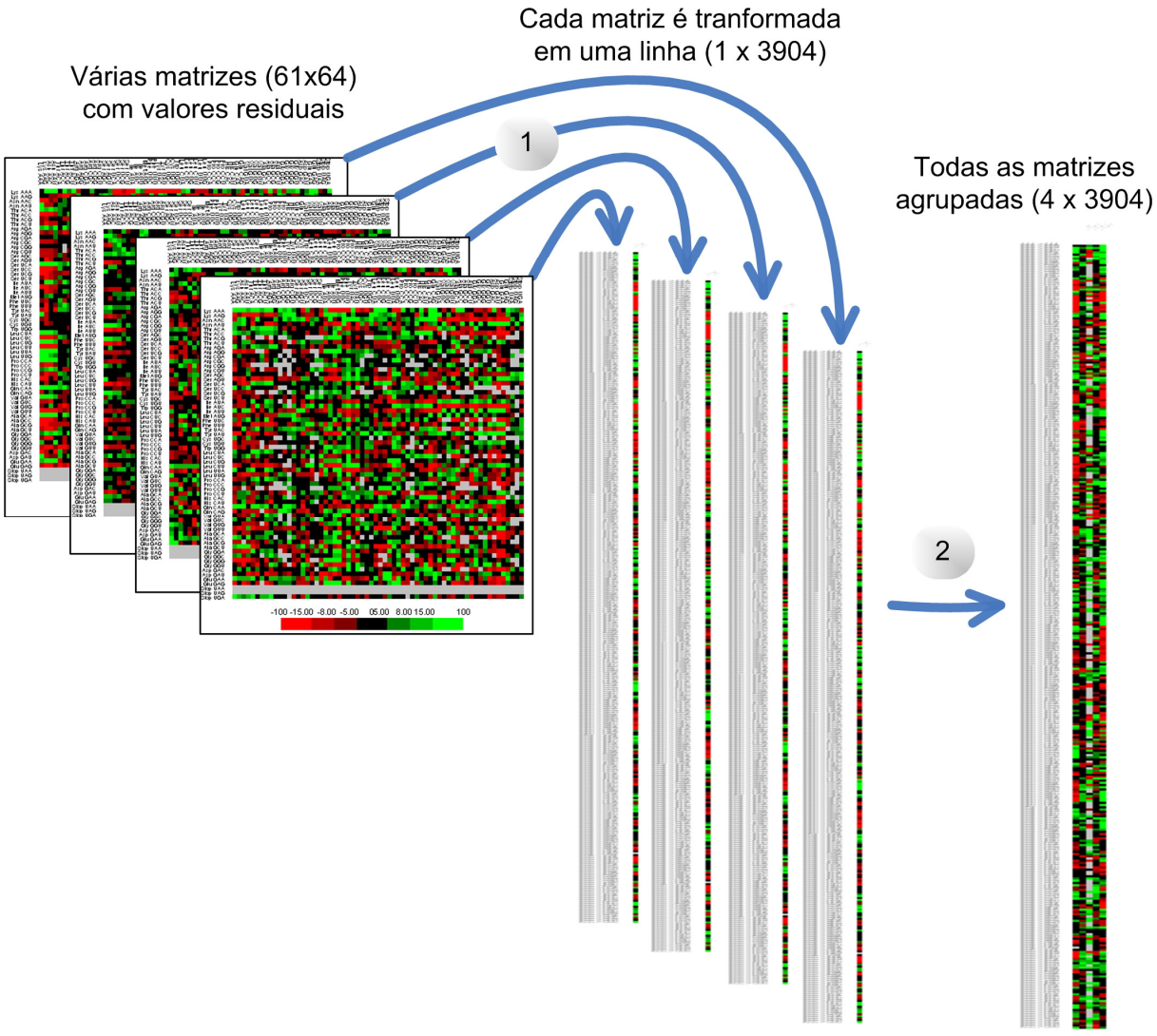

Figura 58 - Processo de formação de mapas de contexto

9 Bibliografia

[1] F. Sanger, G. M. Air, B. G. Barrell, N. L. Brown, A. R. Coulson, C. A. Fiddes, C. A. Hutchison, P. M. Slocombe, and M. Smith, "Nucleotide sequence of bacteriophage phi X174 DNA," *Nature,* vol. 265, pp. 687-95, Feb 24 1977.

[2] J. Bennetzen and B. Hall, "Codon selection in yeast.," *J Biol Chem.,* vol. 257, pp. 3026-31, 1982.

[3] M. Gouy and C. Gautier, "Codon usage in bacteria: correlation with gene expressivity," *Nucleic Acids Res,* vol. 10, pp. 7055-74, Nov 25 1982.

[4] B. Irwin, J. D. Heck, and G. Wesley, "Codon Pair Utilization Biases Influence Translational Elongation Step Times," *The Journal of Biological Chemistry,* vol. 270, pp. 22801-22806, 1995.

[5] E. T. Young, J. S. Sloan, and K. V. Riper, "Trinucleotide repeats are clustered in regulatory genes in Saccharomyces cerevisiae," *Genetics,* vol. 154, pp. 1053-68, 2000.

[6] L. S. Folley and M. Yarus, "Codon contexts from weakly expressed genes reduce expression in vivo," *J. Mol. Biol.,* vol. 209, pp. 359-78, 1989.

[7] *Grande Dicionário da Língua Portuguesa*: Porto Editora, 2004.

[8] B. Alberts, D. Bray, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Essencial cell Biology. An introduction to the Molecular Biology of the Cell*. New York & London: Garland Publishing, Inc., 1998.

[9] D. L. Hartl and E. W. Jones, *Genetics: Analysis of Genes and Genomes*, 6 edition ed. Sudbury: Jones & Bartlett Publishers, August 2004.

[10] M. Schwartz and J. Vissing, "Paternal inheritance of mitochondrial DNA," *N Engl J Med,* vol. 347, pp. 576-80, Aug 22 2002.

[11] D. L. Hartl and E. W. Jones, *Genetics, Analysis of Genes and Genomes*. Sudbury: Jones and Bartlett Publishers, Inc., 2001.

[12] A. Elzanowski and J. Ostel, "The Genetic Codes." vol. 2008: National Center for Biotechnology Information (NCBI), 2008.

[13] J. Shine and L. Dalgarno, "Determinant of cistron specificity in bacterial ribosomes," *Nature,* vol. 254, pp. 34-8, Mar 6 1975.

[14] J. D. Watson and F. H. Crick, "Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid," *Nature,* vol. 171, pp. 737-8, Apr 25 1953.

[15] G. Varani and W. H. McClain, "The G x U wobble base pair. A fundamental building block of RNA structure crucial to RNA function in diverse biological systems," *EMBO Rep,* vol. 1, pp. 18-23, Jul 2000.

[16] P. Licznar, N. Mejlhede, M. F. Prere, N. Wills, R. F. Gesteland, J. F. Atkins, and O. Fayet, "Programmed translational -1 frameshifting on hexanucleotide motifs and the wobble properties of tRNAs," *EMBO J,* vol. 22, pp. 4770-8, Sep 15 2003.

[17] C. Marck and H. Grosjean, "tRNomics: analysis of tRNA genes from 50 genomes of Eukarya, Archaea, and Bacteria reveals anticodon-sparing strategies and domain-specific features," *RNA,* vol. 8, pp. 1189-232, Oct 2002.

[18] D. L. Hartl and E. W. Jones, *Genetics: Analysis of Genes and Genomes*, 6 edition ed. Sudbury: Jones & Bartlett Publishers, 2005.

[19] E. B. Kramer and P. J. Farabaugh, "The frequency of translational misreading errors in E. coli is largely determined by tRNA competition," *RNA,* vol. 13, pp. 87-96, Jan 2007.

[20] D. V. Freistroffer, M. Kwiatkowski, R. H. Buckingham, and M. Ehrenberg, "The accuracy of codon recognition by polypeptide release factors," *Proc Natl Acad Sci U S A,* vol. 97, pp. 2046-51, Feb 29 2000.

[21] R. H. Buckingham and H. Grosjean, *The accuracy of mRNA-tRNA recognition.* vol. 41. London: Chapman and Hall, 1986.

[22] M. F. Princiotta, D. Finzi, S. B. Qian, J. Gibbs, S. Schuchmann, F. Buttgereit, J. R. Bennink, and J. W. Yewdell, "Quantitating protein synthesis, degradation, and endogenous antigen processing," *Immunity,* vol. 18, pp. 343-54, Mar 2003.

[23] F. M. Wurm, "Production of recombinant protein therapeutics in cultivated mammalian cells," *Nat Biotechnol,* vol. 22, pp. 1393-8, Nov 2004.

[24] C. Gustafsson, S. Govindarajan, and J. Minshull, "Putting engineering back into protein engineering: bioinformatic approaches to catalyst design.," *Curr Opin Biotechnol.,* vol. 14, pp. 366-70, 2003.

[25] I. Barrai, C. Scapoli, C. Nesti, G. Poli, R. Gambari, and M. Beretta, "Codon usage and evolutionary rates of proteins," *J Theor Biol,* vol. 166, pp. 331-7, Feb 7 1994.

[26] N. Sueoka, "Directional mutation pressure and neutral molecular evolution," *Proc Natl Acad Sci U S A,* vol. 85, pp. 2653-7, Apr 1988.

[27] M. Archetti, "Selection on codon usage for error minimization at the protein level," *J Mol Evol,* vol. 59, pp. 400-15, Sep 2004.

[28] H. S. Najafabadi, H. Goodarzi, and N. Torabi, "Optimality of codon usage in Escherichia coli due to load minimization," *J Theor Biol,* vol. 237, pp. 203-9, Nov 21 2005.

[29] J. M. Comeron and M. Aguadé, "An evaluation of measures of synonymous codon usage bias," *J. Mol. Evol.,* vol. 47, pp. 268-274, 1998.

[30] Y. Nakamura, T. Gojobori, and T. Ikemura, "Codon usage tabulated from international DNA sequence databases: status for the year 2000," *Nucleic Acids Res,* vol. 28, p. 292, Jan 1 2000.

[31] P. M. Sharp and W. H. Li, "The codon Adaptation Index - a measure of directional synonymous codon usage bias, and its potential applications," *Nucleic Acids Res,* vol. 15, pp. 1281-95, 1987.

[32] T. Ikemura, "Correlation between the abundance of Escherichia coli transfer RNAs and the occurrence of the respective codons in its protein genes: a proposal for a synonymous codon choice that is optimal for the E. coli translational system.," *J Mol Biol.,* vol. 151, pp. 389-409, 1981.

[33] F. Wright, "The 'effective number of codons' used in a gene," *Gene,* vol. 87, pp. 23-29, 1990.

[34] R. H. Buckingham, "Codon context and protein synthesis: enhancements of the genetic code," *Biochimie,* vol. 76, pp. 351-4, 1994.

[35] A. Fedorov, S. Saxonov, and W. Gilbert, "Regularities of context-dependent codon bias in eukaryotic genes," *Nucleic Acids Res,* vol. 30, pp. 1192-7, Mar 1 2002.

[36] W. P. Tate, E. S. Poole, and S. A. Mannering, "Hidden infidelities of the translational stop signal," *Prog Nucleic Acid Res Mol Biol,* vol. 52, pp. 293-335, 1996.

[37] A. A. Shah, M. C. Giddings, J. B. Parvaz, R. F. Gesteland, J. F. Atkins, and I. P. Ivanov, "Computational identification of putative programmed translational frameshift sites," *Bioinformatics,* vol. 18, pp. 1046-53, 2002.

[38] E. J. Murgola, F. T. Pagel, and K. A. Hijazi, "Codon context effects in missense suppression," *J Mol Biol,* vol. 175, pp. 19-27, May 5 1984.

[39] S. Tork, I. Hatin, J. P. Rousset, and C. Fabret, "The major 5' determinant in stop codon read-through involves two adjacent adenines," *Nucleic Acids Res,* vol. 32, pp. 415-21, 2004.

[40] M. J. Telford and P. W. Holland, "Evolution of 28S ribosomal DNA in chaetognaths: duplicate genes and molecular phylogeny," *J Mol Evol,* vol. 44, pp. 135-44, Feb 1997.

[41] G. R. Cochrane and M. Y. Galperin, "The 2010 Nucleic Acids Research Database Issue and online Database Collection: a community of data resources," *Nucleic Acids Res,* vol. 38, pp. D1-4, Jan 2009.

[42] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, "Quantitative monitoring of gene expression patterns with a complementary DNA microarray," *Science,* vol. 270, pp. 467-70, Oct 20 1995.

[43] M. Y. Galperin, "The Molecular Biology Database Collection: 2008 update," *Nucleic Acids Res,* vol. 36, pp. D2-4, Jan 2008.

[44] Y. Tateno, "[International collaboration among DDBJ, EMBL Bank and GenBank]," *Tanpakushitsu Kakusan Koso,* vol. 53, pp. 182-9, Feb 2008.

[45] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and D. L. Wheeler, "GenBank," *Nucleic Acids Res,* vol. 36, pp. D25-30, Jan 2008.

[46] T. Kulikova, R. Akhtar, P. Aldebert, N. Althorpe, M. Andersson, A. Baldwin, K. Bates, S. Bhattacharyya, L. Bower, P. Browne, M. Castro, G. Cochrane, K. Duggan, R. Eberhardt, N. Faruque, G. Hoad, C. Kanz, C. Lee, R. Leinonen, Q. Lin, V. Lombard, R. Lopez, D. Lorenc, H. McWilliam, G. Mukherjee, F. Nardone, M. P. Pastor, S. Plaister, S. Sobhany, P. Stoehr, R. Vaughan, D. Wu, W. Zhu, and R. Apweiler, "EMBL Nucleotide Sequence Database in 2006," *Nucleic Acids Res,* vol. 35, pp. D16-20, Jan 2007.

[47] H. Sugawara, O. Ogasawara, K. Okubo, T. Gojobori, and Y. Tateno, "DDBJ with new system and face," *Nucleic Acids Res,* vol. 36, pp. D22-4, Jan 2008.

[48] M. Kanehisa and S. Goto, "KEGG: kyoto encyclopedia of genes and genomes," *Nucleic Acids Res,* vol. 28, pp. 27-30, Jan 1 2000.

[49] K. F. Aoki and M. Kanehisa, "Using the KEGG database resource," *Curr Protoc Bioinformatics,* vol. Chapter 1, p. Unit 1 12, Oct 2005.

[50] P. Flicek, B. L. Aken, K. Beal, B. Ballester, M. Caccamo, Y. Chen, L. Clarke, G. Coates, F. Cunningham, T. Cutts, T. Down, S. C. Dyer, T. Eyre, S. Fitzgerald, J. Fernandez-Banet, S. Graf, S. Haider, M. Hammond, R. Holland, K. L. Howe, K. Howe, N. Johnson, A. Jenkinson, A. Kahari, D. Keefe, F. Kokocinski, E. Kulesha, D. Lawson, I. Longden, K. Megy, P. Meidl, B. Overduin, A. Parker, B. Pritchard, A. Prlic, S. Rice, D. Rios, M. Schuster, I. Sealy, G. Slater, D. Smedley, G. Spudich, S. Trevanion, A. J. Vilella, J. Vogel, S. White, M. Wood, E. Birney, T. Cox, V. Curwen, R. Durbin, X. M. Fernandez-Suarez, J. Herrero, T. J. Hubbard, A. Kasprzyk, G. Proctor, J. Smith, A. Ureta-Vidal, and S. Searle, "Ensembl 2008," *Nucleic Acids Res,* vol. 36, pp. D707-14, Jan 2008.

[51] K. E. Rudd, "EcoGene: a genome sequence database for Escherichia coli K-12," *Nucleic Acids Res,* vol. 28, pp. 60-4, Jan 1 2000.

[52] M. B. Arnaud, M. C. Costanzo, M. S. Skrzypek, G. Binkley, C. Lane, S. R. Miyasato, and G. Sherlock, "The Candida Genome Database (CGD), a community resource for Candida albicans gene and protein information," *Nucleic Acids Res,* vol. 33, pp. D358-63, Jan 1 2005.

[53] J. M. Cherry, C. Adler, C. Ball, S. A. Chervitz, S. S. Dwight, E. T. Hester, Y. Jia, G. Juvik, T. Roe, M. Schroeder, S. Weng, and D. Botstein, "SGD: Saccharomyces Genome Database," *Nucleic Acids Res,* vol. 26, pp. 73-9, Jan 1 1998.

[54] C. H. Wu, R. Apweiler, A. Bairoch, D. A. Natale, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M. J. Martin, R. Mazumder, C. O'Donovan, N. Redaschi, and B. Suzek, "The Universal Protein Resource (UniProt): an expanding universe of protein information," *Nucleic Acids Res,* vol. 34, pp. D187-91, Jan 1 2006.

[55] A. Bairoch and B. Boeckmann, "The SWISS-PROT protein sequence data bank," *Nucleic Acids Res,* vol. 19 Suppl, pp. 2247-9, Apr 25 1991.

[56] W. C. Barker, J. S. Garavelli, H. Huang, P. B. McGarvey, B. C. Orcutt, G. Y. Srinivasarao, C. Xiao, L. S. Yeh, R. S. Ledley, J. F. Janda, F. Pfeiffer, H. W.

Mewes, A. Tsugita, and C. Wu, "The protein information resource (PIR)," *Nucleic Acids Res,* vol. 28, pp. 41-4, Jan 1 2000.

[57] P. Puigbo, A. Romeu, and S. Garcia-Vallve, "HEG-DB: a database of predicted highly expressed genes in prokaryotic complete genomes under translational selection," *Nucleic Acids Res,* vol. 36, pp. D524-7, Jan 2008.

[58] D. Gilbert, "Sequence file format conversion with command-line readseq," *Curr Protoc Bioinformatics,* vol. Appendix 1, p. Appendix 1E, Feb 2003.

[59] D. D. Womble, "GCG: The Wisconsin Package of sequence analysis programs," *Methods Mol Biol,* vol. 132, pp. 3-22, 2000.

[60] J. D. Retief, "Phylogenetic analysis using PHYLIP," *Methods Mol Biol,* vol. 132, pp. 243-58, 2000.

[61] A. Lim and L. Zhang, "WebPHYLIP: a web interface to PHYLIP," *Bioinformatics,* vol. 15, pp. 1068-9, Dec 1999.

[62] J. F. Peden, "Analysis of Codon Usage," in *Department of Genetics* Nottingham, UK: University of Nottingham, 1999, p. 214.

[63] F. Supek and K. Vlahovicek, "INCA: synonymous codon usage analysis and clustering by means of self-organizing map," *Bioinformatics,* vol. 20, pp. 2329-30, Sep 22 2004.

[64] J. E. Stajich and E. Birney, "The Bioperl project: motivation and usage," *ACM SIGBIO Newsletter,* vol. 20, pp. 13-14, 2000

[65] B. Chapman and J. Chang, "Biopython: Python tools for computational biology," *ACM SIGBIO Newsletter,* vol. 20, pp. 15-19, 2000

[66] "BioRuby."

[67] M. Pocock, T. Down, and T. Hubbard, "BioJava: open source components for bioinformatics," *ACM SIGBIO Newsletter,* vol. 20, pp. 10-12, 2000

[68] "NCBI C++ Toolkit."

[69] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs," *Nucleic Acids Res,* vol. 25, pp. 3389-402, Sep 1 1997.

[70] W. R. Pearson and D. J. Lipman, "Improved tools for biological sequence comparison," *Proc Natl Acad Sci U S A,* vol. 85, pp. 2444-8, Apr 1988.

[71] W. R. Pearson, "Flexible sequence similarity searching with the FASTA3 program package," *Methods Mol Biol,* vol. 132, pp. 185-219, 2000.

[72] N. Harte, V. Silventoinen, E. Quevillon, S. Robinson, K. Kallio, X. Fustero, P. Patel, P. Jokinen, and R. Lopez, "Public web-based services from the European Bioinformatics Institute," *Nucleic Acids Res,* vol. 32, pp. W3-9, Jul 1 2004.

[73] R. Lopez, V. Silventoinen, S. Robinson, A. Kibria, and W. Gish, "WU-Blast2 server at the European Bioinformatics Institute," *Nucleic Acids Res,* vol. 31, pp. 3795-8, Jul 1 2003.

[74] S. S. Sturrock and J. F. Collins, *MPsrch V1.3 User Guide.* Edinburgh: University of Edinburgh, 1993.

[75] Z. Ning, A. J. Cox, and J. C. Mullikin, "SSAHA: a fast search method for large DNA databases," *Genome Res,* vol. 11, pp. 1725-9, Oct 2001.

[76] W. J. Kent, "BLAT--the BLAST-like alignment tool," *Genome Res,* vol. 12, pp. 656-64, Apr 2002.

[77] L. Holm and C. Sander, "New structure--novel fold?," *Structure,* vol. 5, pp. 165-71, Feb 15 1997.

[78] R. L. Tatusov, S. F. Altschul, and E. V. Koonin, "Detection of conserved segments in proteins: iterative scanning of sequence databases with alignment blocks," *Proc Natl Acad Sci U S A,* vol. 91, pp. 12091-5, Dec 6 1994.

[79] G. J. Barton, "Computer speed and sequence comparison," *Science,* vol. 257, pp. 1609-10, Sep 18 1992.

[80] "National Center for Biotechnology Information (NCBI) ToolBox."

[81] T. A. Hall, "BioEdit: a user-friendly biological sequence alignment editor and analysis program for Windows 95/98/NT," *Nucl. Acids. Symp.,* vol. 41, pp. 95-98, 2007.

[82] CLC_bio, "CLC RNA Workbench ".

[83] J. Archuleta, W. C. Feng, and E. Tilevich, "A pluggable framework for parallel pairwise sequence search," *Conf Proc IEEE Eng Med Biol Soc,* vol. 2007, pp. 127-30, 2007.

[84] J. D. Thompson, D. G. Higgins, and T. J. Gibson, "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice," *Nucleic Acids Res,* vol. 22, pp. 4673-80, Nov 11 1994.

[85] K. B. Li, "ClustalW-MPI: ClustalW analysis using distributed and parallel computing," *Bioinformatics,* vol. 19, pp. 1585-6, Aug 12 2003.

[86] T. Hall, "BioEdit," 2007.

[87] K. B. Nicholas, H. B. J. Nicholas, and D. W. Deerfield, "GeneDoc: Analysis and Visualization of Genetic Variation," *EmbNew,* vol. 4, p. 14, 1997

[88] J. M. Rouillard, W. Lee, G. Truan, X. Gao, X. Zhou, and E. Gulari, "Gene2Oligo: oligonucleotide design for in vitro gene synthesis," *Nucleic Acids Res,* vol. 32, pp. W176-80, Jul 1 2004.

[89] G. Wu, N. Bashir-Bello, and S. J. Freeland, "The Synthetic Gene Designer: a flexible web platform to explore sequence manipulation for heterologous expression," *Protein Expr Purif,* vol. 47, pp. 441-5, Jun 2006.

[90] G. Lithwick and H. Margalit, "Hierarchy of sequence-dependent features associated with prokaryotic translation," *Genome Res,* vol. 13, pp. 2665-73, Dec 2003.

[91] A. Pingoud, J. Alves, and R. Geiger, "Chapter 8: Restriction Enzymes," in *Enzymes of Molecular Biology, Methods of Molecular Biology*, 16 ed, N. H. Press, Ed. Totowa, 1993, pp. 107-200.

[92] W. Arber and S. Linn, "DNA modification and restriction," *Annu Rev Biochem,* vol. 38, pp. 467-500, 1969.

[93] R. J. Roberts, T. Vincze, J. Posfai, and D. Macelis, "REBASE--enzymes and genes for DNA restriction and modification," *Nucleic Acids Res,* vol. 35, pp. D269-70, Jan 2007.

[94] H. Jin, Q. Zhao, E. I. Gonzalez de Valdivia, D. H. Ardell, M. Stenstrom, and L. A. Isaksson, "Influences on gene expression in vivo by a Shine-Dalgarno sequence," *Mol Microbiol,* vol. 60, pp. 480-92, Apr 2006.

[95] A. Villalobos, J. E. Ness, C. Gustafsson, J. Minshull, and S. Govindarajan, "Gene Designer: a synthetic biology tool for constructing artificial DNA segments," *BMC Bioinformatics,* vol. 7, p. 285, 2006.

[96] A. Antoun, M. Y. Pavlov, M. Lovmar, and M. Ehrenberg, "How initiation factors tune the rate of initiation of protein synthesis in bacteria," *EMBO J,* vol. 25, pp. 2539-50, Jun 7 2006.

[97] B. S. Laursen, H. P. Sorensen, K. K. Mortensen, and H. U. Sperling-Petersen, "Initiation of protein synthesis in bacteria," *Microbiol Mol Biol Rev,* vol. 69, pp. 101-23, Mar 2005.

[98] A. C. Chang, H. A. Erlich, R. P. Gunsalus, J. H. Nunberg, R. J. Kaufman, R. T. Schimke, and S. N. Cohen, "Initiation of protein synthesis in bacteria at a translational start codon of mamalian cDNA: effects of the preceding nucleotide sequence," *Proc Natl Acad Sci U S A,* vol. 77, pp. 1442-6, Mar 1980.

[99] C. M. Stenstrom, E. Holmgren, and L. A. Isaksson, "Cooperative effects by the initiation codon and its flanking regions on translation initiation," *Gene,* vol. 273, pp. 259-65, Aug 8 2001.

[100] S. Takyar, R. P. Hickerson, and H. F. Noller, "mRNA helicase activity of the ribosome," *Cell,* vol. 120, pp. 49-58, Jan 14 2005.

[101] D. M. Hoover and J. Lubkowski, "DNAWorks: an automated method for designing oligonucleotides for PCR-based gene synthesis," *Nucleic Acids Res,* vol. 30, p. e43, May 15 2002.

[102] D. Bercovich, Z. Regev, T. Ratz, A. Luder, Y. Plotsky, and Y. Gruenbaum, "Quantitative ratio of primer pairs and annealing temperature affecting PCR products in duplex amplification," *Biotechniques,* vol. 27, pp. 762-4, 766-8, 770, Oct 1999.

[103] P. Puigbo, E. Guzman, A. Romeu, and S. Garcia-Vallve, "OPTIMIZER: a web server for optimizing the codon usage of DNA sequences," *Nucleic Acids Res,* vol. 35, pp. W126-31, Jul 2007.

[104] S. Jayaraj, R. Reid, and D. V. Santi, "GeMS: an advanced software package for designing synthetic genes," *Nucleic Acids Res,* vol. 33, pp. 3011-6, 2005.

[105] T. M. Lowe and S. R. Eddy, "tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence," *Nucleic Acids Res,* vol. 25, pp. 955-64, Mar 1 1997.

[106] B. Everitt, *The Analysis of Contingency Tables*: Chapman & Hall/CRC, 1992.

[107] W. Mendenhall, R. J. Beaver, and B. M. Beaver, *Introduction To Probability And Statistics*: Thomson Brooks/Cole, 2005.

[108] D. J. Sheskin, *Parametric and nonparametric statistical procedures*: Chapman & Hall, 2000.

[109] V. M. Kruglov, "Complete convergence of the Pearson statistics " *Mathematical Notes,* vol. 66, pp. 515-519, 1999.

[110] S. J. Haberman, "The analysis of residuals in cross-classified tables," *Biometrics,* vol. 29, pp. 205-220, 1973.

[111] O. G. Berg and P. J. Silva, "Codon bias in Escherichia coli: the influence of codon context on mutation and selection," *Nucleic Acids Res,* vol. 25, pp. 1397-404, Apr 1 1997.

[112] P. J. Avery and D. A. Henderson, "Fitting Markov chain models to discrete state series such as DNA sequences," *Applied Statistics,* vol. 48, pp. 53-61, 1999.

[113] A. Fedorov, S. Saxonov, and W. Gilbert, "Regularities of context-dependent codon bias in eukaryotic genes," *Nucleic Acids Res,* vol. 30, pp. 1192-1197, 2002.

[114] B. S. Everitt, *Cluster Analysis*, 3 ed., 1998.

[115] E. Reis, *Estatística multivariada aplicada*. Porto, 2001.

[116] J. A. Hartigan, *Clustering Algorithms*, 1975.

[117] R. C. Tryon and D. E. Bailey, *Cluster analysis*. New York, 1970.

[118] M. Hoon, S. Imoto, and S. Miyano, "The C Clustering Library," 1.43 ed, 2008.

[119] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proc Natl Acad Sci U S A,* vol. 95, pp. 14863-8, Dec 8 1998.

[120] S. Datta, "Comparisons and validation of statistical clustering techniques for microarray gene expression data," *Bioinformatics,* vol. 19, pp. 459-66, Mar 1 2003.

[121] Y. Cheng and G. M. Church, "Biclustering of expression data," *Proc Int Conf Intell Syst Mol Biol,* vol. 8, pp. 93-103, 2000.

[122] S. C. Madeira and A. L. Oliveira, "Biclustering algorithms for biological data analysis: a survey," *IEEE/ACM Trans Comput Biol Bioinform,* vol. 1, pp. 24-45, Jan-Mar 2004.

[123] D. J. Reiss, N. S. Baliga, and R. Bonneau, "Integrated biclustering of heterogeneous genome-wide datasets for the inference of global regulatory networks," *BMC Bioinformatics,* vol. 7, p. 280, 2006.

[124] A. Prelic, S. Bleuler, P. Zimmermann, A. Wille, P. Buhlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler, "A systematic comparison and evaluation of biclustering methods for gene expression data," *Bioinformatics,* vol. 22, pp. 1122-9, May 1 2006.

[125] S. C. Madeira and A. L. Oliveira, "An Efficient Biclustering Algorithm for finding Genes with Similar Patterns in Time-Series Expression Data," *Series in Advances in Bioinformatics and Computational Biology,* pp. 67-80, 2007.

[126] J. Ihmels, S. Bergmann, and N. Barkai, "Defining transcription modules using large-scale gene expression data.," *Bioinformatic,* vol. 20, pp. 1993-2003, 2004.

[127] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini, "Discovering local structure in gene expression data: the order-preserving submatrix problem," *J Comput Biol,* vol. 10, pp. 373-84, 2003.

[128] T. M. Murali and S. Kasif, "Extracting conserved gene expression motifs from gene expression data," *Pac Symp Biocomput,* pp. 77-88, 2003.

[129] S. Barkow, S. Bleuler, A. Prelic, P. Zimmermann, and E. Zitzler, "BicAT: a biclustering analysis toolbox," *Bioinformatics,* vol. 22, pp. 1282–1283, 2006.

[130] J. Ihmels, G. Friedlander, S. Bergmann, O. Sarig, Y. Ziv, and N. Barkai, "Revealing modular organization in the yeast transcriptional network," *Nat Genet,* vol. 31, pp. 370-7, Aug 2002.

[131] X. Liu and L. Wang, "Computing the maximum similarity bi-clusters of gene expression data," *Bioinformatics,* vol. 23, pp. 50-6, Jan 1 2007.

[132] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *J Mol Biol,* vol. 215, pp. 403-10, Oct 5 1990.

[133] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *J Mol Biol,* vol. 48, pp. 443-453, 1970.

[134] D. S. Hirschberg, "A linear space algorithm for computing maximal common subsequences," *Comm. A.C.M.,* vol. 18, pp. 341-343, 1975.

[135] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *J Mol Biol,* vol. 147, pp. 195-197, 1981.

[136] A. J. Ropelewski, H. B. Nicholas, Jr., and D. W. Deerfield, 2nd, "Mathematically complete nucleotide and protein sequence searching using Ssearch," *Curr Protoc Bioinformatics,* vol. Chapter 3, p. Unit3 10, Feb 2004.

[137] D. F. Feng and R. F. Doolittle, "Progressive sequence alignment as a prerequisite to correct phylogenetic trees," *J Mol Evol,* vol. 25, pp. 351--360, 1987.

[138] R. C. Edgar and S. Batzoglou, "Multiple sequence alignment," *Curr Opin Struct Biol,* vol. 16, pp. 368-73, Jun 2006.

[139] C. Notredame, D. G. Higgins, and J. Heringa, "T-coffee: A novel method for fast and accurate multiple sequence alignment," *J Mol Biol,* vol. 302, pp. 205-217, 2000.

[140] R. C. Edgar, "MUSCLE: multiple sequence alignment with high accuracy and high throughput," *Nucleic Acids Res,* vol. 32, pp. 1792-7, 2004.

[141] J. Hein, "An algorithm for statistical alignment of sequences related by a binary tree," in *Pacific Symposium on Biocomputing*, 2001, p. 179.

[142] J. Hein, C. Wiuf, B. Knudsen, M. B. Møller, and G. Wibling, "Statistical alignment: computational properties, homology testing and goodness-of-fit.," *J Mol Biol,* vol. 302, pp. 265-279, 2000.

[143] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*, 1st edition ed. Massachusetts: Addison-Wesley, 1998.

[144] G. Moura, M. Pinheiro, R. Silva, I. Miranda, V. Afreixo, G. Dias, and e. al., "Comparative context analysis of codon pairs on an ORFeome scale," *Genome Biol.,* vol. 6, p. R28, 2005.

[145] M. Pinheiro, V. Afreixo, G. Moura, A. Freitas, M. A. Santos, and J. L. Oliveira, "Statistical, computational and visualization methodologies to unveil gene primary structure features," *Methods of Information in Medicine,* vol. 45, pp. 163-168, 2006.

[146] G. Moura, M. Pinheiro, J. Arrais, L. Carreto, A. Freitas, J. L. Oliveira, and M. A. S. Santos, "Large scale comparative codon-pair context analysis unveils general rules governing evolution of ORFeomes in the three domains of life.," 2007.

[147] M. Pinheiro, J. O. Oliveira, G. Moura, and M. Santos, "Studying the evolution of codon context in conserved gene sequences " *International Conference on Biocomputation, Bioinformatics, and Biomedical Technologies,* pp. 159-163, 2008.

[148] A. V. Freitas, M. Pinheiro, V. Afreixo, J. Duarte, J. L. Oliveira, G. Moura, and M. Santos, "A median-based Iterative Signature Algorithm," in *IASC 07 - Statistics for Data Mining, Learning and Knowledge Extraction*, University of Aveiro, Portugal, 2007.

[149] C. H. Freudenreich, S. M. Kantrow, and V. A. Zakian, "Expansion and length-dependent fragility of CTG repeats in yeast," *Science,* vol. 279, pp. 853-6, Feb 6 1998.

[150] S. E. Massey, G. Moura, P. Beltrao, R. Almeida, J. R. Garey, M. F. Tuite, and M. A. S. Santos, "Comparative evolutionary genomics unveils the molecular mechanism of reassignment of the CTG codon in Candida," *Genome Res,* vol. 13, pp. 544-557, 2003.

[151] N. Sueoka, "Translation-coupled violation of Parity Rule 2 in human genes is not the cause of heterogeneity of the DNA G+C content of third codon position," *Gene,* vol. 238, pp. 53-8, Sep 30 1999.

[152] M. Koyuturk, W. Szpankowski, and M. Gama, "Biclustering gene-features matrices for statistically significant dense patterns," *Proceeding of the 2004 IEEE Computational System Bioinformatics Conference,* pp. 480-484, 2004.

[153] A. V. Freitas, V. Afreixo, M. Pinheiro, J. L. Oliveira, G. Moura, and M. Santos, "Improving the performance of the Iterative Signature Algorithm for the identification of relevant patterns," *Statistical analysis and data mining,* Submitted 30-07-2009 2009.

[154] R. H. Buckingham, "Codon context," *Experientia,* vol. 46, pp. 1126-33, Dec 1 1990.

[155] J. Duan and M. A. Antezana, "Mammalian mutation pressure, synonymous codon choice, and mRNA degradation," *J. Mol. Evol.,* vol. 57, pp. 649-701, 2003.

[156] S. W. Chan, I. R. Henderson, and S. E. Jacobsen, "Gardening the genome: DNA methylation in Arabidopsis thaliana," *Nat Rev Genet,* vol. 6, pp. 351-60, May 2005.

[157] M. Widmann, M. Clairo, J. Dippon, and J. Pleiss, "Analysis of the distribution of functionally relevant rare codons," *BMC Genomics,* vol. 9, p. 207, 2008.

[158] S. Henikoff and J. G. Henikoff, "Amino acid substitution matrices from protein blocks.," *Proc Natl Acad Sci U S A,* vol. 89, pp. 10915-10919, 1992.

[159] M. O. Dayhoff and R. M. Schwartz, *Atlas of Protein Sequence and Structure* vol. 3. Washington D.C.: Nat. Biomed. Res. Found., 1978.

[160] S. R. Eddy, "Where did the BLOSUM62 alignment score matrix come from?," *Nat Biotechnol,* vol. 22, pp. 1035-1036, 2004.

[161] C. D. Livingstone and G. J. Barton, "Protein sequence alignments: a strategy for the hierarchical analysis of residue conservation," *Comput Appl Biosci,* vol. 9, pp. 745-56, Dec 1993.