BIOINFORMÀTICA:

ADD-IN D'ANÀLISI I MANIPULACIÓ DE SEQÜÈNCIES PER MICROSOFT WORD

Memòria del Projecte Fi de Carrera d'Enginyeria en Informàtica realitzat per Daniel Muñoz Fernández i dirigit per Ivan Erill Sagalés

Bellaterra, 1 de Febrer de 2008

SUMARI

ÍNDEX DE FIGURES

ÍNDEX D'EXEMPLES

1 RESUM

L'anàlisi i manipulació de seqüències genètiques requereix de certes eines d'edició pròpies d'aquestes, com poden ser la traducció de seqüències de DNA a proteïna, o l'obtenció de seqüències complementàries de DNA. Tot i que existeixen alguns programes destinats a la manipulació de seqüències, el cert és que en bioinformàtica, com en d'altres camps, l'editor de text més abastament utilitzat és Microsoft Word, amb les conseqüents mancances a l'hora de manipular seqüències genètiques i la necessitat de canviar constantment de programa i format per treballar amb elles.

L'objectiu del projecte consisteix en el desenvolupament d'un *add-in* d'anàlisi i manipulació de seqüències, senzill i de fàcil ús, integrable en l'entorn Microsoft Word per permetre la manipulació de seqüències genètiques directament des de Microsoft Word, estalviant temps i complicacions a l'usuari final.

2 INTRODUCCIÓ

2.1 BIOINFORMÀTICA

El terme Bioinformàtica, contracció de l'original francès (*biologie* + *information* + *automatique*), és probablement un dels únics triomfs llatins recents a l'hora d'encunyar nous termes i un dels camps afins a la informàtica amb un índex de creixement més alt en els darrers anys.

L'origen de la Bioinformàtica com a camp se sol situar a finals dels anys 1990, seguint l'estela de la consecució del projecte Genoma Humà (HGP) per part d'un consorci internacional (Lander, Linton et al. 2001). Davant la massiva quantitat de dades d'origen biològic generades arrel d'aquest projecte (el genoma humà consta de 3.000.000.000 bases nucleiques de seqüència de DNA), el problema de com tractar-les, desxifrar-les i interpretar-les va generar la necessitat de crear un camp a cavall entre la informàtica i la biologia amb l'objectiu d'automatitzar el processament de dades biològiques.

La Bioinformàtica, coneguda també com Biologia Computacional i dividida clàssicament en Genòmica (estudi del genoma[1]) i Proteòmica (estudi del proteoma[2]), es pot definir de manera laxa com l'ús d'algorismes i ordinadors per facilitar la comprensió i el modelatge de dades o problemes biològics. En conseqüència, per comprendre la problemàtica abordada per la Bioinformàtica es fa necessària una breu introducció als conceptes de DNA, RNA, proteïna, seqüències genètiques i expressió genètica.

2.2 CONCEPTES BÀSICS DE BIOLOGIA MOLECULAR

La biologia molecular estudia els fenòmens de procés d'informació en els éssers vius en el seu nivell més bàsic: el molecular. Així, la biologia molecular té com principals fonts d'estudi els elements bàsics de la vida a nivell molecular, que són les molècules de DNA, RNA i les proteïnes.

[1] Genoma: seqüència completa de la molècula de DNA que defineix un ésser viu. Al genoma existeixen multitud d'elements funcionals, d'entre els quals en destaca l'estructura familiarment coneguda com a "gen".
[2] Proteoma: conjunt complet de les proteïnes expressades per un ésser viu durant la seva existència.

2.2.1 DNA

El DNA (àcid desoxiribonucleic) i el seu parent proper (el RNA) constitueixen els principals components del material genètic de la majoria d'organismes, sent el components químics primari dels cromosomes i el material amb el que els gens estan codificats i s'expressen.

La funció principal del DNA és mantenir la informació genètica necessària per crear un ésser viu idèntic (o gairebé) a aquell del que prové, fet que s'acompleix mitjançant l'expressió dels gens codificats al DNA, a través del codi genètic, donant lloc a proteïnes, de forma que podem interpretar el DNA com una espècie de recepta per crear les proteïnes.

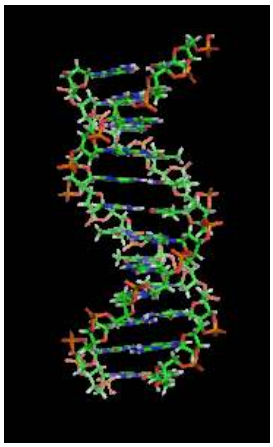


Figura 1 DNA

El DNA i el RNA estan formats per 4 tipus de nucleòtids, diferenciats per les seves bases nitrogenades dividides en dos grups: dues de puríniques (Adenina i Guanina) i dues de pirimidíniques (Citosina i Timina[3]).

La seva estructura és una parella de cadenes llargues de nucleòtids, complementades de la següent forma:

Adenina (A) amb Timina (T) → A - T

Citosina (C) amb Guanina (G) → C - G

2.2.2 PROTEÏNES

Les proteïnes són macromolècules (molècules de gran mida) formades per aminoàcids. Igual que en el cas del DNA i el RNA, les proteïnes són molècules polimèriques; és a dir, consisteixen en la concatenació seqüencial dels seus blocs bàsics, els aminoàcids.

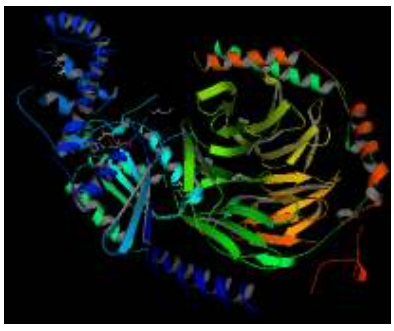


Figura 2 Model d'una proteïna en 3D.

[3] La principal diferència entre DNA i RNA és el canvi de Timina (T) al DNA per Uracil (U) al RNA. Tot i que hi ha d'altres diferències significatives entre ambdues molècules, en l'àmbit d'aquest projecte només es tindrà en compte aquesta.

Existeixen 20 tipus d'aminoàcids diferents a la natura que, en funció de certes propietats químiques i elèctriques (polaritat, hidrofòbia, etc.) donaran lloc a diferents tensions superficials sobre l'estructura de la proteïna. Aquest joc de tensions pot donar lloc a incomptables plegaments de la seqüència proteica original, oferint gran versatilitat en un procés de replegament seqüencial força complex i que té lloc en quatre etapes diferenciades.

2.3 EXPRESSIÓ GENÈTICA I TRADUCCIÓ

Com s'ha comentat, el DNA pot ser vist com una recepta per l'elaboració de proteïnes. L'estructura principal continguda en un genoma són els gens. Un gen és un element autocontingut amb capacitat d'expressar una proteïna i, perquè això sigui possible, és necessari que contingui certs elements:

Figura 3 El gen, l'element bàsic de l'expressió genètica, està compost per un promotor, un lloc d'acoblament del ribosoma (RBS), una regió codificant i un terminador.

Els elements bàsics del gen són:

- El promotor: una regió que permetrà (en condicions favorables) l'acoblament d'un holoenzim, la RNA-polimerasa, encarregat de la transcripció (passar el missatge de DNA a RNA.
- El *Ribosome Binding Site* (RBS, o regió d'acoblament del ribosoma): lloc on s'acoblarà, a una vegada transcrit a mRNA, el ribosoma. Al ribosoma hi té lloc la traducció: convertir el missatge de mRNA a aminoàcids i crear la seqüència de proteïna.
- L'*Open Reading Frame* (ORF, o regió codificant): el què hom entén tradicionalment com a gen. És la regió de codi que serà traduïda de mRNA a aminoàcids i, per tant, la que codifica la proteïna resultant.
- Terminador: la regió on acaba la transcripció (que no la traducció, que acaba abans, al final de l'ORF), promovent el desacoblament de la RNA-polimerasa.

El procés d'expressió genètica comença amb la *transcripció* del gen a RNA missatger (mRNA). En aquest procés, un enzim especial (la RNA-polimerasa) s'acobla al promotor del gen i en copia el missatge en una molècula de RNA. És en aquesta molècula de RNA on s'acobla llavors el ribosoma per dur a terme la *traducció* de la regió codificant (ORF) a proteïna, fent ús del RNA de transferència (tRNA) que, a la pràctica, implementa el que es coneix com el codi genètic.

First Position	Second Position of Codon: T	C	A	G	Third Position
T	TTT Phe [F]	TCT Ser [S]	TAT Tyr [Y]	TGT Cys [C]	T
	TTC Phe [F]	TCC Ser [S]	TAC Tyr [Y]	TGC Cys [C]	C
	TTA Leu [L]	TCA Ser [S]	TAA *Ter* [end]	TGA *Ter* [end]	A
	TTG Leu [L]	TCG Ser [S]	TAG *Ter* [end]	TGG Trp [W]	G
C	CTT Leu [L]	CCT Pro [P]	CAT His [H]	CGT Arg [R]	T
	CTC Leu [L]	CCC Pro [P]	CAC His [H]	CGC Arg [R]	C
	CTA Leu [L]	CCA Pro [P]	CAA Gln [Q]	CGA Arg [R]	A
	CTG Leu [L]	CCG Pro [P]	CAG Gln [Q]	CGG Arg [R]	G
A	ATT Ile [I]	ACT Thr [T]	AAT Asn [N]	AGT Ser [S]	T
	ATC Ile [I]	ACC Thr [T]	AAC Asn [N]	AGC Ser [S]	C
	ATA Ile [I]	ACA Thr [T]	AAA Lys [K]	AGA Arg [R]	A
	ATG Met [M]	ACG Thr [T]	AAG Lys [K]	AGG Arg [R]	G
G	GTT Val [V]	GCT Ala [A]	GAT Asp [D]	GGT Gly [G]	T
	GTC Val [V]	GCC Ala [A]	GAC Asp [D]	GGC Gly [G]	C
	GTA Val [V]	GCA Ala [A]	GAA Glu [E]	GGA Gly [G]	A
	GTG Val [V]	GCG Ala [A]	GAG Glu [E]	GGG Gly [G]	G

Figura 4 El codi genètic. Les files corresponen a la primera i tercera posició del codó, seguint l'ordre T→C→A→G; les columnes corresponen a la segona posició del codó. Així la casella (1,2,3) correspon a fila 1 (T), columna 2 (C) sub-fila 3 (A): TCA→Serina (S).

Durant la traducció, grups de 3 bases d'una cadena de DNA es tradueixen en un aminoàcid d'una proteïna. És a dir, un aminoàcid és produït a partir de la traducció de 3 nucleòtids. Així doncs, el codi genètic ofereix (4^3) 64 paraules codi, però a la natura només existeixen 20 aminoàcids. Aquest fet fa que el codi genètic sigui redundant, és a dir, a cada aminoàcid li correspon més d'un codó (conjunt de 3 bases de DNA), fet que li atorga força robustesa enfront de mutacions a la cadena original de DNA, però que dificulta enormement la traducció inversa a partir de la seqüència resultant de proteïna.

2.4 MANIPULACIÓ I EDICIÓ DE SEQÜÈNCIES

2.4.1 NECESSITAT

Les seqüències genètiques (entenent aquí tant seqüències de DNA, RNA o proteïna) constitueixen avui en dia una eina fonamental de treball en biologia i en camps afins.

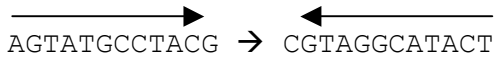
Existeix, per tant, una necessitat real de tractar i manipular aquests tipus de seqüències genètiques de forma habitual, a fi de dur a terme diferents transformacions i canvis a les seqüències, bé per tal d'emular diversos processos que tenen lloc a la natura (com l'obtenció de cadenes complementàries, la traducció de DNA a proteïna o la detecció de llocs i motius al DNA), o bé per implementar tècniques que permetin aplicar tractaments útils sobre seqüències (com traduir a DNA a partir de seqüències proteiques, alinear seqüències per extreure informació, etc.).

2.4.2 TIPUS DE FUNCIONALITATS HABITUALS

Existeixen molts tipus de processos i tractaments de seqüències de DNA, RNA i proteïna que poden ser d'interès per diferents usuaris. No obstant, el comú d'operacions sobre seqüències dut a terme per la majoria d'usuaris es com dividir en els següents grups funcionals:

- ***Conversió de formats*** (FASTA, Genbank, MSF, ...): Com en molts altres camps de la Informàtica, la proliferació de diferents formats (més o menys justificada) amb l'aparició de nous programes i algorismes és un dels maldecaps constants en Bioinformàtica. Així, la conversió entre diferents formats és una funcionalitat habitual en l'edició de seqüències genètiques.

- ***Edició*** (girar, complementar, ...): Aquestes funcionalitats consisteixen en fer transformacions a la seqüència. Un exemple en seria girar i complementar una seqüència de DNA de forma simultània, a fi d'observar com és la seqüència de DNA que es troba a la cadena complementària.



- ***Recerca*** (de llocs, paraules...): Per recerca entenem la localització de paraules, característiques o llocs específics al DNA prèviament definits. Per exemple, ens podria interessar localitzar al DNA regions que codifiquen proteïnes, motius sobre els que s'acoblen certes proteïnes, o regions del DNA amb una curvatura elevada.

- ***Traducció*** (de DNA a proteïna i viceversa): Per diferents motius, a l'usuari final li pot interessar simular la traducció de DNA a proteïna o la traducció de proteïna a DNA, també coneguda com traducció inversa.

AGTATGCCTACG ⇔ SMPT

- ***Alineament*** (d'una o vàries seqüències): El procés d'alineament és el d'establir homologies posicionals entre dues o més seqüències. És a dir, a partir de dues o més seqüències, busquem obtenir una relació d'identitat entre les diferents posicions de cadascuna, de forma que puguem identificar relacions funcionals o evolutives entre els les diferents seqüències genètiques[4]. Les seqüències alineades s'escriuen amb les lletres en columnes en les que s'insereixen espais perquè les zones amb idèntica o similar estructura quedin alineades. Existeixen dos tipus d'alineament (global i local) en funció de si pretenem alinear completament la longitud total de les seqüències (global) o si només busquem trobar-ne les regions de més homologia.

```
Global FTFTALILLAVAV
       F--TAL-LLA-AV

Local  FTFTALILL-AVAV
       --FTAL-LLAAV--
```

- ***Filogènia***: La filogènia estudia la generació d'arbres evolutius (i.e. que mostren les relacions evolutives entre varies espècies o entitats que es suposa que tenen una ascendència comú) a partir d'alineaments de seqüències, a fi d'extreure coneixement sobre les relacions evolutives entre diferents seqüències i, en conseqüència, entre els organismes als quals aquestes pertanyen.

[4] Degut a que estableix una hipòtesi d'homologia posicional entre seqüències, l'alineament és un procés crític en Bioinformàtica, ja que errors en l'alineament inicial poden dur a conclusions força errònies en anàlisis posteriors basats en l'alineament.

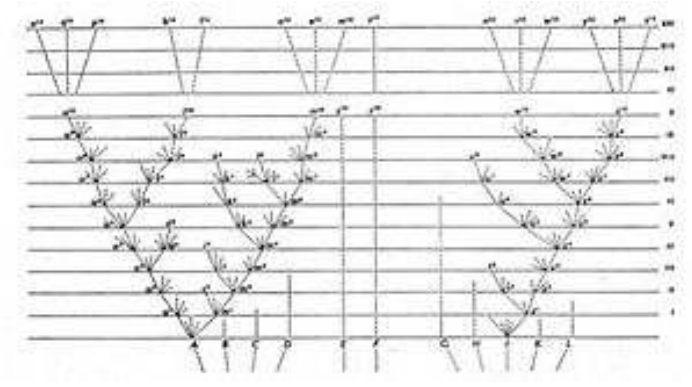


Figura 5 Diagrama dibuixat per Charles Darwin en L'Origen de les Espècies.

2.4.3 ESTAT DE L'ART

En l'actualitat existeixen força aplicacions destinades a biòlegs, bioinformàtics i a tota persona relacionada amb el món de la biologia que tracta amb seqüències genètiques. Aquestes aplicacions es poden catalogar en dos tipus, unes que són software independent, és a dir, programes instal·lables que realitzen unes funcions, i d'altres que són aplicacions web, és necessari disposar de connexió a Internet i treballar *online* des del navegador. El problema és que cap d'elles s'integra en una plataforma comuna de fàcil ús, com podria ser algun processador de textos existent.

En el cas de les aplicacions web solen ser aplicacions bàsiques de tractament de seqüències, distribuïdes en diferents *applets*. En conseqüència, el seu ús quan es treballa amb seqüències sobre un editor convencional implica haver de canviar d'eina per poder obtenir un resultat, que després haurem de traslladar de nou al nostre processador. Pel que fa a les plataformes *standalone*, aquestes solen ser aplicacions força complexes, amb moltes opcions i sovint nombrosos subprogrames, fet que en dificulta l'ús per tasques bàsiques i comuns. A més a més, el seu preu acostuma a ser força elevat i, de nou, impliquen la necessitat de canviar d'eina, si fem servir un processador de textos convencional, per obtenir un resultat que després altra vegada haurem de traslladar al nostre processador.

Alguns exemples que podem trobar al mercat són els següents:

- *DnaStar*:

  Consisteix en una suite integrada de set mòduls (subprogrames) que poden ser comprats en qualsevol combinació. Els diversos subprogrames estan encaminats a tasques com la manipulació bàsica de seqüències (recerca bàsica, traducció, complementació), la identificació de parells de *primers* per PCR o la manipulació d'alineaments.

  http://www.dnastar.com/

- *BioEdit*:

  BioEdit és un editor gratuït i molt complet d'alineaments de seqüències biològiques escrit per Windows 95/98/NT/2000/XP, però no integra cap tipus de funcionalitat bàsica per l'edició de seqüències.

  http://www.mbio.ncsu.edu/BioEdit/bioedit.html

- *Sequence manipulation suite*:

  La SMS és una col·lecció de programes en JavaScript per la generació, formatejament i anàlisi de cadenes curtes de DNA i proteïna, desenvolupat per la Universitat d'Alberta. La col·lecció és força complerta però difícil d'ús degut a la seva varietat i a la necessitat constant de copiar i enganxar seqüències sobre els *forms* JavaScript.

  http://www.bioinformatics.org/sms2/

- *Manipulate and Display a DNA Sequence*:

  Aquest web és un exemple clàssic de la funcionalitat distribuïda per la xarxa. Permet només l'edició bàsica de seqüències (*revers*, *complement*, *revers-complement*) i no té lligams amb d'altres webs o eines de manipulació més avançada o específica.

  http://www.vivo.colostate.edu/molkit/manip/

- *Leto 1.0*:

  Leto és un sistema potent de traducció inversa (veure *Rev.Translate*) de seqüències de proteïna a DNA, però de nou es tracta d'un sistema aïllat i, en aquest cas, que a més requereix de registre pel seu ús.

  http://www.entelechon.com/bioinformatics/backtranslation.php?lang=eng

- Detecció de motius

  Existeixen al web nombroses eines per la detecció i el descobriment de motius al DNA, basades en diferents tècniques. Aquí se'n citen només algunes:

  - *Meme*:

    Descobriment de motius per maximització de l'expectació a partir d'una col·lecció de seqüències susceptibles de contenir el/s motiu/s.

    http://meme.sdsc.edu/meme/

  - *Virtual FootPrint*:

    Suite de reconeixement de patrons simples o compostos de DNA.

    http://www.prodoric.de/vfp/

- *PredictRegulon*:

  Sistema de recerca de patrons en DNA basat en teoria de la informació.

  http://210.212.212.6/prindex.html

3 PLANTEJAMENT I OBJECTIUS

La finalitat d'aquest projecte és aconseguir integrar un *add-in* d'edició y manipulació de seqüències genètiques dins d'un processador de textos, a fi de facilitar les tasques bàsiques d'edició i manipulació de seqüències dels usuaris finals.

Per arribar a aquesta finalitat s'han plantejat uns objectius parcials a assolir:

- *Anàlisi de requeriments d'usuari, d'aplicació i de les funcions a implementar.*
  A quin tipus d'usuari va dirigit, cóm s'implementarà, de quines funcions disposarà,...

- *Desenvolupament d'una plataforma genèrica i escalable.*
  Que sigui modulable i de codi obert per facilitar ampliacions futures.

- *Implementació i test de l'eina creada.*
  Creació de l'eina i testejar els resultats obtinguts.

- *Distribució de l'eina resultat en departaments de biologia.*
  Fer arribar l'eina a usuaris finals per tal que sigui útil i no quedi en l'oblit.

4 ANÀLISI DE REQUERIMENTS

4.1 REQUERIMENTS D'USUARI

El perfil dels usuaris que utilitzaran l'eina a desenvolupar és el de biòleg que es dedica o treballa usualment al camp de la biologia molecular. Tot i això, qualsevol persona interessada en aquest àmbit pot ser un usuari potencial, ja que la facilitat d'ús de la que es pretén dotar a l'eina i les seves funcionalitats la converteixen en una eina d'ampli abast al món de la biologia.

A fi d'acotar els requeriments funcionals per part d'usuari que implicava la realització d'aquest projecte, es va elaborar un qüestionari[5] per determinar certs aspectes fonamentals de l'eina (*add-in*) proposada: la plataforma (processador de textos) sobre la que s'havia d'executar, la viabilitat i utilitat del projecte i les funcionalitats principals que calia cobrir per donar servei a la majoria d'usuaris. Així doncs, es va repartir el qüestionari entre una mostra de 35 biòlegs del Departament de Genètica i Microbiologia de la Facultat de Biociències de la UAB. Al qüestionari es demanava als enquestats quin era el processador de textos que utilitzaven, si els seria útil una eina de tractament i edició de seqüències integrada al seu processador de textos i quines funcionalitats considerarien útil que contingués.

Segons aquesta enquesta es va poder validar que el processador de textos més utilitzat entre la comunitat de biòlegs era el Microsoft Word, convertint aquesta plataforma en la plataforma d'elecció pel desenvolupament d'aquest projecte de forma definitiva.

A més, les funcionalitats més votades van ser les de filtratge i conversió entre diferents formats, els mètodes d'edició [com per exemple, girar i complementar seqüències, traducció (DNA→Proteïna) i traducció inversa (Proteïna→DNA)], la recerca de motius en seqüències de DNA, l'alineament de seqüències i la recerca automàtica de regions codificants (ORFs).

Es va analitzar també la mecànica de treball de diversos usuaris a l'hora de manipular seqüències en diferents programes d'edició, a fi de veure quin flux de procés els resultaria més convenient pel tractament de seqüències. La conclusió va ser que la majoria d'usuaris preferia un mètode sense *tags*, basat simplement en la selecció amb el ratolí d'una/es

[5] El qüestionari es pot trobar a l'annex d'aquesta memòria.

seqüència/es i la crida mitjançant menú (preferentment sobre comandes de teclat) d'una funció específica. A partir de consultes posteriors amb biòlegs, es va anotar també la possibilitat d'utilitzar el *clipboard* com a sistema d'entrada/sortida opcional del programa, a fi de poder acumular diferents operacions sobre una mateixa seqüència.

4.2 REQUERIMENTS FUNCIONALS

Determinada Microsoft Word (i en conseqüència Microsoft Office) com la plataforma d'elecció de cara a desenvolupar l'*add-in* d'edició i manipulació de seqüències proposat en aquest projecte, es va procedir a analitzar els diversos mètodes d'implementació possibles.

4.2.1 IMPLEMENTACIÓ AMB MACROS

Donat que Microsoft Word duu incorporat un intèrpret de Visual Basic (VBA) amb el que programar macros i amb prou versatilitat com per dissenyar funcions complexes, una primera possibilitat que es va considerar va ser la d'implementar el projecte com un conjunt de macros de Microsoft Word amb VBA. Tot i que aquesta opció tenia l'atractiu d'una fàcil modificació per part de tercers usuaris (sobretot biòlegs) per implementar noves funcionalitats (donada la facilitat de VBA i la seva integració directa en Microsoft Word), aquesta opció es va desestimar a causa d'una baixa facilitat d'instal·lació i distribució (és complicat automatitzar la instal·lació de macros en Microsoft Word com a paquet distribuïble), i d'una baixa velocitat, ja que les macros executades des de Microsoft Word sobre VBA són aplicacions interpretades i no compilades.

4.2.2 IMPLEMENTACIÓ AMB VS.NET I VSTO

Una altra possibilitat va ser utilitzar una eina de programació que integrava els controls propis de Microsoft Word dins d'una plataforma de programació com és Visual Studio .NET. Aquesta eina s'anomena Visual Studio Tools for Office (VSTO)[6].

[6] Visual Studio Tools for Office és la primera eina de desenvolupament per construir aplicacions per Microsoft Office. Conté llibreries integrades utilitzables des de la plataforma .NET que conformen la passarel·la directa al model de dades de les aplicacions de Microsoft Office.

Aquesta opció va ser l'escollida per les raons que van desestimar la creació d'una macro, és a dir, la facilitat d'instal·lació, ja que es crea un paquet autoinstal·lable, i la seva velocitat al ser una aplicació compilada.

Una vegada escollida es van obrir dos possibles camins a seguir, el llenguatge de programació en el que crear l'aplicació. Per una banda teníem C# i per l'altra VBasic. Entre aquestes opcions es va escollir VB pel fet que s'integra molt millor amb Microsoft Word, ja que el paquet de Microsoft Office disposa d'una eina d'edició de VBasic, a més de considerar-lo un llenguatge de programació més senzill a l'hora de que futurs usuaris de l'eina (principalment biòlegs) puguin fer modificacions o ampliacions, ja que considero aquest treball com una aplicació de codi obert.

4.2.3 ALTRES ALTERNATIVES D'IMPLEMENTACIÓ CONSIDERADES

A més d'aquestes dues opcions d'integració de l'eina en Microsoft Word, es van analitzar altres opcions per veure si realment no eren factibles, entre elles la creació d'una eina que treballes amb entrades i sortides al *clipboard*. Aquesta solució va ser descartada ja que dificultava la comprensió i utilització de l'eina pel fet de no poder visualitzar les entrades i sortides directament i haver-les de copiar sobre un editor de textos, el que implica tenir dues aplicacions diferents obertes, fet que xoca amb una de les premisses inicials del projecte, la realització d'una eina integrada per evitar la utilització de varies aplicacions a l'hora.

Una altra opció va ser la realització d'una eina externa amb un control de Microsoft Word integrat. Aquesta opció també es va descartar pel mateix que l'anterior, no integra i s'ha d'obrir una altra eina.

Per tant, finalment es va escollir la realització d'una aplicació compilada i instal·lable com a barra d'eines de Microsoft Word, realitzada amb l'eina VSTO.

4.3 REQUERIMENTS HARDWARE/SOFTWARE

Els requeriments hardware que es necessitarien per la utilització de l'aplicació desitjada serien els mateixos que per fer córrer Microsoft Word 2003 o superior, és a dir, si es pot instal·lar i utilitzar Microsoft Word 2003 o superior, es pot instal·lar i utilitzar aquesta eina.

Els requeriments software necessaris s'han de separar en dos blocs, els requeriments per a la implementació i creació de l'eina i els requeriments per a la utilització d'aquesta.

Els requeriments per a la implementació són el Microsoft Office 2003 o 2007, el Visual Studio .NET 2005, el VSTO 2005 (descarregable gratuïtament de la web de Microsoft) i les PIA o assemblats d'interoperabilitat primaris (també descarregables gratuïtament).

Per a la utilització de l'eina són gairebé els mateixos, amb la única diferència que no cal disposar del Visual Studio .NET, amb el Microsoft Office, el VSTO i les PIA és suficient. Per tant, es pot considerar que per a la utilització de l'eina no cal desemborsar cap quantitat de diners, ja que se suposa que l'usuari de l'eina ja serà usuari de Microsoft Office, que és l'únic que necessita que sigui de pagament.

4.4 REQUERIMENTS TEMPORALS

Després d'un anàlisi inicial, on es van estimar el número i tipus de funcionalitats escollides i les diverses plataformes de desenvolupament descrites anteriorment, es va considerar que, amb les funcionalitats escollides i amb la facilitat que aporta el VSTO per a la implementació de l'eina, el projecte plantejat era factible de ser realitzat en un semestre. Partint d'aquest supòsit, es va realitzar un esquema inicial de planificació temporal, tal i com es mostra a continuació mitjançant un diagrama de Gantt:

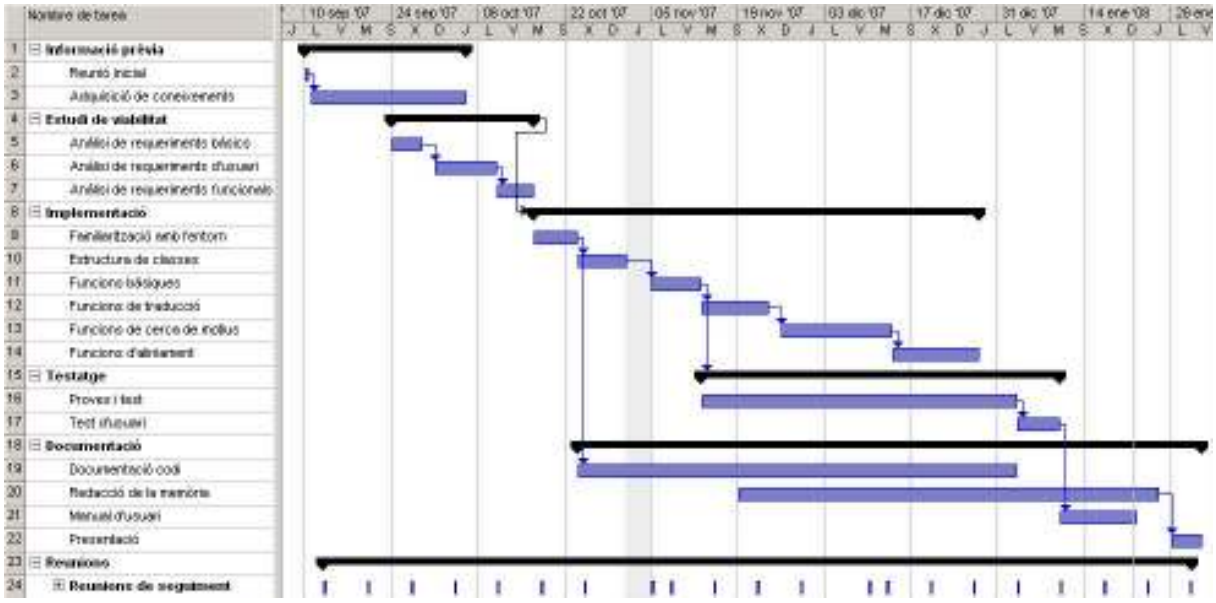


Figura 6 Diagrama de Gannt. Planificació temporal de les diferents parts del projecte.

A la figura anterior es poden observar els diferents mòduls en que es van separar les tasques a realitzar. La planificació comença lògicament amb la realització d'una reunió per discutir els termes del projecte i assolir una sèrie d'acords i finalitats. Immediatament, s'inicia un procés d'adquisició de coneixements en el camp escollit (la Bioinformàtica), imprescindibles tant per la comprensió detallada de diversos objectius del projecte, així com per la pròpia realització i la consecució del projecte.

Una vegada assolits uns coneixements mínims en l'àmbit del projecte, el següent pas consistiria en la realització d'un estudi de viabilitat, on s'estudiarien els requeriments de l'usuari (partint de les premisses inicials), els requeriments funcionals i els de hardware/software.

Una vegada decidit què es vol implementar, com implementar-ho i amb quines eines, es passaria a desenvolupar la part de programari, és a dir, la realització de l'estructura de classes i la implementació de les diferents agrupacions de funcionalitats, que comprendria la major part del desenvolupament del projecte i conclouria amb testos i proves de l'eina i les seves funcionalitats.

A més a més, i durant el transcurs de les diferents fases esmentades a dalt, s'aniria realitzant de forma palatina la documentació del projecte (memòria, manual d'usuari, ...). També es va preveure que, pel correcte funcionament del projecte, caldria dur a terme una sèrie de reunions setmanals amb el director de projecte a fi d'anar validant resultats i poder discutir funcionalitats i temes de desenvolupament.

5 IMPLEMENTACIÓ

Aquesta secció descriu el procés d'implementació del projecte a partir de l'anàlisi de requeriments dut a terme. El desenvolupament s'inicia amb una proposta d'estructura de classes bàsiques que conformaran l'esquelet de l'*add-in* a partir d'un anàlisi del flux i l'esquema de funcionament del programa. Tal i com s'explicarà als següents apartats, a partir d'aquesta estructura bàsica i dels diferents requeriments de funcionalitat, l'*add-in* es complementarà amb noves classes que permetin fer-ne el tractament adequat i proveir-ne les funcionalitats requerides.

5.1 ANÀLISI DE ESTRUCTURA DE CLASSES BÀSICA

De cara a fer un anàlisi de l'estructura de classes bàsica, primerament es va dur a terme un anàlisi de casos bàsic per veure quins elements havia de manipular l'*add-in* i quin tipus d'operacions bàsiques calia dur a terme.

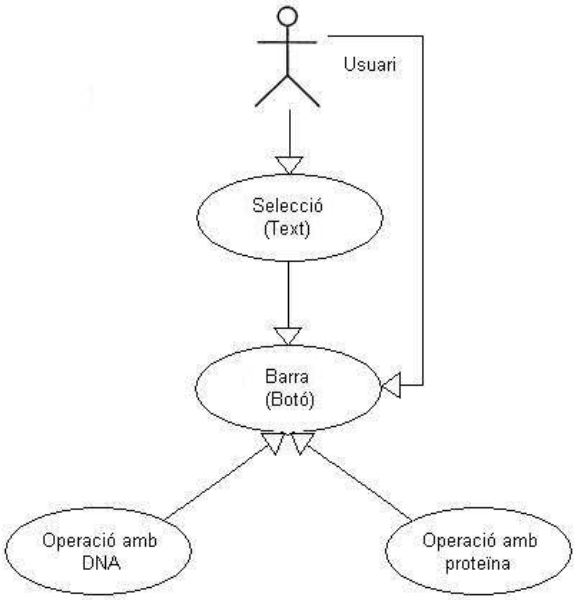
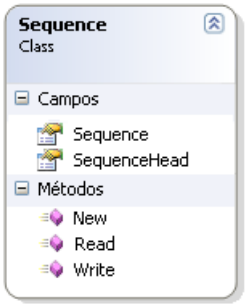


Figura 7 Diagrama de casos d'ús.

5.1.1 CLASSE *SEQUENCE* I DERIVADES

En vista de les funcionalitats triades durant el procés d'anàlisi de requeriments i d'una estimació futura de possibles funcionalitats addicionals basada en l'anàlisi de diferents eines al mercat (veure *Pàgina 16*), es va concloure que l'*add-in* sempre manipularia seqüències. En conseqüència, es va decidir que tenia lògica crear una classe seqüència (*Sequence*) capaç de centrar les operacions amb seqüències. D'altra banda, però, també es va fer palesa la

necessitat de tractar de forma diferent les seqüències de DNA i de proteïna, bé per oferir funcionalitats especials en una de les dues (e.g. traduir al DNA, traduir-invers a proteïna) o bé per modificar el tractament d'una funcionalitat (e.g. repertori de caràcters admesos al buscar patrons en una o altra). A resultes d'això, es va decidir derivar dues subclasses de *Sequence*, una per seqüències de DNA (*SequenceDNA*) i una altra per seqüències de proteïna (*SequenceProt*).



La classe *Sequence* conté dos atributs: *SequenceHead* ha estat implementat per contenir la capçalera de les seqüències en format FASTA i *Sequence* per contenir la resta de la seqüència. A més, la classe *Sequence* conté 3 mètodes: el constructor, un mètode de lectura i un d'escriptura.

Figura 8 Esquema gràfic bàsic de la classe *Sequence*.

Lògicament, els mètodes que siguin compartits per totes dues subclasses (com poden ser els mètodes d'entrada/sortida o alguna manipulació concreta de les seqüències, com veurem posteriorment) estaran continguts a la classe pare, mentre que els mètodes propis o especialitzats de cada tipus de seqüència es trobaran a les classes derivades.

5.1.2 CLASSE *COLLECTION*

Des d'un inici es va considerar el fet que l'usuari podia desitjar aplicar un tractament (e.g. traduir, buscar patrons...) a vàries seqüències simultàniament. Així doncs, tenint en compte la possibilitat d'introduir varies seqüències d'entrada alhora, es va decidir crear una classe col·lecció (*Collection*) que consistís en una classe que agrupés seqüències (tant de DNA com de proteïna). La introducció de la classe *Collection* per gestionar l'entrada de seqüències per part de l'usuari implicava que l'entrada sempre fos una col·lecció de seqüències. En el cas de desitjar treballar amb una única seqüència d'entrada, l'esquema de treball funcionaria igual, ja que només calia definir una col·lecció amb una única seqüència.

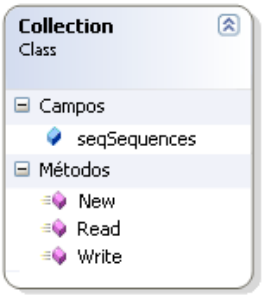


Figura 9 Esquema gràfic bàsic de la classe *Collection*.

La classe *Collection* conté un atributs: *seqSequences*, que ha estat implementat per contenir la llista de seqüències que pot contenir un objecte *Collection*. A més, la classe *Sequence* conté 3 mètodes: el constructor, un mètode de lectura i un d'escriptura.

Com es detallarà més endavant, l'ús de col·leccions de seqüències és força habitual en Bioinformàtica. Per exemple, quan es vol dur a terme un alineament de seqüències (veure *Pàgina 62*), l'entrada sempre serà un conjunt de seqüències no alineades. Dintre de les funcionalitats proposades en aquest projecte, es va observar que la definició de la classe *Collection* es podia aprofitar abastament en la implementació de l'alineament i en la recerca de motius o patrons en seqüències (veure *Pàgina 49*), on els motius s'acostumen a definir mitjançant col·leccions de seqüències.

5.1.3 CLASSES D'OPCIONS DE CONFIGURACIÓ

Per englobar tot el tema d'opcions configurables per l'usuari, es va decidir crear vàries classes. Una d'elles conté totes les opcions o paràmetres de configuració que l'usuari pot modificar. Aquesta classe es va anomenar *Configuration*.

Les opcions de *Configuration* són guardades en un arxiu XML, així que, per poder accedir fàcilment a elles, es va crear una altra classe (*XmlAcces*) que contenia dos mètodes per poder llegir i escriure a arxius XML. Una vegada creada, la classe *XmlAcces* es podia re-aprofitar per d'altres funcionalitats a l'hora de guardar informació diversa en arxius (en traducció (veure *Pàgina 44*) el codi genètic es guarda com a XML).

A part d'aquestes dues classes, perquè l'usuari pogués introduir i modificar les opcions de configuració es va crear una altra classe (*Options*), que contingués un formulari per facilitar la consulta i modificació de les opcions.

A continuació es mostra l'esquema inicial de classes descrit anteriorment:

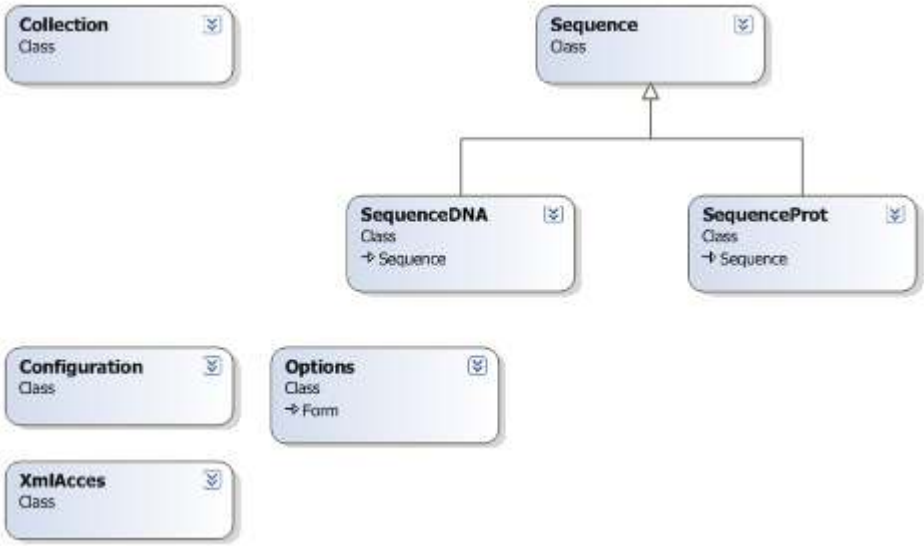


Figura 10 Esquema inicial de classes. Classes bàsiques de tractament de seqüències.

Tal i com ja s'ha comentat, aquest diagrama és l'esquelet bàsic de l'estructura de classes de l'*add-in*, i només s'hi mostren les classes principals. Per facilitar-ne la comprensió, la resta de classes que hi interaccionen s'aniran descrivint a mesura que se'n descrigui la funcionalitat en els propers apartats.

5.2 ANÀLISI DEL FLUX DEL PROGRAMA

En paral·lel al disseny de l'estructura bàsica de classes, i per ajudar a definir la pròpia estructura de classes i el funcionament general de l'*add-in*, es va fer un anàlisi del funcionament i del flux bàsic del programa, basat en el diagrama de casos mostrat anteriorment (*Figura 7*). A continuació es mostra el diagrama de flux obtingut a resultes d'aquest anàlisi:

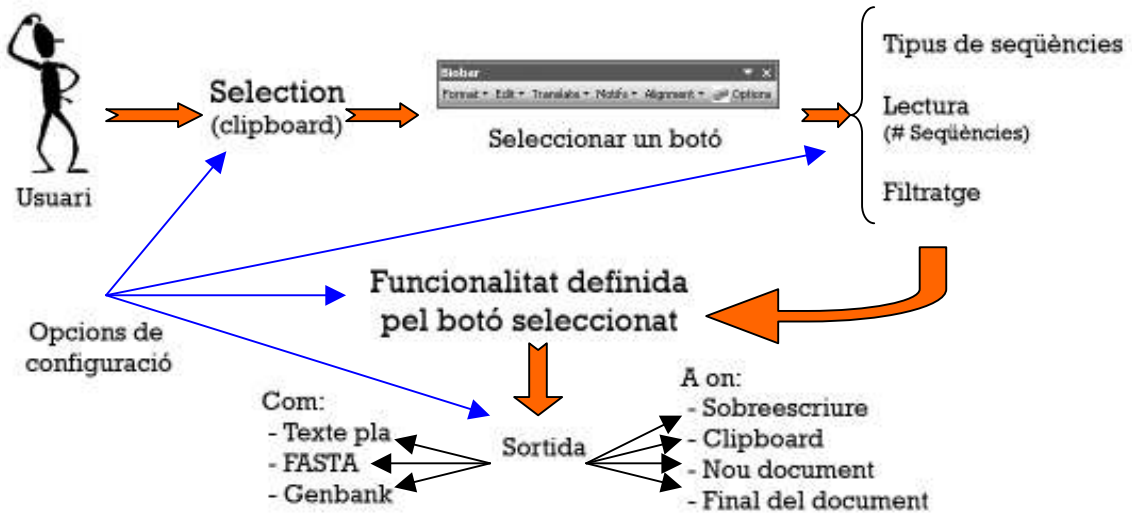


Figura 11 Esquema de funcionament i flux bàsic del programa.

Com es pot veure al diagrama anterior, el primer que ha de succeir per començar l'execució de l'*add-in* és que l'usuari seleccioni el text a processar sobre un full de Microsoft Word. Alternativament, com s'explica a continuació, l'usuari pot haver col·locat el text a processar al *clipboard*. Tot seguit, l'usuari ha de seleccionar una de les funcionalitats disponibles mitjançant els botons de l'eina, fet que passarà el control a la instància resident a memòria del nostre *add-in*.

Tot i que cadascun dels botons de la barra de l'*add-in* duu a terme una funcionalitat específica, tots ells comparteixen una funcionalitat comuna dedicada al tractament de l'entrada. Com es detalla a continuació, aquest tractament passa per decidir (comprovant les opcions a *Configuration*) d'on obtenir el text a processar (selecció o *clipboard*), esbrinar el número de seqüències que hi ha, el format en el que estan, i determinar-ne el tipus (DNA o proteïna) per fer-ne el filtratge corresponent.

Processada l'entrada, el control passa al mètode específic definit pel botó escollit, que durà a terme la funcionalitat triada (e.g. *reverse-complement* o traducció). Una vegada duts a terme els processos necessaris, el control passa de nou a un conjunt de mètodes comuns a tots els botons i dedicats al tractament de la sortida. En aquest cas, i accedint de nou a *Configuration*, caldrà decidir on volem escriure les sortides (sobreescriure, *clipboard*, document nou o final de document) i en quin format (text pla, FASTA o Genbank).

5.3 GESTIÓ INICIAL DE L'E/S

Com s'ha comentat anteriorment, existeixen varies opcions configurables relacionades amb l'entrada/sortida.

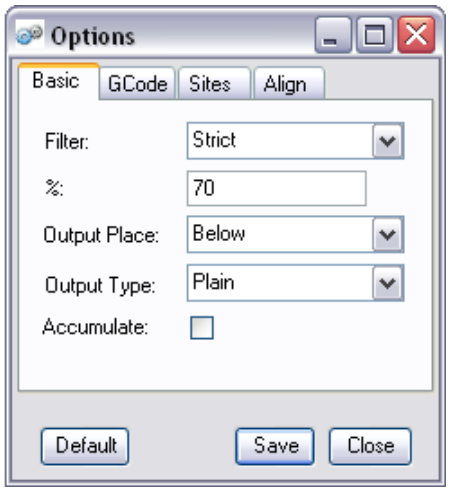


Figura 12 Formulari d'opcions de l'eina.

- El filtratge estricte o IUB.
- El percentatge per discriminar seqüències de DNA o proteïnes.
- El tipus de format de les seqüències de sortida i la seva posició.
- Si l'entrada es fa des del *clipboard* en el cas de no existir selecció (*Accumulate*).

5.3.1 ENTRADA

L'entrada sempre es gestiona de la mateixa manera per a totes les funcionalitats. Existeixen dues opcions d'on agafar les seqüències inicials. La primera és directament d'una selecció feta per l'usuari al document de Microsoft Word, i la segona consisteix en que si no hi ha cap text seleccionat per l'usuari, es comprova si hi ha algun al *clipboard*. Aquesta segona opció només s'aplica en el cas que el camps *Accumulate* de les opcions de configuració estigui marcat, en cas contrari, es mostra un avís per pantalla.

Una vegada es disposa del text d'entrada, s'ha d'avaluar si el que tenim són seqüències de DNA o de proteïna. Això es comprova mitjançat una altra opció existent a l'arxiu de configuració, on es determina a partir de quin percentatge de caràcters (A, C, G i T) es pot considerar que l'entrada és una seqüència de DNA o de proteïna. Un exemple d'aquest cas seria el mostrat a continuació:

Seqüència:	ACGTAGTGTA***MJ***TTGCA
Bases:	4 A´s
	2 C’s
	4 G´s
	5 T’s
Total:	15 bases
Longitud seqüència:	17
Resultat:	(15/17)*100 = 88,23%

Exemple 1 Càlcul del percentatge de discriminació entre DNA i proteïna.

Partint de la seqüència ACGTAGTGTA***MJ***TTGCA, considerant que es estricta, es realitza un comptatge de bases correctes, en aquest cas 4 A´s, 2 C’s, 4 G´s i 5 T’s, que fan un total de 15 bases. Com la longitud total de la seqüència és 17, calculem el percentatge de bases de DNA i obtenim un resultat (88,23%). Si a les nostres opcions el paràmetre de percentatge és menor al valor obtingut, aquesta seqüència serà tractada com una seqüència de DNA, en cas contrari, serà considerada una seqüència de proteïna.

En aquest punt, es disposaria d’una cadena de text que hauríem detectat si el que conté són seqüències de DNA o de proteïna. El següent pas consisteix en separar totes les seqüències existents i crear una col·lecció de seqüències, a la vegada que es van filtrant les seqüències per eliminar-ne possibles caràcters erronis.

Arribats a aquest punt ja disposem d’una col·lecció de seqüències d’entrada (1 o més d’una) amb la que realitzar qualsevol funcionalitat, així que es podria considerar finalitzada la part de pre-procés d’entrada.

5.3.2 SORTIDA

Pel què fa a la sortida, existeixen dos casos diferenciats de tractament, El primer cas correspon a la majoria de funcionalitats i explota un mètode comú (*Write*) definit a les classes *Collection* i *Sequence*. El segon cas correspon a la de recerca de motiuson es fa servir un mètode especial (*WriteMotifs*) tal i com es detalla a continuació.

Write

Existeixen dues opcions a tenir en compte a l'hora de tractar la sortida: en quin format es retorna el resultat i a on es retornarà.

Tenint en compte això, es va decidir implementar dos mètodes *Write*, un a *Collection* i un altre a *Sequence*, cadascún amb un repertori de funcions pròpies.

Sequence.Write

El mètode *Write* de *Sequence* és l'encarregat de triar en quin format s'ha de retornar la sortida. Comprovant les opcions de configuració, el mètode determina el format (text pla, FASTA o Genbank)[7] i processa la seqüència resultant del procés escollit a fi de deixar-la en el format escollit.

El resultat (i.e. la seqüència resultant) del procés de cada seqüència és retornat per *Sequence.Write* i s'acumula a un camp (*Result*) de la classe *Collection*, ja que serà aquesta classe la qui realitzi l'escriptura final, una vegada tingui tots els resultats de totes les seqüències processades correctament concatenats.

Collection.Write

El mètode *Write* de *Collection* crida seqüencialment al mètode *Write* de *Sequence* per cadascuna de les seqüències que conformen l'objecte *Collection*. Cada crida a *Sequence.Write* retornarà una seqüència en el format de sortida desitjat, que s'acumularà al camp *Result* de *Collection*. Quan tots els resultats ja estan degudament concatenats al camp *Result*, *Collection.Write* comprova l'arxiu de configuració per destriar el destí de la sortida. En aquest cas existeixen 4 opcions possibles: retornar el resultat sobreescrivint el text d'entrada, retornar-lo a sota del text existent, en un document nou, o copiar-lo al *clipboard*. *Collection.Write* es limita llavors a escriure el camp *Result* al destí escollit.

Resumint doncs, el mètode *Write* de *Collection* té com a funció acumular les sortides de cada seqüència de l'objecte Collection i obtenir el destí de la sortida a partir de les opcions d'usuari, per finalment escriure-hi el camp *Result* com a text de sortida. Per la seva banda, el

[7] Aquests formats seran explicats més endavant

mètode *Write* de *Sequence* es dedica únicament a retornar la seqüència resultant en el format escollit, prèvia consulta a l'arxiu de configuració.

WriteMotifs

Aquest cas només és utilitzat per les funcionalitats de recerca de motius. Això és degut a que pels mètodes de recerca de motius no obtenim una seqüència de sortida, sinó una taula amb un seguit de llocs trobats, la seva posició dins de la seqüència i el seu contingut d'informació.

Igual que en el cas del *Write*, per al *WriteMotifs* també existeixen dos mètodes, un a la classe *Sequence* i l'altre a la *Collection*.

Sequence.WriteMotifs

El mètode *WriteMotifs* de *Sequence* és l'encarregat d'escriure el resultat obtingut de una seqüència a la fila corresponent de la taula creada per mètode *WriteMotifs* de *Collection*.

Collection.WriteMotifs

El mètode *WriteMotifs* de *Collection* discerneix entre les opcions d'on es retorna el resultat, encara que ara només es tindran en compte dues, escriure a sota del text, o en un document nou, ja que no té utilitat eliminar la seqüència inicial amb un sobreescriure i no es viable guardar la taula resultant al *clipboard*. Una vegada escollit la opció de sortida, el mètode *WriteMotifs* de *Collection* crida seqüencialment al mètode *Write* de *Sequence* per cadascuna de les seqüències que conformen l'objecte *Collection*. Cada crida a *Sequence.Write* escriu el resultat a la fila corresponent de la taula creada pel *WriteMotifs* de *Sequence.*

Resumint doncs, el mètode *WriteMotifs* de *Collection* té com a funció obtenir el destí de la sortida a partir de les opcions d'usuari, per escriure-hi una taula amb 3 columnes i tantes files com seqüències resultants s'hagin obtingut. Per la seva banda, el mètode *WriteMotifs* de *Sequence* es dedica únicament a escriure la seqüència resultant a la fila corresponent de la taula.

5.4 DESCRIPCIÓ I IMPLEMENTACIÓ DE LES FUNCIONALITATS

Tot seguit s'explicaran les diferents funcionalitats implementades a l'eina, així com la seva utilitat i les seves característiques. Per facilitar la comprensió, s'han agrupat en grups segons el seu tipus de funcionalitat, igual que s'ha fet a la barra d'eines.

5.4.1 FORMAT

En aquesta categoria s'engloben 3 funcionalitats, una que consisteix en el filtratge de les seqüències d'entrada, i dues que s'encarreguen de canviar el format de les seqüències per l'escollit segons la opció.

Filtratge

El filtratge de seqüència consisteix, bàsicament, en convertir una seqüència qualsevol, proveïda per l'usuari, en una seqüència de DNA o proteïna preparada per la seva manipulació i anàlisi, traient-ne els caràcters que no corresponen a seqüències de DNA o proteïnes.

Per exemple, donada la seqüència:

```
01 ATGCCATTAC CATCAGTTAA CATGACTTAA CCAGGCCAGT
41 CCAGGCCAGT CATGACCGAT ATGCCATTAC CATGACCGAT
```

Obtindríem:

```
ATGCCATTACCATCAGTTAACATGACTTAACCAGGCCAGTCCAGGCCAGTCATGACCGATATGCCATTACCATGA
CCGAT
```

d'on s'han tret espais, números i salts de línia, deixant únicament la cadena de DNA.

Seqüències de aminoàcids

Les seqüències de DNA i de proteïnes estan formades per un nombre limitat de caràcters, repetits durant la seqüència. En el cas de les seqüències de proteïnes, aquests caràcters són `"G"`, `"A"`, `"V"`, `"L"`, `"I"`, `"M"`, `"F"`, `"W"`, `"P"`, `"S"`, `"T"`, `"C"`, `"Y"`, `"N"`, `"Q"`, `"D"`, `"E"`, `"K"`, `"R"`, `"H"`, `"*"`.

A continuació es mostra una taula amb l'alfabet d'aminoàcids en seqüències proteiques:

Aminoàcid	**Codi (3 Lletres)**	**Codi (1 lletra)**
Glycine	Gly	G
Alanine	Ala	A
Valine	Val	V
Leucine	Leu	L
Isoleucine	Ile	I
Methionine	Met	M
Phenylalanine	Phe	F
Tryptophan	Trp	W
Proline	Pro	P
Serine	Ser	S
Threonine	Thr	T
Cysteine	Cys	C
Tyrosine	Tyr	Y
Asparagine	Asn	N
Glutamine	Gln	Q
Aspartic acid	Asp	D
Glutamic acid	Glu	E
Lysine	Lys	K
Arginine	Arg	R
Histidine	His	H

Figura 13 Alfabet d'aminoàcids.

A més a més, en proteïnes s'admet un caràcter especial "*" que simbolitza la presència al DNA que d'ona lloc a la seqüència de proteïna d'un codó de stop de fi de traducció (UAG, UAA,UGA).

Seqüències de DNA

En el cas de DNA existeixen dues opcions: que les seqüències siguin estrictes o formades per caràcters IUB (posicions degenerades o parcialment desconegudes). Les seqüències estrictes estan composades pels caràcters `"A"`, `"T"`, `"C"`, `"G"`, mentre que les IUB es composen per `"A"`, `"T"`, `"C"`, `"G"`, `"R"`, `"Y"`, `"M"`, `"K"`, `"S"`, `"W"`, `"H"`, `"B"`, `"V"`, `"D"`, `"N"`, englobant els estrictes i permetent la definició de posicions degenerades, corresponents a varis possibles caràcters estrictes (e.g. W representa A/T). A la taula següent es pot veure el conjunt complet d'equivalències entre caràcters IUB i caràcters estrictes:

Codi IUB	Definició	Mnemònic
A	A	**A**denine
C	C	**C**ytosine
G	G	**G**uanine
T	T/U	**T**hymine/**U**racil
M	A/C	a**M**ino
R	A/G	pu**R**ine
W	A/T	**W**eak
S	C/G	**S**trong
Y	C/T	p**Y**rimidine
K	G/T	**K**eto
V	A/C/G	not T
H	A/C/T	not G
D	A/G/T	not C
B	C/G/T	not A
N	A/C/G/T	u**N**known

Figura 14 IUB Nucleotide code.

Resumint, el filtratge el que fa es eliminar els caràcters incorrectes segons si la seqüència d'entrada es de DNA o de proteïna i, en el cas que sigui de DNA, segons la opció escollida a la configuració (*Strict* o *IUB*).

Conversió de formats

Les opcions de presentació de seqüències són necessàries per convertir aquesta eina en una eina d'ús més general, compatible amb altres aplicacions i que permeti formatar en Microsoft Word les seqüències de forma adient.

Existeixen varis per representar seqüències de DNA i proteïnes, i gairebé es podria dir que cada aplicació té el seu propi format. En qualsevol cas, els formats que totes les aplicacions accepten són FASTA i Genbank, ja que són els més estesos. Per aquesta raó es va decidir afegir aquestes dues opcions de format a l'eina.

Format FASTA

El format FASTA defineix un estàndard d'emmagatzematge de seqüències de DNA i proteïna en arxius.

Una seqüència en format FASTA comença amb una línia de descripció, seguida de les línies de dades de la seqüència. La línia de descripció es distingeix de les dades de la seqüència mitjançant un símbol "major que" (>) a la primera columna. La paraula que segueix al símbol ">" és l'identificador de seqüència, i la resta de la línia n'és la descripció (ambdues opcionals). No ha d'existir cap espai entre el símbol ">" i la primera lletra de l'identificador, i es recomana que les línies corresponents a seqüència continguin un salt de línia cada 80 caràcters. En un arxiu FASTA, es considera que una seqüència finalitza si una nova línia comença amb ">", el que indica l'inici d'una nova seqüència, o quan es troba un EOF.

Un exemple de seqüència en format FASTA seria el següent:

```
>637343908 site-specific recombinase, phage integrase family [Shewanella
oneidensis MR-1: NC_004347] (-)strand
CGTGAAAGTTAATGAATGCCTTGATCCAATACAGATAGGTTTTTTCCGTTCTAATACTGTAACCACGCATACGCA
TTTCTTGTTGAAGGCCGCTTAGAAATGGGCTTTTTAGACTCATGATAACCCTCACCACCAAACGCTGTAATTTTA
TACAGTATTATTTAACATTCCCGGATATCAAGGAAAATAGTGAAAAAATTTTGCGCGTTTGTACATGTGCTA
```

Format Genbank

El format Genbank defineix un estàndard d'emmagatzematge de seqüències de DNA i proteïna anotades en arxius.

Una seqüència en format Genbank està formada per grups de 10 caràcters concatenats i separats entre ells per un espai. Cada 60 caràcters hi ha, a més, un salt de línia, i cada inici de línia està marcat pel número de caràcter corresponent.

Un exemple en format Genbank seria:

```
 1 CGTGAAAGTT AATGAATGCC TTGATCCAAT ACAGATAGGT TTTTTCCGTT CTAATACTGT
 61 AACCACGCAT ACGCATTTCT TGTTGAAGGC CGCTTAGAAA TGGGCTTTTT AGACTCATGA
121 TAACCCTCAC CACCAAACGC TGTAATTTTA TACAGTATTA TTTAACATTC CCGGATATCA
181 AGGAAAATAG TGAAAAAATT TTGCGCGTTT GTACATGTGC TA
```

Implementació

- **Filtratge**

Aquesta funcionalitat ha estat implementada a les classes filles de *Sequence*, és a dir, a *SequenceDNA* i a *SequenceProt*. Això és degut a que segons sigui d'un tipus o d'un altre, la seqüència podrà contenir uns caràcters o uns altres, fet que no permet implementar-ne un filtratge comú a la classe pare.

Aquest procediment utilitza l'atribut *Sequence* de *Sequence* (l'atribut continent de la seqüència), el modifica eliminant-ne els caràcters erronis i sobreescriu el mateix atribut, deixant-hi per tant la seqüència filtrada.

Pel cas de seqüències de DNA la funció és `Public Sub InFilter(ByRef Config As Configuration`, mentre que per proteïna és `Public Sub InFilter()`. La diferència en els paràmetres és que per seqüències de DNA s'ha de distingir el filtratge entre estricte o IUB.

Opcions:

Per aquesta funcionalitat, i només pel cas de DNA, s'utilitza la opció *Filter* (1), que determina si, al filtrar, es farà de forma estricta o en codi IUB.

(1) Opció **Filter** de la pestanya **Basic** del formulari d'opcions.

- **Conversió a FASTA**

Aquesta funcionalitat s'ha implementat com un mètode (`ConvertFASTA`) de la classe *Sequence*, ja que les diferències entre DNA i proteïna no afecten al procés de conversió.

```
Public Function ConvertFASTA() As String
```

El mètode s'ha definit sense paràmetres, ja que obté la capçalera de la seqüència i la seqüència en sí dels atributs *Sequence* i *SequenceHead*[8] de la classe *Sequence* i en retorna el valors concatenats amb un salt de línia entremig.

- **Conversió a Genbank**

Aquesta funcionalitat, igual que l'anterior canvi de format, s'ha implementat com un mètode (`ConvertGenbank`) de la classe *Sequence*, ja que les diferències entre DNA i proteïna no afecten al procés de conversió.

```
Public Function ConvertGenbank() As String
```

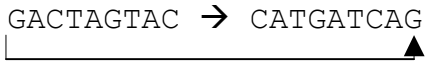
El mètode s'ha definit sense paràmetres, ja que obté la seqüència de l'atribut *Sequence* de la classe *Sequence* i en retorna la seqüència resultant amb el número del caràcter de cada línia i amb els espaiadors corresponents.

5.4.2 EDICIÓ BÀSICA

L'edició bàsica de seqüència comprèn la manipulació de seqüències de DNA o proteïna per dur a terme certes operacions d'interès amb aquestes (e.g. girar i complementar una seqüència de DNA de forma simultània, a fi d'observar com és la seqüència de DNA que es troba a la cadena complementària). A continuació es descriuen aquestes operacions.

Reverse

Girar (*reverse*) una seqüència de DNA consisteix, bàsicament, a invertir la seqüència llegint-la de dreta a esquerra. Per exemple:



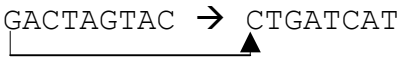
Complement

Una altra funcionalitat habitual és la de complementar (*complement*) una seqüència de DNA. Aquest procés consisteix en obtenir la mateixa seqüència (i.e. mateix ordre) però amb les bases complementades.

[8] A l'atribut *SequenceHed* es guarda la capçalera de la seqüència en cas que la seqüència estigui en format FASTA.

Les bases es complementen seguint les següents bijectives: G ↔C i A↔T, ja que aquestes són precisament les bases que es troben encarades a la doble cadena de DNA. La raó és que G (Guanina) i C (Citosina) són capaces de formar tres enllaços pont d'hidrogen entre elles, mentre que A (Adenina) i T (Timina) només en poden formar dos. Aquesta diferència de força entre els enllaços GC i AT provoca tensions a la doble cadena que acaben recargolant-la, donant-li la famosa forma de doble hèlix.

Un exemple bàsic de complementació seria:



Revers-Complement

També podem unir les dues funcionalitats anteriors per obtenir-ne una de nova, que consisteix en girar i complementar (*reverse-complement*) una seqüència. Al dur a terme aquesta funcionalitat obtenim la seqüència de DNA llegida tal i com es llegiria recorrent la cadena complementària de DNA. Per exemple:

Implementació

- ***Reverse***

Aquesta funcionalitat ha estat implementada dins la classe pare *Sequence*, ja que tant les seqüències de DNA com les de proteïna poden ser girades. És una funció que té com a nom *Reverse* i modifica l'atribut *Sequence* que conté la seqüència original amb la seqüència girada.

```
Public Sub Reverse()
```

- ***Complement***

Aquesta funcionalitat només està definida a la classe *SequenceDNA*, ja que no té sentit complementar una cadena de proteïna, perquè aquestes seqüències estan formades per aminoàcids i els aminoàcids no tenen un complement natural.

```
Public Sub Complement()
```

Aquest mètode, igual que *Reverse*, modifica l'atribut *Sequence* i hi guarda el valor resultant de l'operació, és a dir, la seqüència complementada.

En cas que la seqüència d'entrada estigui en format *IUB*, els caràcters també seran complementats amb els complements naturals dels seus constituents.

Per exemple, R={A, G} serà complementat pet Y={T, C}, ja que T i C són els complements naturals de A i G.

A continuació es mostra la taula completa de complements *IUB*:

Base	Complement
A	T
C	G
G	C
T	A
R	Y
Y	R
M	K
K	M
S	S
W	W
H	D
B	V
V	B
D	H
N	N

Figura 15 Complementació de les bases de DNA.

- ***Reverse-Complement***

Per la mateixa raó que no té sentit complementar una cadena de proteïna, tampoc no té sentit girar i complementar aquest tipus de cadenes. Per tant, *Reverse-Complement* només està definit a la classe *SequenceDNA*.

```
Public Sub ReverseComplement()
```

Reverse-Complement també modifica l'atribut *Sequence* i hi guarda el valor resultant de l'operació, és a dir la seqüència girada i complementada.

5.4.3 TRADUCCIÓ

Les funcionalitats de traducció engloben els processos de convertir una seqüència de DNA a proteïna i el seu invers, convertir-ne una de proteïna a DNA. A continuació s'explicaran totes dues funcionalitats i es podrà comprovar que, a diferència de la traducció de DNA a proteïna, per a la traducció inversa cal tenir en compte vàries consideracions.

Traducció

La traducció d'una seqüència de DNA consisteix en un mapatge de la seqüència de DNA de longitud L en una de proteïna de longitud L/3, seguint el codi genètic i llegint la seqüència de DNA per triplets (anomenats codons). Per exemple:

```
GACTAGTACGTACTAGTC
        ↓
D*YVLV
```

El codi genètic (*Figura 4)* és la regla de correspondència entre els nucleòtids en que es basen els àcids nucleics i els aminoàcids en que es basen les proteïnes.

La traducció d'una seqüència de DNA consisteix en un mapatge de la seqüència de DNA de longitud L en una de proteïna de longitud L/3, seguint el codi genètic. Per exemple:

```
GACTAGTACGTACTAGTC
        ↓
D*YVLV
```

Tot i que la majoria d'éssers vius utilitzen el mateix codi, existeixen petites variacions i cal considerar la possibilitat d'utilitzar diferents codis genètics.

Traducció inversa

La traducció inversa d'una seqüència d'aminoàcids de longitud L en una de DNA de longitud 3L consisteix en el procés invers a la traducció. És a dir, donat un aminoàcid, assignar el triplet (o codó) que li correspon a la seqüència de DNA. Donat que el codi genètic és redundant, amb varis codons (grups de 3 bases de DNA) codificant el mateix aminoàcid, desfer el mapatge DNA→proteïna de forma fidedigna és impossible.

Entre d'altres utilitats de la traducció inversa, cal destacar el disposar d'una seqüència de proteïna i voler obtenir-ne la de DNA per poder dissenyar primers o altres molècules que s'enganxin/tallin/etc. específicament al DNA que codifica la proteïna. Una altra utilitat, menys obvia que l'anterior, és voler clonar un gen d'un organisme en un altre. Perquè el plegament de la proteïna sigui òptim, cal adaptar el *codon usage* de l'organisme en que es vol clonar el gen. Això significa que s'ha de fer una traducció inversa que optimitzi el *codon usage* de l'organisme que rep el gen.

La opció més senzilla per solucionar el problema de la redundància en la traducció inversa és introduir redundància en el resultat d'aquesta, és a dir, realitzar una traducció inversa amb caràcters *IUB*, utilitzant simultàniament la taula del codi genètic (*Figura 4*) i la del codi IUB (*Figura 14*) a fi de descriure les posicions ambigües al DNA. Per exemple, si la proteïna conté un aminoàcid P, al traduir-lo inversament aplicant el codi genètic s'obtindrien, de forma equiprobable, els codons CCT, CCC, CCA o CCG. Si s'utilitza el codi *IUB*, aquestes sortides es poden unificar com CCN. Així, en una traducció inversa sobre l'organisme *Escherichia coli* obtindríem:

YVLV → TAYGTNYTNGTN

Per obtenir una seqüència estricta en traducció inversa que s'aproximi a la real, es pot fer ús del *codon usage*. El *codon usage* és el percentatge d'ús de cada codó que l'organisme, de mitja, fa servir per codificar proteïnes. Una estratègia simple com triar el codó més utilitzat per cada aminoàcid permet donar eficiències relativament bones (70%) en la traducció inversa. Per fer-ho, cal permetre a l'usuari carregar les dades de *codon usage* de l'organisme, llegint-les en el format estàndard de la *Codon Usage Database* (http://www.kazusa.or.jp/codon/). Amb l'exemple anterior s'obtindria el següent resultat:

YVLV → TACGTACTAGTC

Implementació

- **Classes auxiliars**

Per implementar aquesta funcionalitat s'han creat dues noves classes, la classe *GeneticCode* i la classe *GCode*.

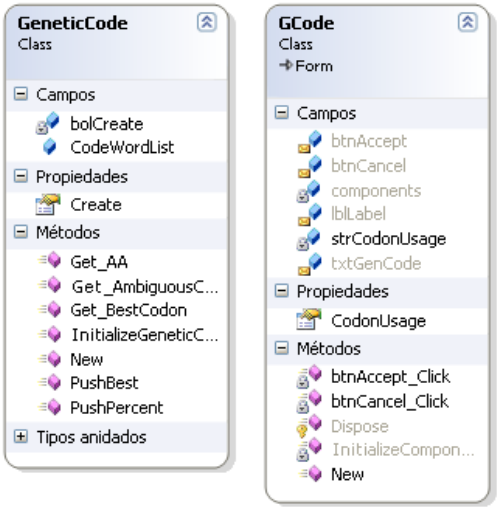


Figura 16 Diagrama de classes. Codi genètic.

GeneticCode

La classe *GeneticCode* s'encarrega de definir l'estructura d'un codi genètic en forma de llista i carregar-lo des de format XML. Concretament carrega cada codó amb el seu aminoàcid corresponent, i el codó ambigu que li correspon, en una estructura *CodeWord*, que és l'element bàsic de la llista.

A més, pel cas de traducció inversa de proteïna a DNA (si a les opcions s'ha escollit que sigui un filtratge estricte i no ambigu) també es carrega a *CodeWord* el percentatge de cada aminoàcid, segons el codi genètic facilitat per l'usuari en un format determinat.

L'estructura *CodeWord* està doncs formada pels següents camps:

```
Codon (String)          → Codó
AA (String)             → Aminoàcid
Percent (Double)        → Percentatge d'ús de l'amoniàcid
Best (Boolean)          → Marca si és el millor AA possible
                           (major ús)
AmbiguousCodon (String) → Codi ambigu (conté el codi IUB del codó
                           degenerat corresponent a aquest aminoàcid)
```

Aquesta classe disposa de tres mètodes per obtenir un codó o un aminoàcid segons l'aminoàcid o el codó desitjat, permetent doncs un doble accés per contingut a fi d'implementar la traducció i la traducció inversa. Aquests mètodes són:

```
Public Function Get_AA(ByVal Codon As String) As String
```

→ Obtè l'aminoàcid corresponent al codó facilitat per paràmentre.

```
Public Function Get_BestCodon(ByVal AA As String) As String
```

→ Obtè el codó corresponent a l'aminoàcid facilitat per paràmentre.

```
Public Function Get_AmbiguousCodon(ByVal AA As String) As String
```

→ Obtè el codó ambigu corresponent a l'aminoàcid facilitat per paràmentre.

Els codis genètics disponibles es troben a un arxiu XML (GeneticCode.xml), on s'accedeix al crear un nou objecte *GeneticCode*.

GCode

La segona classe (*GCode*) és un formulari per poder obtenir el codi genètic i l'ús de codons d'un organisme a partir de l'estàndard definit a *Codon Usage Database*. En el cas que l'usuari seleccioni la opció en que se li demana que introdueixi un codi genètic, aquest formulari apareixerà perquè l'usuari pugui facilitar dita informació prèvia consulta al web.

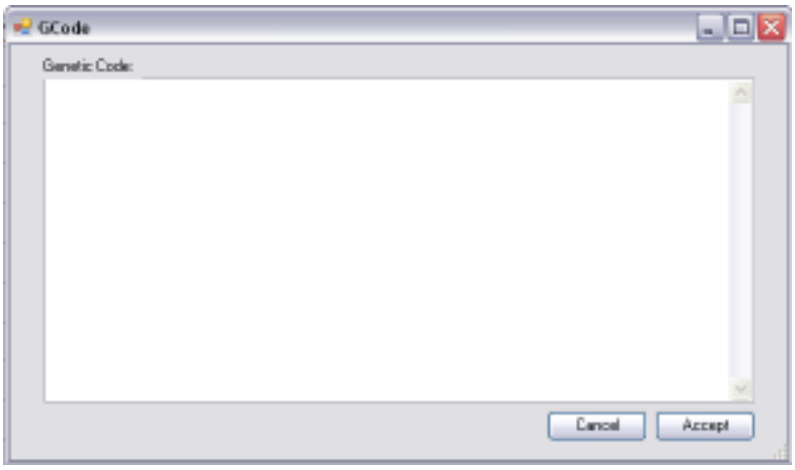


Figura 17 Formulari per l'obtenció del codi genètic per part de l'usuari.

- ***Translate***

Donat que aquesta funcionalitat consisteix en la traducció de cadenes de DNA a proteïna, ha estat implementada a la classe *SequenceDNA*.

```
Public Sub Translate(ByVal GenCode As GeneticCode,
  ByRef seqProt As SequenceProt)
```

El mètode *Translate* rep com a paràmetres el codi genètic sobre el que es farà la traducció i una seqüència de proteïna on es guardarà el resultat, ja que en aquest cas, el resultat serà una seqüència de proteïna.

En el cas que rebem una seqüència de DNA amb codi *IUB*, aquesta serà convertida prèviament a codi estricte mitjançant la funció *Collapse*, que és cridada des del codi de la funció *Translate*.

```
Public Sub Collapse()
```

El *Collapse* es fa perquè és impossible, en la majoria de casos, assignar un únic aminoàcid a un triplet de DNA degenerat. La funció *Collapse* converteix una seqüència *IUB* en estricta, amb la mateixa probabilitat per a cada caràcter. Per exemple, si una de les bases és R, existirà una probabilitat del 50% que ens retorni una A i un altre 50% que el resultat sigui una G. En el cas d'una N existiria un 25% de probabilitats que el resultat fos una A, C, G o T.

També s'utilitzarà la funció `Public Function Get_AA(ByVal Codon As String) As String` que, com s'ha explicat a dalt, per a un codó donat ens retornarà el seu aminoàcid corresponent al codi genètic escollit.

Opcions:

Per a la traducció es necessita el codi genètic a utilitzar (1), i aquest es troba definit com a una opció de configuració, en la que l'usuari pot escollir entre diferents codis genètics a utilitzar.

(1) Opció **Genetic Code** de la pestanya **GCode** del formulari d'opcions.

- ***RevTranslate***

A diferència de la traducció, aquesta funcionalitat només ha estat implementada en la classe *SequenceProt*, ja que consisteix en la traducció de cadenes de proteïna a DNA.

```
Public Sub RevTranslate(ByVal GenCode As GeneticCode,
  ByRef Config As Configuration, ByRef seqDNA As SequenceDNA)
```

El mètode *RevTranslate* rep com a paràmetres el codi genètic sobre el que es farà la traducció, la configuració i una seqüència de DNA on es guardarà el resultat, ja que en aquest cas el resultat serà una seqüència de DNA.

Com s'ha comentat anteriorment, per fer una traducció inversa necessitem les opcions de configuració, ja que l'usuari pot traduir la seqüència d'entrada en una de DNA en format *IUB* i admetent ambigüitat, o bé pot haver carregat un *codon usage* i desitjar obtenir la "millor" seqüència.

Segons la tria de l'usuari respecte al tipus de traducció inversa esmentat a dalt, el mètode *RevTranslate* obtindrà el codó ambigu o el millor codó per a cada aminoàcid de la seqüència, utilitzant els mètodes comentats anteriorment:

```
Public Function Get_BestCodon(ByVal AA As String) As String
```

→ Obtè el codó corresponent a l'aminoàcid facilitat per paràmentre.

```
Public Function Get_AmbiguousCodon(ByVal AA As String) As String
```

→ Obtè el codó ambigu corresponent a l'aminoàcid facilitat per paràmentre.

Opcions:

Per a la traducció inversa es necessita el codi genètic a utilitzar (1), que es troba definit com a una opció de configuració (en la que l'usuari pot escollir entre diferents codis genètics a utilitzar) i si la traducció es farà buscant la millor opció o amb codons ambigus (2).

(1) Opció **Genetic Code** de la pestanya **GCode** del formulari d'opcions.

(2) Opció **Rev. Translate** de la pestanya **GCode** del formulari d'opcions.

5.4.4 DETECCIÓ DE MOTIUS

Les recerques en seqüència busquen identificar motius o característiques d'alt nivell en una seqüència de DNA. Per exemple, una característica d'alt nivell al DNA és l'existència de regions que codifiquen proteïnes (ORF – *Open Reading Frames*). D'entre els motius, pot resultar interessant identificar motius sobre els que s'acoblen certes proteïnes (típicament en regions promotores de gens) per tal de dur a terme diferents regulacions sobre l'expressió de gens.

Informació en motius al DNA

La presència de seqüències operadores al DNA, reconegudes específicament per proteïnes a fi de realitzar una funció específica com activar o reprimir un gen, evidencia clarament que el DNA conté informació més enllà del missatge codificat als gens, i que aquesta informació es transmet de forma efectiva mitjançant un procediment de reconeixement específic de certs motius al DNA per part de certes proteïnes. Si hi ha, doncs, un procés de transmissió d'informació resulta lògic aplicar la teoria de la informació per analitzar-lo.

Contingut d'informació d'un motiu

Normalment, quan estudiem motius al DNA amb teoria de la informació és perquè tenim a la nostra disposició una col·lecció alineada de N seqüències de DNA de longitud L que sabem que són reconegudes per una proteïna. Donada aquesta col·lecció alineada de seqüències, que defineix un motiu o patró reconegut per la proteïna, podem inferir fàcilment les freqüències d'aparició de cada base a cada posició del motius.

Col·lecció	1	2	3	4	5
ATGTC	A	T	G	T	C
GTAAT	G	T	A	A	T
ATAGT	A	T	A	G	T
ATAAT	A	T	A	A	T
ATGAC	A	T	G	A	C
AAGAC	A	A	G	A	C
ATCGC	A	T	C	G	C
ATTTT	A	T	T	T	T
ATTGG	A	T	T	G	G
AGATC	A	G	A	T	C
TAGTT	T	A	G	T	T

	1	2	3	4	5
A	0.82	0.18	0.36	0.45	0.00
C	0.00	0.00	0.09	0.09	0.36
T	0.09	0.73	0.18	0.18	0.55
G	0.09	0.09	0.36	0.27	0.09

Exemple 2 Col·lecció alineada de seqüències i freqüències d'aparició.

Resulta obvi que, en aquest cas, la nostra col·lecció de seqüències ens indica, per cada posició del motiu, quina és la nostra incertesa sobre la base que ocupa aquella posició donada una seqüència que formi part de la col·lecció.

Contingut d'informació per una posició del motiu

Per una posició donada del motiu es pot obtenir l'entropia condicionada H(X|Y) que correspon a l'incertesa mitja sobre el valor (X) de la posició L, sabent que es tracta de la posició L d'una seqüència reconeguda (Y) per la proteïna:

$$H(X \mid Y) = H_{after}(l) = \left[-\sum_{S_l \in \Omega} \left(p(S_l) \cdot \log_2 \left(p(S_l) \right) \right) \right]$$

on L correspon a la posició del motiu, $p(S_l)$ és la freqüència de la base S, amb $\Omega = \{A, C, T, G\}$ a la posició L del motiu.

que és la incertesa que es manté sobre una posició determinada una vegada es sap que la proteïna s'enganxa a una seqüència.

Abans de saber que la proteïna s'enganxa a una seqüència, la incertesa respecte a cadascuna de les posicions d'aquesta és igual i ve donada per la distribució de bases al genoma. Així doncs,

$$H(X) = H_{before}(l) = \left[-\sum_{S \in \Omega} \left(f(S) \cdot \log_2 \left(f(S) \right) \right) \right]$$

on L correspon a la posició del motiu, $f(S)$ és la freqüència de cadascuna de les bases $S \in \Omega$, $\Omega = \{A, C, T, G\}$ a la posició L del motiu.

De manera que, per una posició determinada del motiu, s'obté:

$$I(l) = H_{before}(l) - H_{after}(l) = \left[-\sum_{S\in\Omega} \left(f(S) \cdot \log_2 \left(f(S) \right) \right) \right] - \left[-\sum_{S_l\in\Omega} \left(p(S_l) \cdot \log_2 \left(p(S_l) \right) \right) \right]$$

que és la reducció d'incertesa (el guany d'informació) que s'obté per una posició d'una seqüència en observar/saber que la proteïna reconeix la seqüència.

Seguint l'exemple anterior (*Exemple 2*), es pot calcular la reducció d'incertesa respecte a la posició 3 d'una seqüència al saber que aquesta és reconeguda per la proteïna que reconeix el motiu definit per la col·lecció com:

Collection	**1**	**2**	**3**	**4**	**5**
ATGTC	A	T	G	T	C
GTAAT	G	T	A	A	T
ATAGT	A	T	A	G	T
ATAAT	A	T	A	A	T
ATGAC	A	T	G	A	C
AAGAC	A	A	G	A	C
ATCGC	A	T	C	G	C
ATTTT	A	T	T	T	T
ATTGG	A	T	T	G	G
AGATC	A	G	A	T	C
TAGTT	T	A	G	T	T

	1	**2**	**3**	**4**	**5**
A	0.82	0.18	0.36	0.45	0.00
C	0.00	0.00	0.09	0.09	0.36
T	0.09	0.73	0.18	0.18	0.55
G	0.09	0.09	0.36	0.27	0.09

Genome freqs	
A	0.246
C	0.254
T	0.246
G	0.254

$$\begin{aligned} I(3) &= H_{before}(3) - H_{after}(3) = -\left[0.246 \cdot \log_2(0.246) \cdot 2 + 0.254 \cdot \log_2(0.254) \cdot 2\right] + \\ &+ \left[0.36 \cdot \log_2(0.36) + 0.09 \cdot \log_2(0.09) + 0.18 \cdot \log_2(0.18) + 0.36 \cdot \log_2(0.36)\right] = \\ &= -\left[-0.995 - 1.004\right] + \left[-0.531 - 0.314 - 0.447 - 0.531\right] = -\left[-1.999\right] + \left[-1.823\right] = 0.177 \end{aligned}$$

Exemple 3 Càlcul de la reducció d'incertesa respecta a la posició 3.

Avaluació de seqüències

Quan busquem llocs d'acoblament de proteïnes o seqüències particulars en una seqüència, normalment ho fem en referència a un patró o motiu establert. Cal doncs un criteri per avaluar la contribució a la unió amb la proteïna de cadascuna de les bases d'una seqüència particular.

Existeixen diferents mètodes per calcular aquest factor, com per exemple, el model de *Berg & von Hippel* (Berg and von Hippel 1988), però una de les més intuïtives i sòlides és utilitzar el RI_{dif}.

Aquest criteri, introduït per Michael C. O'Neill (O'Neill 1991), és una forma elegant d'avaluar l'adequació d'una seqüència a una col·lecció de seqüències que defineixen un motiu prototip utilitzant el contingut d'informació de la col·lecció de forma diferencial. Això implica calcular el contingut d'informació a cada posició RI(l) de forma normal [RI^-(l)] i recalcular-lo després afegint la nova seqüència a avaluar com a part de la col·lecció [RI^+(l)]. El nou índex d'informació per cada posició vindrà donat per:

$$RI_{dif}(l) = RI^-(l) \cdot (RI^+(l) - RI^-(l))$$

En definitiva, primer es calcula el RI sense la seqüència afegida i després amb la seqüència afegida, de forma que el diferencial (per cada posició de forma independent) donarà idea de com de bé "casa" la seqüència a avaluar amb el motiu ja establert.

A continuació es mostra un exemple de càlcul del RI_{dif} per facilitar-ne la comprensió:

Collection	**1**	**2**	**3**
AGT	A	G	T
AAT	A	A	T
CGT	C	G	T
AGC	A	G	C
GAT	G	A	T
ACC	A	C	C
CAT	C	A	T
CAA	C	A	A
AAT	A	A	T
ACG	A	C	G
CGA	C	G	A
AGG	A	G	G
AAA	A	A	A
ATG	A	T	G
ATG	A	T	G
ATT	A	T	T
AAT	A	A	T
CGT	C	G	T
AGC	A	G	C
CCT	C	C	T
CTT	C	T	T
ACA	A	C	A
CCT	C	C	T
CTC	C	T	C
CAT	C	A	T
ATA	A	T	A

	1	**2**	**3**
A	0,58	0,31	0,19
C	0,38	0,19	0,15
T	0,00	0,23	0,50
G	0,04	0,27	0,15

Genome freqs	
A	0.25
C	0.25
T	0.25
G	0.25

$$RI^{-} = 1.064408$$

Sequence to eval			
CAA	C	A	A
Motif freqs	0,38	0,31	0,19
Base freqs	0,25	0,25	0,25

Es torna a calcular el RI afegint la nova seqüència, i s'obté:

$$RI^{+} = 1.052849$$

Per acabar, amb els dos valors calculats, substituïm a la fórmula i s'obté:

$$RI_{dif} = 1.064408 \cdot (1.052849 - 1.064408) = -0.00866$$

que ens indica que la seqüència no "casa" gaire bé amb el motiu establert.

Exemple 4 Càlcul de l'adequació d'una seqüència a una col·lecció de seqüències.

En endavant, i sense citar-ho explícitament, s'assumirà que el mètode utilitzat per avaluar seqüències particulars respecte a un motiu és el descrit a dalt, és a dir, el RI_{dif}.

Implementació

- **Classes auxiliars**

Per implementar aquesta funcionalitat s'han creat varies classes que són necessàries per cada una de les opcions disponibles (*SiteSearch*, *DyadSearch* i *RepeatSearch*).

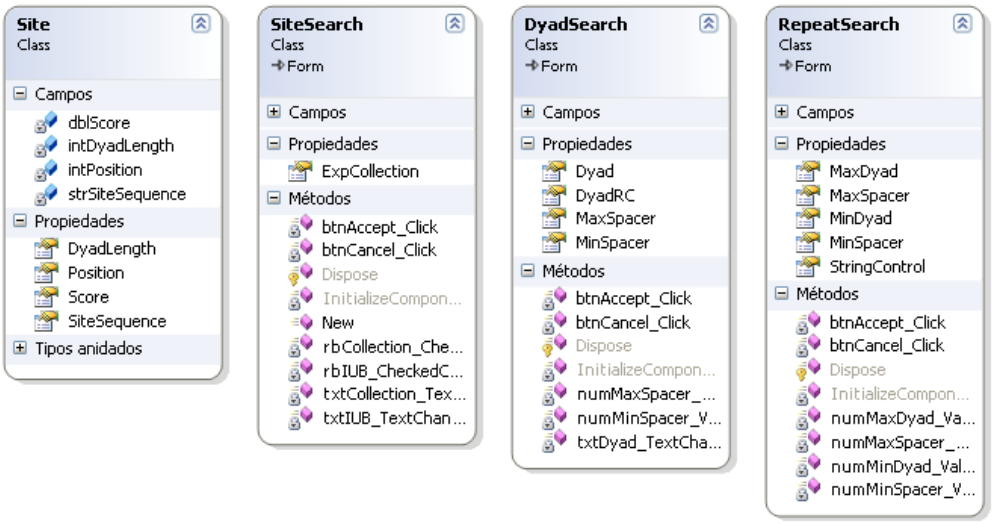


Figura 18 Diagrama de classes. Cerca de llocs.

La primera classe que es pot observar a la figura anterior (*Figura 18*) és la classe *Site*. Aquesta classe defineix el que es necessita per identificar un lloc trobat a la seqüència d'entrada. *Site* defineix la posició del lloc dins de la seqüència, la seva puntuació o *score*, i la seqüència del lloc pròpiament dit. Aquesta informació és la que es mostrarà com a resultat a la sortida de la recerca de llocs.

Position	Sequence	Score
4	GGAGA	-1,38081601289227
5	GAGAA	-0,374299343340978
7	GAATA	0,026771820485334
12	GCGTA	-1,38081601289227
15	TACGA	-1,38081601289227
18	GAGCG	-1,1764416709936
22	GCGAT	-1,38081601289227
24	GATTC	-0,775370507167291
29	GAGCA	-0,775370507167291
31	GCATC	-1,38081601289227

Figura 19 Exemple de resultants d'una cerca de llocs.

Les altres tres classes corresponen als formularis de les tres possibilitats de cerca que existeixen a la nostra eina. Cada formulari configura les opcions de cerca de cada una de les funcionalitats.

SiteSearch

La classe *SiteSearch* correspon a la cerca de llocs a partir d'una col·lecció de seqüències alineades o a partir d'una seqüència *IUB* (per cada caràcter *IUB* existent obtenim una col·lecció de caràcters estrictes[9]). Així doncs, a *SiteSearch* l'únic que l'usuari necessita introduir és la col·lecció de seqüències, o bé la seqüència IUB (*Figura 20*).

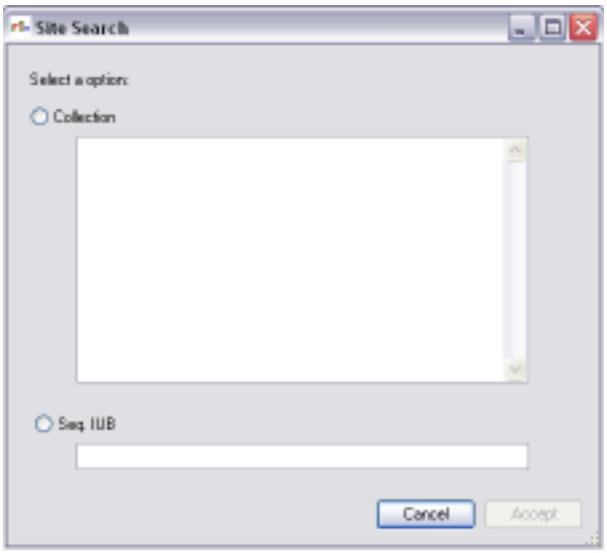


Figura 20 Formulari per definir la cerca segons el mètode *Site Search.*

DyadSearch

Una extensió del mètode anterior, és la detecció de repeticions i repeticions inverses (i.e. palíndroms) gestionada per la classe *DyadSearch*. Degut a raons de construcció de la proteïna, la majoria de motius als que s'acoblen les proteïnes tenen forma palindròmica o de repetició directe amb espaiador, el que es coneix com estructura *dyad-spacer-dyad*[10] (parell-espaiador-parell).

[9] Per exemple, WT generaria una col·lecció amb els motius AT i TT, corresponents a descompondre la W en A i T.

[10] *Dyad* és literalment "parell" i s'aplica als motius *dyad-spacer-dyad* en base a que cadascun dels *dyads* es correspon amb l'altre (bé per repetició directa o per palindròmica) i hi ha d'haver dos, amb un espaiador més o menys variable.

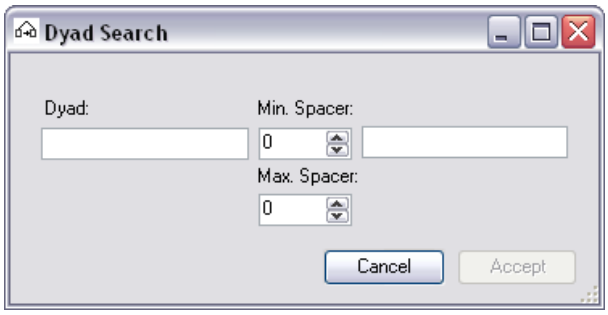


Figura 21 Formulari per definir la cerca segons el mètode *Dyad Search.*

A partir d'una seqüència *IUB* de cada *dyad* i de la longitud de l'espaiador (entre uns valors màxim i mínim escollits) l'usuari pot definit la cerca que vol realitzar (*Figura 21*). L'usuari introdueix la seqüència *IUB* del primer *dyad*, i al segon *dyad* apareix la seqüència corresponent segons tingui selecciona la opció *Search* com a *Palindrome* o *Direct repeat*. Aquesta seqüència es modificable posteriorment per l'usuari pel cas que desitgi realitzar algun canvi. A més dels *dyads*, l'usuari introdueix una longitud màxima i mínima per l'espaiador que separarà els dos *dyads*.

RepeatSearch

Una tercera opció de recerca, gestionada per la classe *RepeatSearch*, és la recerca de motius de tipus *dyad-spacer-dyad* lliure, acotada per un número de *mismatches* entre *dyads*. La diferència d'aquesta amb l'anterior és que a *RepeatSearch* no existeix cap seqüència pels *dyads*, sinó una longitud màxima i mínima per cada un d'ells, igual que per l'espaiador.

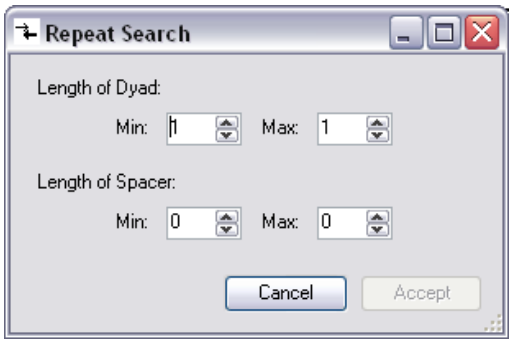


Figura 22 Formulari per definir la cerca segons el mètode *Repeat Search*

- ***SiteSearch***

Aquest mètode consisteix en la identificació de sub-cadenes en una seqüència donada, fet que permet ressaltar sub-seqüències candidates, i té les següents característiques:

```
Public Sub Site_RI_SiteSearch(ByRef Col As Collection, ByRef
Config As Configuration)
```

El mètode rep com a paràmetres una col·lecció de seqüències, que correspon a la col·lecció de seqüències alineades que defineix el motiu o patró, i la configuració, ja que es necessitaran algunes de les opcions, com ara el *GC Content* i el número de llocs a retornar.

Al crear o carregar la col·lecció de seqüències alineades es calcula el contingut d'informació per cada posició del motiu (*RI*), que serà el que necessitarà aquest mètode per obtenir un resultat.

El càlcul de RI per cada posició del motiu es realitza mitjançant una crida a la funció *CalculateRI* des de el constructor de l'objecte *Collection* que guarda el motiu, i se'n guarda el resultat a cada columna del motiu i el total en un atribut de la classe anomenat *RI_total*.

```
Public Sub CalculateRI()
```

La funció *Site_RI_SiteSearch* necessita d'una altra funció *Site_RI_Eval*, que utilitza *CalculateRIplus* que es troba a la classe *SequenceDNA*.

```
Public Function Site_RI_Eval(ByVal SeqToEval As SequenceDNA, ByRef
   Col As Collection) As Double

Public Function CalculateRIplus(ByVal A As Double, ByVal C As
   Double, ByVal G As Double, ByVal T As Double, ByRef Col As
   Collection) As Double
```

La funció *Site_RI_Eval* afegeix una nova seqüència (de la mateixa mida que la col·lecció) al motiu i calcula el nou *RI*, mitjançant la funció *CalculateRI*. Aquest procés es realitza per tantes finestres de la mida del motiu com pugui obtenir de la seqüència d'entrada, i *Site_RI_Eval* retorna el RI_{dif} per cadascuna d'elles.

$$RI_{dif}(l) = RI^{-}(l) \cdot (RI^{+}(l) - RI^{-}(l))$$

Una vegada obtingut el RI_{dif} per una de les finestres, existeixen 3 opcions a considerar. Primer es comprova si ja està cobert el número de resultats desitjats. Si encara no tenim tots els resultats demanats, la finestra és afegida a la llista de resultats, i aquesta s'ordena segons la puntuació dels llocs (`Public Sub Sort(ByVal Motifs() As Site.Motif)`). Si ja es té el número desitjat de resultats, es comprova si la puntuació o contingut d'informació del lloc trobat és millor que el d'algun de la llista, i en cas afirmatiu es substitueix (mitjançant el

procediment *Change*) el nou lloc pel de pitjor puntuació. En cas que el lloc trobat en última instància sigui pitjor que tots els obtinguts anteriorment, aquest es descarta.

```
Public Sub Change(ByVal Motifs() As Site.Motif, ByVal SiteSequence
   As String, ByVal Position As Integer, ByVal Score As Double, ByVal
   DyadLength As Integer)
```

Un exemple del resultat que podem obtenir al utilitzar *SiteSearch* es mostra a continuació:

Position	Sequence	Score
13	ATACTGTATAAAATTACAGCGT	0,0191445316879159
54	TCGACGATCGAGCGGGCATTAT	-0,206197620614414
71	ATTATATCGATCGGATCGGAGC	-0,20777494415815
84	GATCGGAGCGCGATGCTAGATA	-0,19005570660803
90	AGCGCGATGCTAGATACTGTAT	-0,214250734707577
93	GCGATGCTAGATACTGTATAAA	-0,194524477266979
96	ATGCTAGATACTGTATAAAATT	-0,186537515047867
103	ATACTGTATAAAATTACAGCGT	0,0191445316879159
118	ACAGCGTCTAGCTAGCTACTAG	-0,203133733191472
140	CTACTGATCGTCGATCGTAGCT	-0,204847764114533

Figura 23 Resultat de la funcionalitat *Site Search*.

Com es pot observar, a més de la posició del lloc i del seu contingut d'informació (Score), també es mostra la seqüència concreta amb el grau de conservació de cada posició en la col·lecció (com més gran la mida de la lletra més conservada està la posició).

Opcions:

Per aquesta funcionalitat, s'utilitza la opció GC Content (1), que determina el percentatge de les bases de DNA a fi de calcular l'entropia a priori, la opció de número de resultats (2), on s'introdueix en número de resultats que es desitgen obtenir, i la opció que determina si es farà el càlcul automàtic del número de resultats o s'utilitzarà la opció anterior (3).

(1) Opció **CG Content** de la pestanya **Sites** del formulari d'opcions.

(2) Opció **# Results** de la pestanya **Sites** del formulari d'opcions.

(3) Opció **Automatic # Results**[11] de la pestanya **Sites** del formulari d'opcions.

[11] El càlcul automàtic del número de resultats es basa en estimar la freqüència d'aparició de seqüències d'un motiu a partir de la seva conservació (i.e. el RI) i la longitud de la seqüència on es fa la recerca.

- ***DyadSearch***

Pel cas de la cerca de llocs mitjançant *dyads*, també s'utilitzen les dues funcions comentades a l'apartat anterior: *Site_RI_Eval* i *CalculateRIplus*, i de forma anàloga a com s'ha explicat.

Aquest mètode difereix de l'anterior en que ara es buscaran dues seqüències separades per un número de caràcters (espaiador). És a dir, es generaran dues col·leccions, una per cada *dyad*, es calcularà el seu RI^-, després el RI^+ per cada *dyad* i, al final, es sumaran els RI_{dif} de cadascuna d'elles, considerant-ho el *Score* del motiu compost. La longitud de l'espaiador no afecta ni decideix si un lloc és millor o pitjor.

En resum, es seguirà el mateix mètode explicat anteriorment, amb la diferència que ara, en comptes d'una col·lecció a la que calcular el RI^-, afegir-li una seqüència, calcular el seu RI^+ i calcular el RI_{dif}, n'hi hauran dues, una per cada *dyad*.

Com es pot observar a continuació:

```
Public Sub Site_RI_DyadSearch(ByRef Col As Collection, ByRef ColRC As
   Collection, ByVal MinSpacer As Integer, ByVal MaxSpacer As
   Integer, ByRef Config As Configuration)
```

Site_RI_DyadSearch rep com a paràmentres dues col·leccions (una per cada *dyad*), la configuració d'usuari, i els valors màxim i mínim de l'espaiador entre els *dyads*.

A continuació es mostra un exemple de sortida per una recerca *DyadSearch*.
La sortida en aquest cas tindria el mateix format que el cas anterior (*SiteSearch*), amb la diferència que en aquest cas es mostra el grau de conservació de cada posició en la col·lecció per a cada un dels *dyads*, i aquests estan en negreta per diferenciar-los de l'espaiador que, a més de no estar en negreta, té la mínima mida de lletra possible.

Position	Sequence	Score
5	cGATGCAGATAc	-0,256441068592979
11	aGATACTGTATAa	-0,256441068592979
32	cGTTCGGCGATTc	-0,256441068592979
44	cTATCGGCTATCG	-0,256441068592979
51	cTATCGACGATCG	-0,256441068592979
70	cATTATATCGATCG	-0,256441068592979
72	tTATATCGATCG	-0,256441068592979
94	cGATGCTAGATAc	-0,256441068592979
101	aGATACTGTATAa	-0,256441068592979
144	tGATCGTCGATCG	0,00998981452353931

Exemple 5 Resultat de la funcionalitat *DyadSearch*.

Opcions:

Per aquesta funcionalitat, s'utilitza la opció GC Content (1), que determina el percentatge de les bases de DNA a fi de calcular l'entropia a priori, la opció de número de results (2), on s'introdueix en número de resultats que es desitgen obtenir, i la opció que determina si es farà el càlcul automàtic del número de resultats o s'utilitzarà la opció anterior (3). A més, en aquest cas s'utilitzarà la opció que determina el format per defecte del segon *dyad*, palindròmic o repetició directa del primer (4).

(1) Opció **CG Content** de la pestanya **Sites** del formulari d'opcions.

(2) Opció **# Results** de la pestanya **Sites** del formulari d'opcions.

(3) Opció **Automatic # Results** de la pestanya **Sites** del formulari d'opcions.

(4) Opció **Search** de la pestanya **Sites** del formulari d'opcions.

- ***RepeatSearch***

Aquest tercer mètode de cerca de motius ha estat implementat d'una manera diferents als dos anteriors, ja que en aquest cas no coneixem cap característica de com seran els llocs on s'enganxarà la proteïna. La única informació de que es disposa és la longitud dels *dyads* a buscar i l'espaiador que hi haurà al mig (valors màxims i mínims).

```
Public Sub Site_RI_RepeatSearch(ByVal RSearch As RepeatSearch, ByRef
  Config As Configuration)
```

En aquest cas els paràmetres d'entrada són la configuració i l'objecte de la classe *RepeatSearch* on l'usuari ha introduït el patrons de cerca, és a dir, el valor màxim i mínim dels *dyads* i de l'espaiador.

Ara, com no disposem de cap col·lecció per poder realitzar els càlculs del RI_{dif}, el que es tindrà en compte per puntuar cada porció de seqüència obtinguda serà la quantitat de *matches* (igualtats) entre el primer *dyad* i el segon. Per cada posició del primer *dyad* es compara amb la mateixa posició del segon *dyad*, si coincideixen contem 1 *match*.
Per acabar de calcular la puntuació per a cada lloc trobar, s'ha de tenir en compte la longitud dels *dyads*, ja que ara aquesta pot variar en el cas que la longitud màxima i la mínima siguin diferents. Per tant, la puntuació correspondrà al número d'encerts trobats dividit per la longitud del *dyad*.

A causa de la diferència en el càlcul de la puntuació, aquest mètode no utilitza les funcions *Site_RI_Eval* i *CalculateRI*. A part d'això, la restà de la funcionalitat és la mateixa que en els casos anteriors.

A continuació es mostra un exemple de sortida per una recerca *RepeatSearch*.

Com en aquest cas no disposem de cap col·lecció, no es pot mostrar el grau de conservació de cada posició en la col·lecció per a cada un dels *dyads*, per tant, totes les bases tindran la mateixa mida i bases que corresponen als *dyads* aniran marcades en negreta, ja que en aquest cas, al tenir diferents mides de *dyads*, aquests es podrien confondre amb l'espaiador.

Position	Sequence	Score
16	CTGTATAAAATTACAG	0,8
17	TGTATAAAATTACA	1
37	GGCGATTCTATCGGC	0,8
68	GGCATTATATCGATCGG	0,8
83	GGATCGGAGCGCGATGC	0,833333333333333
83	GGATCGGAGCGCGATGC	0,8
84	GATCGGAGCGCGATG	0,8
86	TCGGAGCGCGATGCTA	0,8
107	TGTATAAAATTACA	1
126	TAGCTAGCTACTAGCTA	0,833333333333333

Exemple 6 Resultat de la funcionalitat *RepeatSearch*.

Opcions:

Per aquesta funcionalitat, s'utilitza la opció GC Content (1), que determina el percentatge de les bases de DNA, la opció de número de results (2), on s'introdueix en número de resultats que es desitgen obtenir, i la opció que determina si es farà el càlcul automàtic del número de resultats o s'utilitzarà la opció anterior (3). A més, en aquest cas s'utilitzarà la opció que determina el format del segon *dyad*, palindròmic o repetició directa del primer (4).

(1) Opció **CG Content** de la pestanya **Sites** del formulari d'opcions.

(2) Opció **# Results** de la pestanya **Sites** del formulari d'opcions.

(3) Opció **Automatic # Results** de la pestanya **Sites** del formulari d'opcions.

(4) Opció **Search** de la pestanya **Sites** del formulari d'opcions.

5.4.5 ALINEAMENT

El problema de l'alineament

Com totes les altres ciències, tant la genòmica com la proteòmica i totes les altres -òmiques fan ús extensiu de la comparació dels elements amb els que treballen, que en el cas de la bioinformàtica acostumen a ser seqüències de DNA, RNA o proteïna. Bé sigui per buscar motius conservats en genòmica comparativa, saber si una regió del genoma és un gen o identificar la funció d'una proteïna, poder comparar dues o més seqüències entre sí es converteix en un prerequisit indispensable a l'hora de treballar en bioinformàtica.

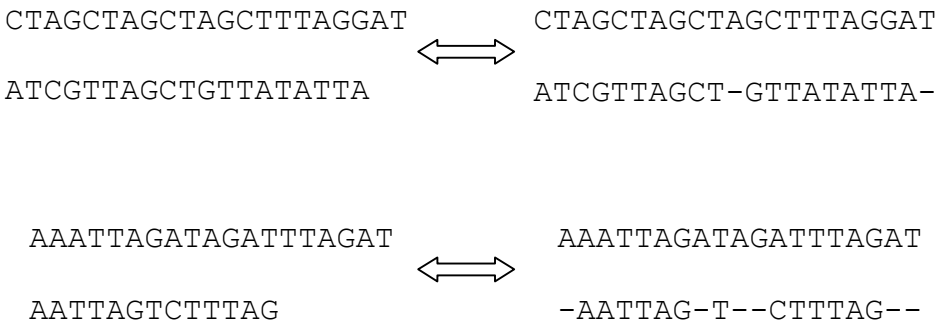
Mètrica i comparació

Com és natural, per poder comparar dues coses cal primer establir una mètrica que n'avaluï la diferència. En informàtica, dos bytes es poden comparar, per exemple, en funció de la suma

de les seves posicions diferents. Abans de posar-nos a comptar diferències o distàncies, però, és convenient decidir què comparem. En informàtica, la discretització i sistematització del procés d'informació ens fan la feina fàcil. En el cas de les seqüències biològiques, la manca de límits ben establerts complica força aquest procés.

Alinear seqüències de forma òptima, és a dir, establir una homologia posicional entre seqüències que ens indiqui quina posició de cada seqüència es correspon a una altra, és un problema NP-complet de molt difícil resolució a mesura que augmentem el número de seqüències.

A continuació es mostren uns exemples de seqüències alineades i no alineades:



Exemple 7 Seqüències no alineades i alineades.

Paràmetres d'alineament

Tal i com es planteja en bioinformàtica, el procés d'alineament és, en essència, un procés d'optimització. De cara a dur-lo a terme cal, evidentment, definir el joc de paràmetres sobre el que es produeix l'optimització, així com la mesura que desitgem optimitzar. En el cas de l'alineament de seqüències en bioinformàtica, bé siguin seqüències d'aminoàcids o bé de proteïnes, el problema versa sempre sobre l'alineament de seqüències d'origen biològic. Aquest tipus de seqüències estan regides per l'evolució per selecció natural i, en conseqüència, serà l'evolució la qui dicti la naturalesa i el rang dels paràmetres, així com la mètrica d'optimització.

Mètriques d'alineament

La mètrica bàsica de qualsevol sistema d'alineament està implícita en el procés mateix d'alineament. És evident que, per avaluar un alineament, el més intuïtiu és comptar el número de caràcters ben alineats, però si no disposem d'informació contextual la qualitat real de l'alineament pot oscil·lar fortament en funció de la permissivitat a la inserció de *gaps*[12] i, malauradament, la pròpia naturalesa de gran part dels esdeveniments evolutius coneguts (insercions i delecions) ens imposa la necessitat d'acceptar *gaps* als alineaments. El procediment d'assignar costos a les diferents possibilitats d'alineament (*matches*[13], *gaps*, *mismatches*[14]) es coneix com *scoring* i, com en el cas dels paràmetres d'alineament, vindrà dictat pels principis evolutius que regulen l'evolució de les seqüències a alinear.

Scoring *bàsic*

El procediment més bàsic imaginable per dur a terme l'avaluació d'un alineament de seqüències és assignar costos fixes a les tres circumstàncies possibles. Un esquema que, de fet, es fa servir comunament en alineament de seqüències de DNA/RNA és assignar un cost +1 (recompensa de fet) al *match*, un cost -1 al *mismatch* i un cost -2 al *gap*.

Un altre factor important en l'assignació de costos als *gaps* prové de la seva interpretació evolutiva. Degut als mecanismes moleculars que intervenen en ells, els processos d'inserció i deleció rarament afecten només un caràcter, sinó que, en la majoria de casos, una cadena substancialment llarga de DNA o RNA és tallada de o inserida a la cadena original. De cara a modelar aquest comportament, els algorismes d'alineament incorporen dues mètriques de cost per *gap*. El GOP (*Gap Opening Penalty*) és el cost associat a l'obertura d'un *gap*. Una vegada disposem d'un *gap* obert, però, els algorismes apliquen el GEP (*Gap Extension Penalty*) en les posicions adjacents al *gap* original. Habitualment el GOP és un ordre de magnitud superior al GEP, a fi de modelar la més alta probabilitat d'un únic esdeveniment evolutiu.

Matrius de substitució

Quan s'han d'avaluar les substitucions en seqüències de proteïnes, els models evolutius (tot i existir) són impossibles d'aplicar degut, fonamentalment, a dues raons. D'una banda, en proteïnes tenim 20 aminoàcids diferents per comptes de 4 nucleòtids, i això dispara

[12] Gap: Inserció o deleció de bases en una seqüència.
[13] Match: Coincidència d'una base o aminoàcid en una posició de dues o més seqüències.
[14] Mismatch: Divergència d'una base o aminoàcid en una posició de dues o més seqüències.

ràpidament el número de paràmetres que ha d'encabir el model. D'altra banda, els canvis entre aminoàcids, a part de no ser equiprobables, tampoc no tenen el mateix efecte evolutivament. Per exemple, la substitució d'un aminoàcid hidròfob per un d'hidròfil tindrà una repercussió evolutiva molt més acusada que la substitució del mateix aminoàcid per un d'hidròfob[15]. Per tot això, quan es volen modelar les diferents probabilitats de substitució entre aminoàcids es fan servir matrius de substitució empíriques per comptes de models teòrics com en el cas del DNA.

C Cys	12																	
S Ser	0	2																
T Thr	-2	1	3															
P Pro	-3	1	0	6														
A Ala	-2	1	1	1	2													
G Gly	-3	1	0	-1	1	5												
N Asn	-4	1	0	-1	0	0	2											
D Asp	-5	0	0	-1	0	1	2	4										
E Glu	-5	0	0	-1	0	0	1	3	4									
Q Gln	-5	-1	-1	0	0	-1	1	2	2	4								
H His	-3	-1	-1	0	-1	-2	2	1	1	3	6							
R Arg	-4	0	-1	0	-2	-3	0	-1	-1	1	2	6						
K Lys	-5	0	0	-1	-1	-2	1	0	0	1	0	3	5					
M Met	-5	-2	-1	-2	-1	-3	-2	-3	-2	-1	-2	0	0	6				
I Ile	-2	-1	0	-2	-1	-3	-2	-2	-2	-2	-2	-2	-2	2	5			
L Leu	-6	-3	-2	-3	-2	-4	-3	-4	-3	-2	-2	-3	-3	4	2	6		
V Val	-2	-1	0	-1	0	-1	-2	-2	-2	-2	-2	-2	-2	2	4	2	4	
F Phe	-4	-3	-3	-5	-5	-5	-4	-6	-5	-5	-2	-4	-5	0	1	2	-1	9
Y Tyr	0	-3	-3	-5	-3	-5	-2	-4	-4	-4	0	-4	-4	-2	-1	-1	-2	7	10
W Trp	-8	-2	-5	-6	-6	-7	-4	-7	-7	-5	-3	2	-3	-4	-5	-2	-6	0	0	17
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W

Figura 24 PAM 250 scoring matrix of Dayhoff mutation data matrix.

Les matrius de substitució, ideades per Dayhoff (Dayhoff, Schwartz et al. 1978) a mitjans dels 1970, són matrius de 20x20 elements en què cada element Mi,j correspon a la probabilitat de substitució d'un aminoàcid (Ai) per un altre (Aj).

Existeixen moltes matrius de substitució obtingudes de forma empírica a partir de col·leccions d'alineaments, com ara, les PAM (*Figura 24*), les BLOSUM (Henikoff and Henikoff 1992) o les GONNET.

[15] Això és el què s'anomenen mutacions silencioses, és a dir, sense efectes "visibles" a nivell de selecció natural.

Alineament global vs. alineament local

Podem dividir l'alineament segons la forma en que processen les seqüències. En aquest sentit, les tècniques d'alineament no es divideixen segons la longitud de les seqüències a processar, sinó sobre la consideració d'aquestes seqüències com a elements compostos o indivisibles. Aquest enfocament divideix les tècniques d'alineament en globals i locals. Els algorismes d'alineament global són aquells que busquen un alineament òptim que inclou tots els caràcters de totes les seqüències. Per contra, els algorismes d'alineament local són aquells que busquen un alineament òptim que inclogui les regions més similars de les cadenes.

Els alineaments globals són més indicats quan les seqüències són similars i de mida semblant, mentre que l'alineament local és més útil en casos de seqüències més dissimilars en les quals existeixen elements similars dins d'un context poc semblant.

Alineament global: l'algorisme de Needleman-Wunsch

L'algorisme de Needleman-Wunsch (Needleman and Wunsch 1970) és el mètode exacte d'alineament més utilitzat. Basat en programació dinàmica, l'algorisme de Needleman-Wunsch està encarat a l'obtenció d'alineaments globals òptims.

Idea general

La idea intuïtiva de l'algorisme de Needleman-Wunsch és molt fàcil de copsar. Donada una mesura de qualitat, l'algorisme busca maximitzar la qualitat de l'alineament, que és equivalent a trobar el millor alineament d'entre tots els possibles. Per fer-ho, l'algorisme de Needleman-Wunsch utilitza una matriu d'alineament, sobre la que traçarà el camí òptim.

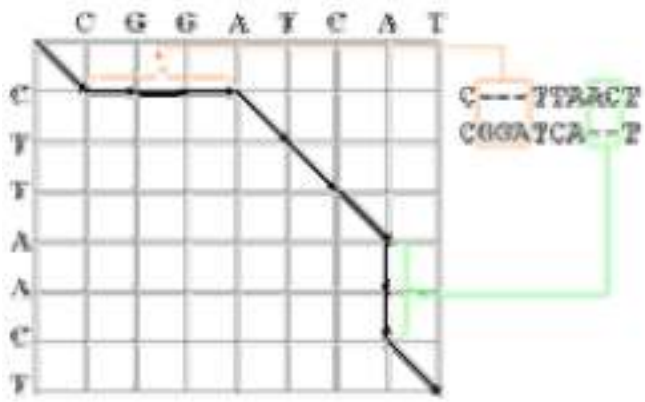


Figura 25 La matriu d'alineament a l'algorisme de Needleman-Wunsch i el concepte visual d'obtenir l'alineament òptim recorrent-la.

La matriu d'alineament és una matriu amb tantes files/columnes com caràcters té cadascuna de les dues seqüències a alinear. L'algorisme de Needleman-Wunsch anota sobre aquesta matriu el cost òptim d'alineament a cada posició de forma progressiva, aprofitant la propietat de sub-problemes sobreposats. Gràcies a que el problema té subestructura òptima, l'alineament global òptim es pot obtenir recorrent el camí maximal entre les diagonals de la matriu. Així doncs, l'algorisme de Needleman-Wunsch té dues parts diferenciades. La primera, de propagació dreta-avall, omple la matriu amb el cost dels sub-alineaments òptims. La segona traça enrere (*trace-back*) el camí òptim que representa l'alineament global. Cal destacar que pot existir més d'una solució òptima.

Propagació

La propagació endavant comença amb una inicialització, que consisteix en inserir una primera fila i columna que representen la possibilitat d'iniciar l'alineament obrint *gaps*[16]. A partir d'aquí, i utilitzant el sistema de *scoring* desitjat, la propagació consisteix en emplenar la matriu de dalt a baix i d'esquerra a dreta, començant per l'extrem superior esquerra.

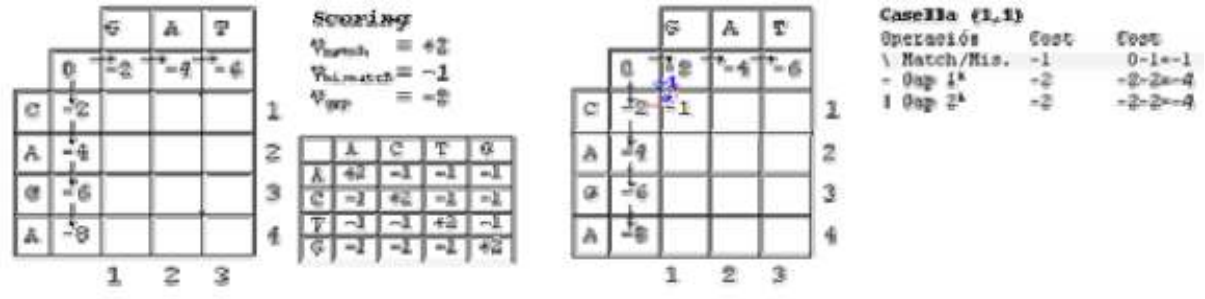


Figura 26 Inicialització de la matriu d'alineament, matriu de *scoring*, i emplenat de la primera casella (1,1).

Per emplenar una cel·la de la matriu d'alineament ens caldrà sempre disposar de totes les solucions òptimes prèvies (i.e. les caselles més a dalt i a l'esquerra). Inicialment, disposem d'aquestes caselles gràcies a la fila/columna *zero* introduïda. A partir d'aquí el càlcul és sistemàtic: per cada casella, avaluarem quin és el moviment que generà un cost mínim en funció de l'estat òptim previ i la funció de *scoring*. Seguint l'exemple de la *Figura 26*, a la primera casella (1,1) tenim tres opcions possibles: (a) un moviment horitzontal, des de (0,1),

[16] Donat que cada posició de la matriu representa el cost de l'alineament en aquell punt, la casella (0,0) té cost zero. A partir d'aquí, acumulem el cost de moure'ns en horitzontal i vertical (i.e. obrint *gap* a una seqüència o l'altra) fins a emplenar la fila/columna zero.

que equival a inserir un *gap* a la primera seqüència, (b) un moviment vertical, des de (1,0) que equival a inserir un *gap* a la segona seqüència, o (c) un moviment diagonal des de (0,0) que equival a una substitució (en aquest cas és un *mismatch* C-G).

El procés de propagació continua de forma ordenada fins a omplir per complet la matriu d'alineament. A partir d'aquí, s'inicia la segona part de l'algorisme, el *trace-back*, que ens donarà l'alineament òptim.

Traçat enrere (*trace-back*)

Acabada la propagació haurem obtingut una matriu on, a cada casella, hi ha acumulat el cost òptim de l'alineament que pot arribar, per camí òptim, a aquella posició. Així, a la casella (2,2) hi tenim un +1, que és el màxim valor d'alineament que podem obtenir si només volem alinear CA i GA: un *match* (+2) i un *mismatch* (- 1). Si durant la propagació endavant s'han guardat els apuntadors de canvi de casella (les sagetes), el procés de *trace-back* és trivial.

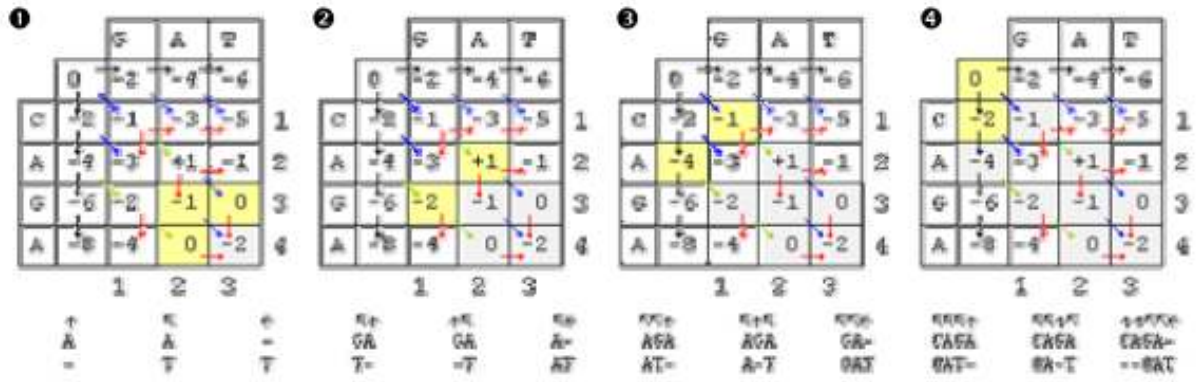


Figura 27 *Trace-back* i reconstrucció de l'alineament òptim. Iniciem el *trace-back* (1) a la darrera casella (4,3) i desfem i anotem tots els camins possibles (groc) des de les quals hi podem haver arribat. Per cada nova casella (2), repetim el procés (3) fins arribar a l'origen (4). Els tres alineaments resultants tenen tots el mateix cost: -2.

Per iniciar el *trace-back* només cal situar-se a la casella de la darrera casella. A partir d'aquí anirem tirant enrere pels camins permesos (els marcats per les sagetes; són camins òptims) per obtenir l'alineament (o alineaments) global òptim (*Figura 27*)

Alineament local: l'algorisme de Smith-Waterman

Com ja s'ha comentat, un dels problemes de l'alineament global és que regions força similars entre les seqüències a alinear es poden perdre degut a la necessitat d'intentar ajustar d'altres regions més grans però més dispars. L'algorisme de Smith-Waterman (Smith and Waterman 1981) és una variació del de Needleman-Wunsch orientada a trobar alineaments locals

òptims. Per aconseguir-ho, l'algorisme de Smith-Waterman introdueix una sèrie de diferències significatives amb el de Needleman-Wunsch.

Diferències amb Needleman-Wunsch

Una de les principals diferències entre el Needleman-Wunsch i l'algorisme de Smith-Waterman es troba en la regla d'emplenat utilitzada en la propagació. La regla d'emplenat de l'algorisme Smith-Waterman és la següent:

$$Casella_{i,j} = \max\left[Casella_{i-1,j-1} + score(a_i, b_j), \max\left(Casella_{i-k,j} - W \cdot k\right), \max\left(Casella_{i,j-l} - W \cdot l\right), 0\right]$$

on $Casella_{i,j}$ és el valor de la casella, $score(a_i, b_j)$ és la puntuació del *match*/*mismatch* entre a i b, W és la penalització per *gap* (típicament -(1+0.3·k)) i k la longitud del *gap*.

En essència, la regla funciona igual que la de Needleman-Wunsch (escollim el tipus de moviment que més puntuació ens dóna). La novetat és la presència de zero com a valor d'emplenat. A la pràctica, això fa que les caselles no puguin tenir valors negatius o, el què és el mateix, que no acumulem puntuació negativa en zones de difícil alineament. Això permet a l'algorisme Smith-Waterman centrar-se en les regions de bon alineament sense perdre-les de vista degut a regions *dolentes* entremig.

L'altra gran diferència amb Needleman-Wunsch la trobem en el *trace-back*. A diferència Needleman-Wunsch, on el *trace-back* comença a la darrera casella i acaba a la primera, a Smith-Waterman el *trace-back* comença a la casella de millor puntuació de la matriu, independentment d'on es trobi, i acaba en trobar un zero. Això deslliura l'algorisme de la necessitat de retornar un alineament global, i li permet retornar l'alineament local més òptim.

Tant el Needleman-Wunsch com el Smith-Waterman són algorismes d'alineament exactes que garanteixen trobar el millor alineament global o local d'entre tots els possibles.

Implementació

Per a la implementació de la funcionalitat d'alineaments és necessària una classe nova, la classe *Matrix*, que servirà per crear les matrius de substitució i d'alineament. Aquest fet fa que sigui convenient crear dues subclasses, una per cada tipus de matriu a emprar, és a dir una subclasse *MatrixSubs* i una altra *MatrixAlign*.

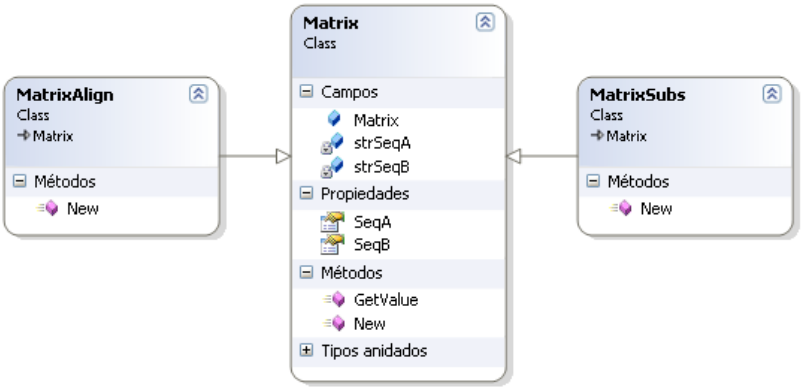


Figura 28 Diagrama de classes. Matrius.

Com es pot observar al diagrama de classes anterior, tots els mètodes i atributs estan continguts a la classe pare i el fet que diferencia a totes dues subclasses és el constructor. El constructor de la classe *MatrixAlign* crea la matriu i inicialitza per defecte els seus valors, mentre que el de la classe *MatrixSubs*, a més de crear la matriu, carrega les dades d'un arxiu XML segons la matriu escollida a les opcions de configuració.

La classe pare conté dos atributs, *SeqA* i *SeqB* que corresponen a les dues seqüències a alinear i, per al cas concret d'una matriu, a la fila i la columna de la mateixa. A més, a part del constructor, la classe pare disposa d'un mètode per retornar el valor d'una casella de la matriu a partir dels caràcters de les seqüències existents com a fila i columna.

Les matrius contenen a cada cel·la una estructura anomenada *Cell*, que conté els següents camps:

```
Value (Double)      → Valor de la cel·la
Kh (Integer)        → Contador de gaps horitzontals[17]
Kv (Integer)        → Contador de gaps verticals[18]
Diagonal (Boolean)  → Determina si s'ha arribat a la cel·la per
                      la diagonal.
```

Com que per a realitzar un alineament es necessiten, com a mínim, dues seqüències, els mètodes per implementar els alineaments es troben a la classe *Collection*, ja que és aquesta classe la que pot treballar amb més d'una seqüència.

[17] Permet dur un comptatge de gaps estesos (GEP) horitzontalment a la matriu d'alineament, per tal de poder calcular el GAP a partir del GOP i el GEP.
[18] Permet dur un comptatge de gaps estesos (GEP) verticalment a la matriu d'alineament, per tal de poder calcular el GAP a partir del GOP i el GEP.

- ***Global Alignment***

Per poder realitzar l'alineament, primer s'han de crear dues matrius, una de substitució i una altra d'alineament. La de substitució estarà carregada amb els valors corresponents segons la opció que l'usuari tingués marcada a les opcions, i la d'alineament és la que s'anirà creant per trobar l'alineament entre les dues seqüències.

Per aquesta raó, a aquesta funció se li passen com a paràmetre les opcions de configuració. Els altres paràmetres que necessita són les dues matrius creades i un valor (*Type*) que determina el tipus d'alineament, si de seqüències de DNA o de proteïna.

```
Public Sub GlobalAlign(ByRef MatAlign As MatrixAlign, ByRef MatSubs As
 MatrixSubs, ByRef Config As Configuration, ByVal Type As String)
```

Aquest mètode aplica l'algorisme de Needleman-Wunsch. Primer inicialitza la matriu (*ford-ward*) i després la recorre des del final per trobar l'alineament (*trace-back*).

Les dues seqüències alineades resultants substitueixen a les d'entrada en la llista de seqüències de la col·lecció. Per tant, per retornar el resultat només cal escriure el mateix objecte *Collection* que s'ha creat a l'inici amb la entrada.

A continuació es mostra un exemple d'alineament global:

Seqüències d'entrada:

```
AGTCAGCTAGGCGGCATCAGAGCTAGCTAGGGCATGATCGAT
CATCGATCGGATATATGCTATCGAGA
```

Seqüències alineades:

```
AGTCAGCTAGGCGGCATCAGAGCTAGCTAGGGCATGATCGAT
---CATCGA-TCGG-AT-ATA--T-GCTA--TC--GA--GA-
```

Exemple 8 Alineament global de seqüències.

Opcions:

Per a la funcionalitat d'alineament global, s'utilitza la opció de matriu de substitució a utilitzar (1), que determina quina matriu de substitució s'utilitzarà en cas d'alinear seqüències de DNA i quina en el cas de proteïna, la opció de GOP (2) i la de GEP (3).

(1) Opció **Substitution Matrix (DNA/Protein)** de la pestanya **Align** del formulari d'opcions.

(2) Opció **GOP** de la pestanya **Align** del formulari d'opcions.

(3) Opció **GEP** de la pestanya **Align** del formulari d'opcions.

- ***Local Alignment***

Per a l'alineament local es segueix el mateix guió que pel global, primer es creen les matrius de substitució i alineament, i es carreguen o inicialitzen, i després es du a terme l'alineament.

Com es pot observar a continuació la funció d'alineament local té els mateixos paràmetres que la d'alineament global.

```
Public Sub LocalAlign(ByRef MatAlign As MatrixAlign, ByRef MatSubs As
 MatrixSubs, ByRef Config As Configuration, ByVal Type As String)
```

En el cas de l'alineament local, en comptes d'utilitzar l'algorisme de Needleman-Wunsch, s'utilitza el de Smith-Waterman.

Les dues seqüències alineades resultants, com en el cas anterior, també substitueixen a les d'entrada en la llista de seqüències de la col·lecció, per tant, per retornar el resultat només cal escriure el mateix objecte *Collection* que s'ha creat a l'inici amb la entrada.

A continuació es mostra un exemple d'alineament local:

Seqüències d'entrada:

```
AGTCAGCTAGGCGGCATCAGAGCTAGCTAGGGCATGATCGAT
CATCGATCGGATATATGCTATCGAGA
```

Seqüències alineades:

```
CAGCTAGGCGGCATCAGAGCTAGCTAGG
CATCGA-TCGG-ATATATGCTATCGAGA
```

On es pot veure que l'algorisme només retorna la zona (local) d'alineament òptim (zona vermella:

```
AGTCAGCTAGGCGGCATCAGAGCTAGCTAGGGCATGATCGAT
   CATCGA-TCGG-ATATATGCTATCGAGA
```

Exemple 9 Alineament local de seqüències.

Opcions:

Per a la funcionalitat d'alineament local, s'utilitza la opció de matriu de substitució a utilitzar (1), que determina quina matriu de substitució s'utilitzarà en cas d'alinear seqüències de DNA i quina en el cas de proteïna, la opció de GOP (2) i la de GEP (3).

(1) Opció **Substitution Matrix (DNA/Protein)** de la pestanya **Align** del formulari d'opcions.

(2) Opció **GOP** de la pestanya **Align** del formulari d'opcions.

(3) Opció **GEP** de la pestanya **Align** del formulari d'opcions.

5.4.6 ALTRES CLASSES

ConvertImage

A l'eina implementada se li han afegit icones als botons per facilitar una mica la comprensió i fer-la més amigable per l'usuari.

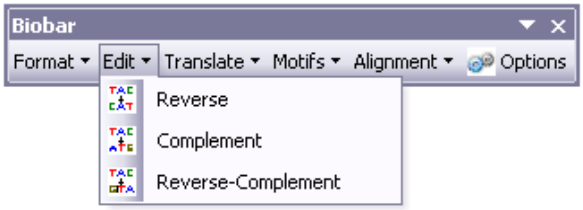


Figura 29 Mostra de les icones dels botons de l'eina.

Per poder instanciar icones als botons s'ha creat una classe *ConvertImage* amb un mètode Convert, que converteix una imatge en una icona acceptada per les propietats dels botons de la barra.

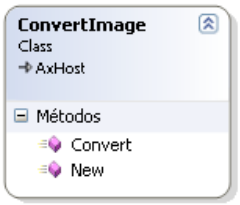
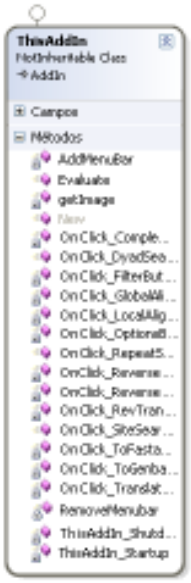


Figura 30 Diagrama de classes. Convertir imatges.

ThisAddIn



Aquesta classe és la classe principal, la que permet crea la barra de menú, afegeix els botons corresponents i conté totes les crides als diferents mètodes.

A la figura de l'esquerra es poden observar els diferents mètodes que corresponen a les crides dels botons.

Figura 31 Diagrama de classes.
Classe principal.

A continuació es mostra un diagrama de classes complet, unint el diagrama de classes bàsic inicial amb les noves classes comentades durant l'explicació de les funcionalitats:

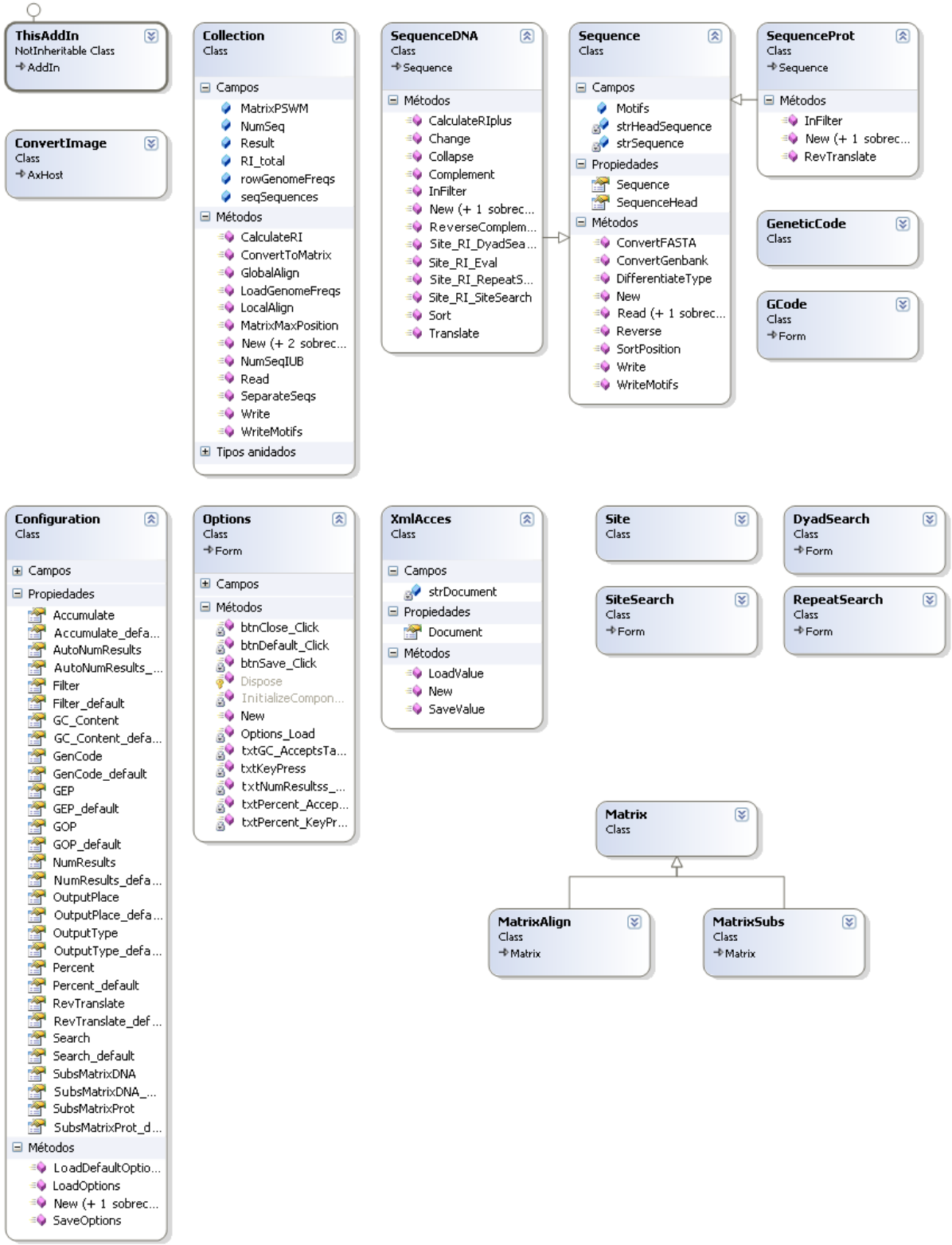


Figura 32 Diagrama de classes. Es poden observar totes les classes existents.

6 TEST FUNCIONAL

6.1 TEST BÀSIC DE FUNCIONAMENT

Durant el procés de desenvolupament de l'eina es van realitzar diferents proves bàsiques de funcionalitat i funcionament dels mètodes, per tal de detectar possibles errors a l'hora d'implementar les diferents funcionalitats.

6.2 GENERACIÓ D'UN PAQUET DE DISTRIBUCIÓ

Al finalitzar la implementació de l'eina i una vegada havent fet les proves convenients es va realitzar un paquet instal·lable per poder distribuir l'aplicació pel seu ús.

Aquesta aplicació necessita d'uns requeriments mínims de programari, com s'ha comentat anteriorment a l'apartat de requeriments.

- VSTO 2005 Runtime.
- PIA (*Primary Interop Assemblies*) o assemblats d'interoperabilitat primària.
- Microsoft Office 2003 o 2007

Alguns d'aquests requeriments, com són el VSTO 2005 i els PIA, són gratuïts i descarregables de la pàgina web de Microsoft. Aquest fet, unit a que no tenen una mida elevada (uns 5 MB entre totes dues aplicacions), va propiciar crear un paquet instal·lable amb l'eina i part dels seus requeriments bàsics, per facilitar la feina als usuaris finals.

6.2.1 TEST D'USUARIS

El paquet instal·lable creat va ser distribuït entre un grup d'usuaris per un testat més exhaustiu. Durant aquest testat es van trobar alguns errors a l'aplicació, que feien que, per alguns valors concrets d'entrada, la sortida fos errònia, i es van donar alguns suggeriments sobre algunes funcionalitats.

Aquests errors van ser solucionats i es varen tenir en compte els suggeriments aportats pels usuaris.

Un dels errors trobats es va identificar en la funcionalitat d'alineament. Aquest error consistia en que quan es feia el *trace-back* sobre la matriu d'alineament i es comparava el valor de la casella amb el de les caselles veïnes, a causa del decimals de les operacions realitzades, els valors no coincidien en cap dels casos. Aquest fet s'ha solucionat arrodonint a 1 decimal els resultats de les operacions.

El suggeriment més sol·licitat pels usuaris va ser la diferenciació entre els *dyads* i l'espaiador, l'ús de diferents colors per cada caràcter i mostrar d'alguna manera el grau de conservació de cada posició en la col·lecció, per a la cerca de motius. Els requeriments dels usuaris van ser adreçats i s'han implementat aquestes característiques, que ja han estat explicades, anteriorment, a l'apartat de recerca de motius.

7 DOCUMENTACIÓ

Per facilitar la feina a nous usuaris, s'ha decidit crear un petit manual on s'expliquen les opcions que l'usuari pot modificar segons les seves necessitats, les diferents funcionalitats existents a l'eina i què és el que es necessita com a entrada i s'obté com a sortida en cada un dels casos. Aquest manual es pot trobar a l'annex de la memòria.

A més de la memòria i del manual d'usuari, cal d'estacar l'ús d'una eina de documentació de codi. Aquesta eina té la particularitat que afegeix comentaris en XML amb el mateix format i camps, fet que facilita la posterior comprensió del codi i les possibles ampliacions futures per part de qualsevol usuari que ho desitgi. Un exemple del format de documentació del codi es mostra a continuació:

```
''' <summary>
''' </summary>
''' <param name="paramA"></param>
''' <param name="paramB"></param>
''' <returns></returns>
```

El tag `<summary>` conté una breu explicació de la funcionalitat de la funció o procediment. En el cas de tenir paràmetres, aquests van al tag `<param>` amb el nom corresponent a cada un d'ells. Si el mètode és una funció existeix el tag `<returns>` que conté una breu explicació de quin és el valor retornat i el se tipus.

Les classes o propietats només contenen el tag `<summary>`.

8 CONCLUSIONS I VALORACIÓ PERSONAL

L'objectiu d'aquest projecte, tal i com indica el seu nom, era la implementació d'un *add-in* d'anàlisi i manipulació de seqüències de DNA i proteïna integrat en un processador de textos com és Microsoft Word. Les característiques principals que es requerien eren les següents:

- Havia de ser simple i de fàcil ús pels usuaris.
- Havia de permetre la manipulació de seqüències de DNA i de proteïna.
- Havia de ser integrable en un processador de textos (Microsoft Word).

L'eina obtinguda compleix amb tots aquests requeriments, gràcies a la tecnologia emprada per implementar-la. Per tant, es pot afirmar que s'ha acomplert el previst inicialment, ja que els resultats obtinguts inclouen totes les especificacions inicials i han estat validats i contrastats amb els usuaris finals.

Per part dels usuaris, la implementació d'aquesta eina ha tingut una gran acceptació i esperem que no hagi defraudat, en cap cas, el resultat obtingut amb el que esperaven.

En les enquestes realitzades als usuaris, es van poder observar algunes funcionalitats i característiques no implementades en el transcurs d'aquest projecte. L'existència de noves funcionalitats desitjades pels usuaris, juntament amb la modularitat i facilitat de programació de l'eina, fa que siguin viables noves versions de l'aplicació amb aquestes característiques. Aquestes versions amb noves funcionalitats podrien ser programades pels propis usuaris, en cas de tenir uns coneixements de programació (i.e. bioinformàtics).

Quan vaig començar aquest projecte tenia la intenció de crear una eina útil i que fos utilitzada per usuaris, ja que no volia realitzar una eina que acabés al fons d'un calaix. Arribats a aquest punt, considero que la feina feta ha estat satisfactòria i que l'eina serà utilitzada per una comunitat d'usuaris (major o menor ho dirà el temps).

ANNEX A: ENQUESTA

Nombre: (anónimo por defecto)

Encuesta:

1.- ¿Qué procesador de textos usa habitualmente?

(marque con una X o especifique)

Ms Word	**Ms Works**	**Lotus Word**	**TextEdit**	**WordPerfect**	**OpenOffice Writer**	**StarOffice Writer**

Otro processador de textos no listado arriba: .....

2.- ¿Le sería útil una herramienta de tratamiento y edición de secuencias de DNA y proteína integrada en su procesador de textos?

(marque con una X)

Nada	**Poco**	**Bastante**	**Mucho**

3.- Describa y valore las opciones deseadas para dicha herramienta:

Opciones:	**Utilidad (1-5)**

4.- ¿Añadiría otras funcionalidades que no considerara tan prioritarias a la herramienta? ¿Cuáles?

ANNEX B: MANUAL D'USUARI DE BIOBAR.

CARACTERÍSTIQUES BÀSIQUES DE LA SEVA FUNCIONALITAT I OPCIONS

OPCIONS

Basic Options: Opcions bàsiques del programa.

- **Filter**: Filtra tots els caràcters incorrectes de la seqüència d'entrada.
  Pel cas de DNA existeixen dues opcions:
  - *Strict*: `"A", "T", "C", "G"`
  - *IUB*: `"A", "T", "C", "G", "R", "Y", "M", "K", "S", "W", "H", "B", "V", "D", "N"`
- **%:** Percentatge de bases estrictes a la seqüència d'entrada a partir del qual es discerneix entre seqüència de DNA i proteïna.
- **Output Place**: Determina el lloc de sortida del resultat.
  - *Replace*: Reemplaça la seqüència d'entrada per la de sortida.
  - *Clipboard*: Guarda el resultat al clipboard.
  - *New Document*: Crea un nou document i hi guarda el resultat.
  - *Below*: Retorna el resultat al final del document actual.
- **Output Type**: Determina el format de sortida de la seqüència.
  - *Plain*: Text pla sense format.
  - *FASTA*: Retorna el resultat com una seqüència en format FASTA.
  - *Genbank*: Retorna el resultat com una seqüència en format Genbank.
- **Accumulate**: Opció booleana. Si està marcada la casella i no hi ha text seleccionat com a entrada comprovarà si hi alguna possible entrada al clipboard.
  Facilita l'acumulació de tasques si els seus resultats són enviats al clipboard.

GCode Options: Opcions relacionades amb els codis genètics.

- **Genetic Code**: Selecciona el codi genètic que s'utilitzarà com a opció.
  Les opcions existents són: Standard, Vertebrate Mitochondrial, Yeast Mitochondrial, ...
- **Rev. Translate**: Determina com es realitzarà la traducció inversa.
  - *Best*: Al carregar el codi genètic es demana a l'usuari que introdueixi el *codon usage* per determina el millor codó per cada aminoàcid.

- *Ambiguous*: No demana a l'usuari cap *codon usage* i retorna la traducció inversa com a codi *IUB*.

Sites Options: Opcions relacionades amb la cerca de llocs.

- **GC Content**: GC Content for Background entropy.
- **# Results**: Nombre de resultats (llocs trobats) que es retornaran.
- **Automatic # Results**: Determina si es desitja que l'eina faci el càlcul automàtic del nombre de resultats a retornar.
- **Search**: Determina de quin tipus serà la *Repeat Search*. A més, s'utilitza per determinar el segon *dyad* a la *Dyad Search*.
    - *Palindrome*: El segon dyad de la cerca serà el palindròmic (revers-complement) del primer.
    - *Direct repeat*: El segon dyad de la cerca serà el mateix que el primer.

Align Options: Opcions relacionades amb l'alineament de seqüències.

- **Substitution Matrixs**: Possibles matrius de substitució a escollir per l'alineament.
    - *DNA*: Matriu de substitució per a l'alineament de seqüències de DNA.
    - *Protein*: Matriu de substitució per a l'alineament de seqüències de proteïna.
- **GOP**: Gap Opening Penalty. Cost associat a l'obertura d'un *gap.*
- **GEP**: Gap Extension Penalty. Cost associat a posicions adjacents al *gap* original.

Entrada/Sortida

Totes les seqüències d'entrada aniran separades per dos retorns de carro, excepte en el cas que estiguin en format FASTA.

Com a lloc de sortida en el cas de cerca de llocs només és tindran en compte 2 opcions, *Below* (que contindrà també *Replace*) i *New Document* (que englobarà també a *Clipboard*).

Funcionalitats

Format

FILTER

Aquesta opció pot rebre tant seqüències de DNA com de proteïnes.

En el cas de seqüències de DNA la resposta obtinguda variarà segons les opcions escollides. Podem obtenir com a resultat la seqüència o seqüències filtrades de forma estricta (A, C, G, T) o filtrades en format *IUB*.

Pel cas de seqüències de proteïnes,no tindrà en compte cap opció, només filtrarà les seqüències d'elements erronis.

Entrada: Una o varies seqüències de DNA o proteïnes.

Sortida: Una o varies seqüències de DNA o proteïnes filtrades.

TO FASTA

Converteix l'entrada en seqüències amb format FASTA. Si la seqüència no disposava de capçalera crea una.

Entrada: Una o varies seqüències de DNA o proteïnes .

Sortida: Una o varies seqüències de DNA o proteïnes en format FASTA.

TO GENBANK

Converteix l'entrada en seqüències amb format Genbank.

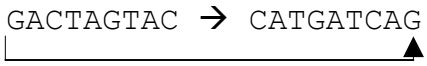
Entrada: Una o varies seqüències de DNA o proteïnes .

Sortida: Una o varies seqüències de DNA o proteïnes en format Genbank.

Edit

REVERSE

La funció inverteix l'ordre de les seqüències d'entrada per obtenir la sortida.

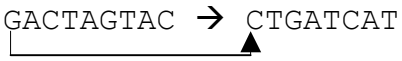


Entrada: Una o varies seqüències de DNA o proteïnes .

Sortida: Una o varies seqüències de DNA o proteïnes.

COMPLEMENT

La funció realitza el complement de les seqüències d'entrada per obtenir la sortida.

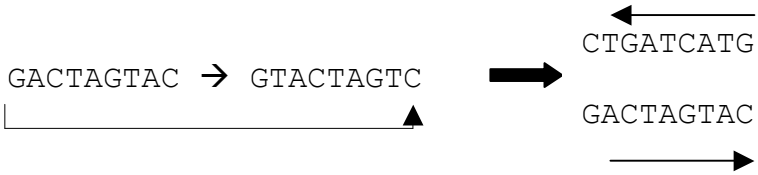


Entrada: Una o varies seqüències de DNA.

Sortida: Una o varies seqüències de DNA.

REVERSE-COMPLEMENT

La funció inverteix l'ordre de les seqüències i després realitza el seu complement per obtenir la sortida.



Entrada: Una o varies seqüències de DNA.

Sortida: Una o varies seqüències de DNA.

Translate

TRANSLATE

Converteix les seqüències de DNA d'entrada en seqüències de proteïnes.

Entrada: Una o varies seqüències de DNA.
Sortida: Una o varies seqüències de DNA.

REVERS TRANSLATE

Converteix les seqüències de proteïnes d'entrada en seqüències de DNA. Existeixen dues opcions, convertir cada AA en el millor codó possible segons un codi genètic facilitat per l'usuari, o convertir-lo en el codó ambigu corresponent.

Entrada: Una o varies seqüències de proteïna.
Sortida: Una o varies seqüències de proteïna.

Motifs

SITE SEARCH

Cerca de llocs a partir d'una col·lecció de seqüències o d'una seqüència *IUB*.

Entrada: Una o varies seqüències de DNA.
Sortida: Llistat de motius trobats (quantitat segons opcions), amb la seva posició i el seu valor.

Paràmetres: Col·lecció de motius o seqüència *IUB*.

DYAD SEARCH

Cerca de llocs *dyad-spacer-dyad* a partir de una seqüència IUB de cada *dyad.*

Entrada: Una o varies seqüències de DNA.
Sortida: Llistat de llocs trobats (quantitat segons opcions), amb la seva posició i el seu valor.

Paràmetres: Dyads i longitud màxima i mínima de l'espaiador dels motius.

REPEAT SEARCH

Cerca de llocs *dyad-spacer-dyad* lliure, acotada per nombre de *mismatchs* entre *dyads*.

Entrada: Una o varies seqüències de DNA.
Sortida: Llistat de motius trobats (quantitat segons opcions), amb la seva posició i el seu valor.

Paràmetres: Longitud màxima i mínima dels dyads i de l'espaiador dels motius.

Alignment

GLOBAL ALIGNMENT

Obté l'alineament global de dues seqüències de DNA o proteïna, segons els valors del GOP i GEP, i la matriu d'alineament, escollits per l'usuari a les opcions.

Entrada: Dues seqüències de DNA o proteïna.
Sortida: Dues seqüències de DNA o proteïna alineades segons la matriu de substitució escollida a les opcions.

LOCAL ALIGNMENT

Obté l'alineament local de dues seqüències de DNA o proteïna, segons els valors del GOP i GEP, i la matriu d'alineament, escollits per l'usuari a les opcions.

Entrada: Dues seqüències de DNA o proteïna.
Sortida: Dues seqüències de DNA o proteïna alineades segons la matriu de substitució escollida a les opcions.

BIBLIOGRAFIA

Baldi, P. and S. Brunak (2001). Bioinformatics - The Machine Learning Approach. Cambridge, MIT Press.

Berg, O. G. and P. H. von Hippel (1988). "Selection of DNA binding sites by regulatory proteins." Trends Biochem Sci **13**(6): 207-11.

Bruney, A. (2006). Professional VSTO 2005: Visual Studio 2005 Tools for Office. Indianapolis, Wiley Publishing, Inc.

Campbell, A. M. and L. J. Heyer (2007). Discovering Genomics, Proteomics And Bioinformatics, Pearson/Benjamin Cummings.

Dale, J. and S. F. Park (2004). Molecular Genetics of Bacteria, John Wiley and Sons.

Dayhoff, M., R. Schwartz, et al. (1978). "A model of evolutionary change in proteins." Atlas of protein sequence and structure. **5**: 345-352.

Henikoff, S. and J. G. Henikoff (1992). "Amino acid substitution matrices from protein blocks." Proc Natl Acad Sci U S A **89**(22): 10915-9.

Lander, E. S., L. M. Linton, et al. (2001). "Initial sequencing and analysis of the human genome." Nature **409**(6822): 860-921.

Lesk, A. M. (2002). Introduction to Bioinformatics. New York, Oxford University Press Inc.

McGrath, K. and P. Stubbs (2006). VSTO for Mere Mortals.

MSDN-Blogs. "Microsoft Visual Studio Tools for the Microsoft Office System." from http://blogs.msdn.com/vsto2/.

MSDN. "Visual Studio Tools para Office." from http://msdn2.microsoft.com/es-es/library/d2tx7z6d%28VS.80%29.aspx.

Needleman, S. B. and C. D. Wunsch (1970). "A general method applicable to the search for similarities in the amino acid sequence of two proteins." J Mol Biol **48**(3): 443-53.

O'Neill, M. C. (1991). "Training back-propagation neural networks to define and detect DNA-binding sites." Nucleic Acids Res **19**(2): 313-8.

Smith, T. F. and M. S. Waterman (1981). "Identification of common molecular subsequences." J Mol Biol **147**(1): 195-7.

RESUM

L'objectiu del projecte consisteix en el desenvolupament d'un *add-in* d'anàlisi i manipulació de seqüències, senzill i de fàcil ús, integrable en l'entorn Microsoft Word per permetre la manipulació de seqüències genètiques directament des de Microsoft Word, estalviant temps, al evitar tenir que canviar constantment de programa i format per treballar amb elles, i, també, complicacions a l'usuari final. L'*add-in* ha estat desenvolupat en Visual Basic + VSTO i ofereix diverses funcionalitats d'edició i anàlisi de seqüències, com ara el complement, la recerca de motius o l'alineament.

RESUMEN

El objetivo del proyecto consiste en el desarrollo de un *add-in* de análisis y manipulación de secuencias, sencillo y de fácil uso, integrable en el entorno Microsoft Word para permitir la manipulación de secuencias genéticas directamente desde Microsoft Word, ahorrando tiempo, al evitar tener que cambiar constantemente de programa y formato para trabajar con ellas, y, también, complicaciones al usuario final. El *add-in* ha sido desarrollado en Visual Basic + VSTO y ofrece diversas funcionalidades de edición y análisis de secuencias, como el complemento, la búsqueda de motivos o el alineamiento.

ABSTRACT

The objective of this project is the development of an add-in for sequence analysis and manipulation, simple and easy to use and integrable within the Microsoft Word environment, to allow manipulation of genetic sequences directly from Microsoft Word, saving time, by avoiding to constant program and format changes, and other nuisances to the final user. The add-in has been developed in Visual Basic + VSTO and offers diverse functionalities of sequence edition and analysis, such as the sequence complement, motif search or sequence alignment.