TESI DI LAUREA

Ouroboros: un metodo per il riconoscimento e l'analisi delle proteine solenoidi

Laureando: **Umberto Dell'Aica**

Relatore: Ch.mo Prof. **Carlo Ferrari**

Dipartimento di Ingegneria dell'Informazione

Correlatore: Ch.mo Prof. **Silvio C.E. Tosatto**

Dipartimento di Biologia

Corso di Laurea Specialistica

in Bioingegneria

Data di laurea: 7 Dicembre 2010

Anno Accademico 2010-2011

A mio padre e mia madre

<<There are more things in heaven and earth, Horatio,
Than are dreamt of in your philosophy.>>

Hamlet, Act 1, scene 5, 159–167

Sommario

CAPITOLO I

Proteine

Le proteine sono composti organici complessi, fondamentali costituenti di tutte le cellule animali e vegetali. Dal punto di vista strutturale sono macromolecole o polimeri, costituite da una catena di residui amminoacidici, e sono generate dalla traduzione di sequenze di nucleotidi contenute nel genoma di un organismo. Fisiologicamente, l'operazione di trascrizione di queste informazioni e la traduzione in una sequenza proteica completa è un procedimento complesso, e omettiamo questa descrizione. E' però di fondamentale importanza capire i soggetti di questo processo: si può considerare il DNA, l'acido deossiribonucleico, che è anch'essa una macromolecola (acido nucleico i cui monomeri sono chiamati nucleotidi) come il custode delle informazioni genetiche, e gli amminoacidi come il prodotto della traduzione di queste informazioni. Gli amminoacidi, strutturati, costituiscono quindi una proteina, e la semplice sequenza specifica degli amminoacidi costituisce la struttura primaria di una proteina.

Le proteine assolvono funzioni di tipo strutturale, immunitario, di trasporto per esempio di ossigeno, di identificazione dell'identità genetica, oppure hanno funzione ormonale, enzimatica, contrattile, energetica. Ciascuna funzione attribuibile ad una proteina è strettamente dipendente dalla sua struttura tridimensionale, ovvero spaziale. La conformazione spaziale di una proteina è fondamentale affinché questa esplichi la sua attività biologica: distruggerla, rompendo i legami idrogeno e ponti disolfuro per mezzo di acidi, basi, calore, radiazioni o agitazione significa denaturare una proteina.

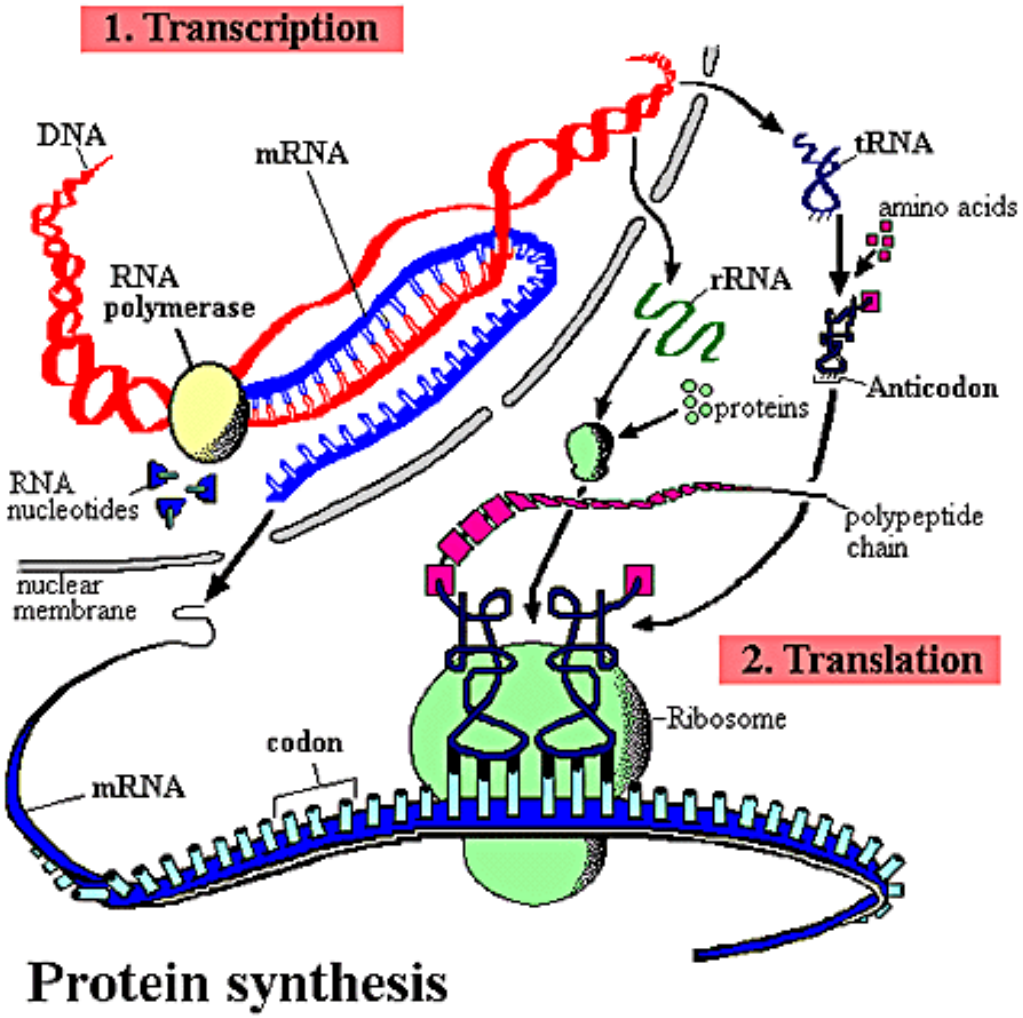


Figura 1: Sintesi proteica mediante trascrizione e traduzione in proteine

Una proteina denaturata, pur mantenendo intatta la sua struttura primaria, non è più in grado di esplicare la sua funzione. Soprattutto per questo motivo, la conoscenza della struttura tridimensionale di una proteina è di capitale importanza per comprendere ed anche intervenire nelle funzioni proteiche, siano queste azioni solamente a scopo informativo oppure, per esempio, di ausilio alla sintesi di farmaci, o alla progettazione di nuove proteine per le più disparate esigenze.

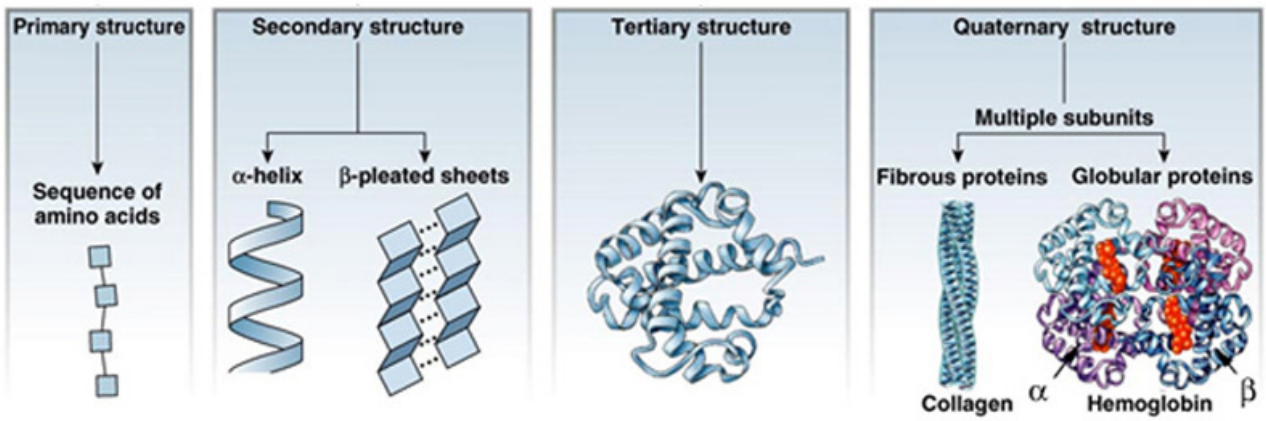


Figura 2 : Le quattro strutture per le proteine

1.1 Struttura delle proteine

Nel paragrafo precedente è stata introdotta la prima descrizione strutturale proteica: la sequenza di amminoacidi o residui costituisce la struttura primaria di una proteina. Esistono altri tre livelli strutturali riconosciuti: la struttura secondaria, ossia la conformazione ordinata che alcuni tratti di proteina possono assumere, e la struttura terziaria, o una disposizione tridimensionale. Infine, vi è la struttura quaternaria, che prevede l'aggregazione di più macromolecole.

1.2 Struttura primaria

La struttura primaria (o covalente, dalla natura dei legami peptidici che la caratterizzano) è la semplice sequenza amminoacidica. Essa non descrive la struttura biologicamente attiva della proteina, ma determina tutte le proprietà chimiche della macromolecola e contiene al suo interno le informazioni sufficienti a definire gli ordini di struttura superiori (struttura secondaria, terziaria ed eventualmente quaternaria) che le conferiscono la propria conformazione funzionalmente attiva. Gli amminoacidi esistenti in natura sono 20, essi possono essere combinati in qualsiasi configurazione,

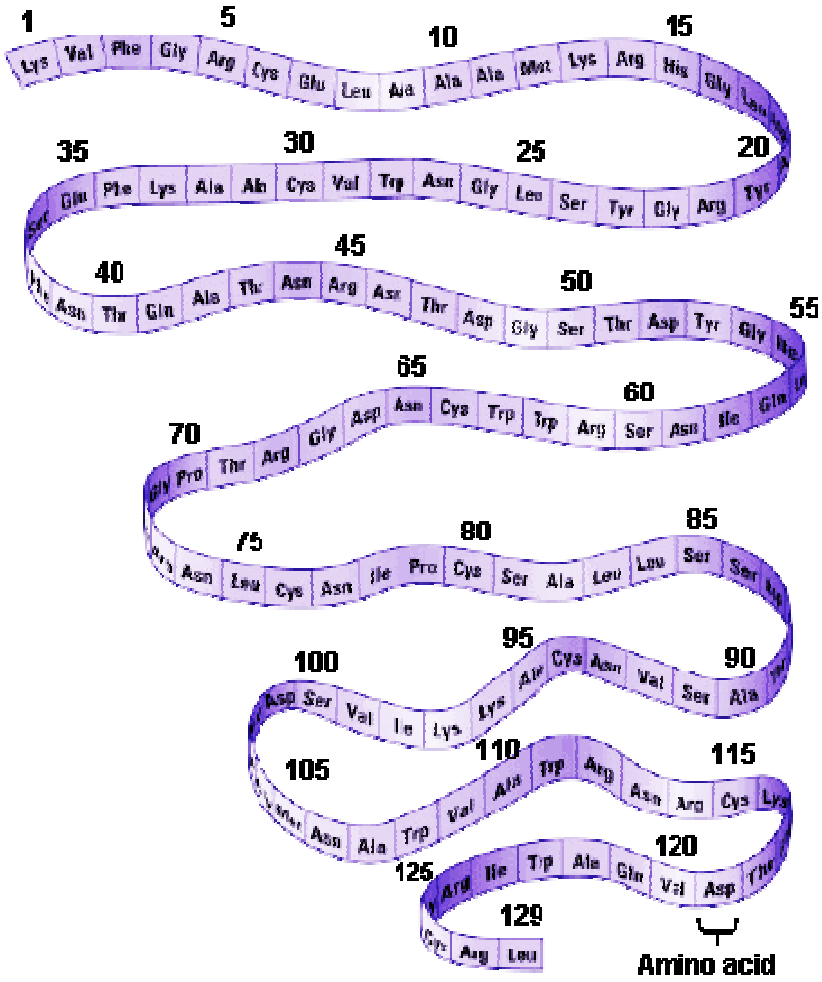


Figura 3: La struttura primaria

dando origine alle diverse strutture proteiche. Descrivendo brevemente un amminoacido, possiamo innanzitutto dire che esso è composto da due parti, catena principale (comune a ogni residuo) e catena laterale, caratterizzante. La lunghezza dei legami e gli angoli di legame che le catene generano differenti geometrie di disposizione degli atomi.

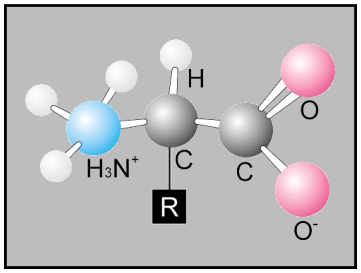


Figura 4 : Struttura generica di un amminoacido

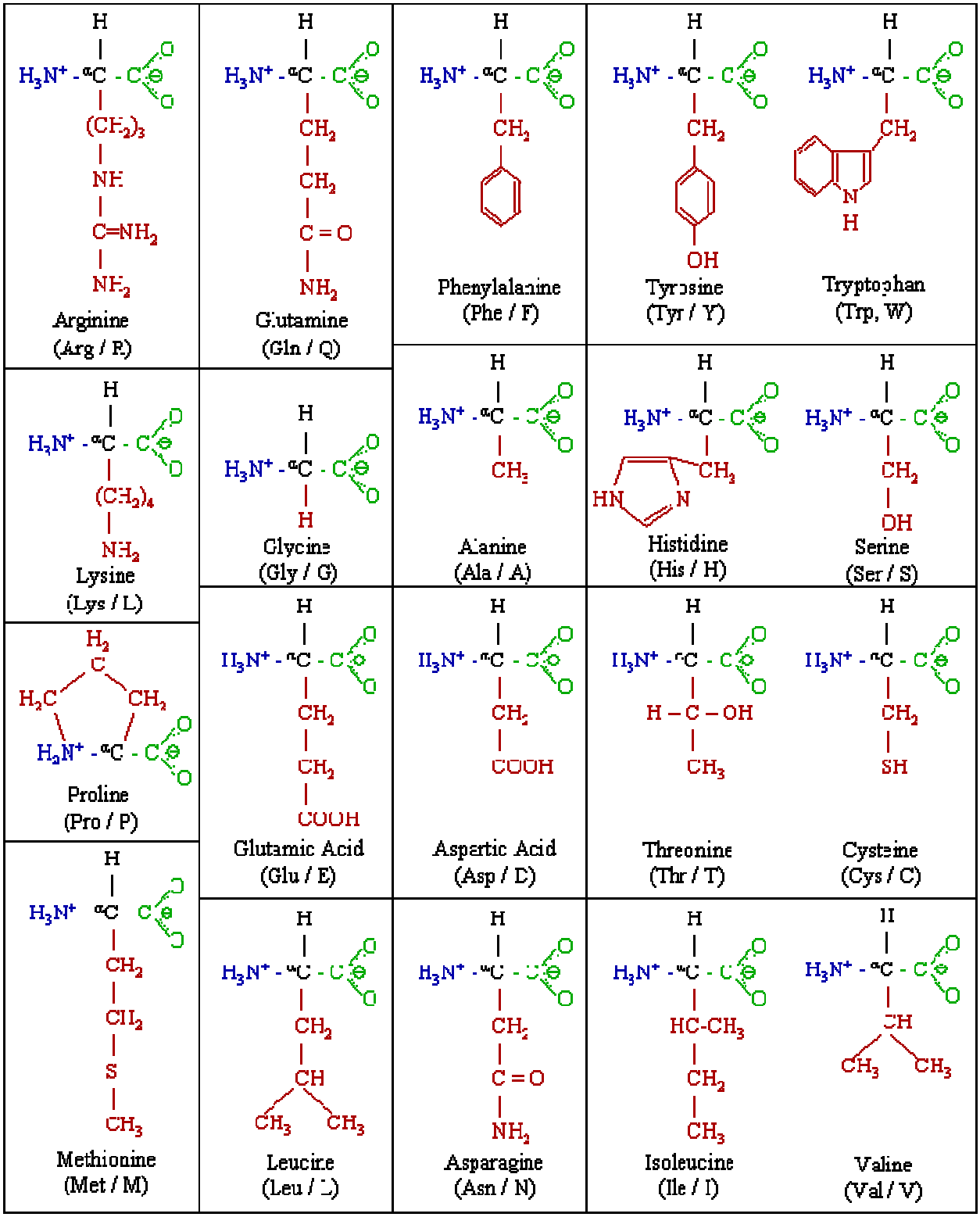


Figura 5: I 20 amminoacidi

1.3 Struttura secondaria

La struttura secondaria di una proteina, è la capacità di una proteina di assumere una struttura spaziale regolare e ripetitiva, può essere di tre diversi tipi: α-elica (alfa elica), struttura β (struttura beta) e loops (o ripiegamenti). Queste tre strutture posseggono delle topologie con geometrie ben definite e fisse nel tempo (nel senso che non variano e sono visibili ai raggi X). La struttura secondaria è determinata da interazioni a corto raggio tra residui amminoacidici della catena polipeptidica.

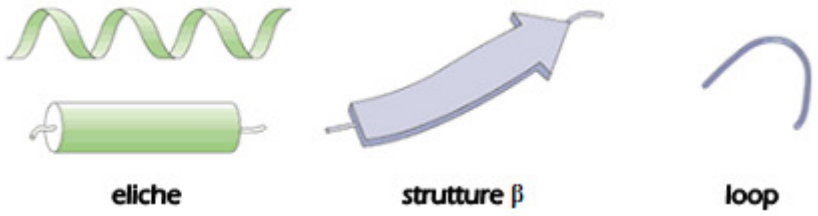


Figura 6: rappresentazioni per alfa eliche, beta strand e loops

I due fattori fondamentali che intervengono nella creazione della struttura secondaria sono:

- la minimizzazione dell'ingombro sterico fra i gruppi R condizionata dalle restrizioni imposte alle coppie di angoli diedri di due piani peptidici consecutivi.
- l'ottimizzazione dei ponti idrogeno infra-catena fra residui non adiacenti.

Nel seguito analizziamo brevemente i tre tipi di struttura sunnominati, dandone una descrizione atta a comprendere i nostri scopi.

Alfa eliche

L'alfa elica è una struttura elicoidale, la più semplice tra le disposizioni strutturali secondarie delle proteine. Si formano quando un numero di amminoacidi susseguenti presentano angoli di legame (angoli φ e ψ) compresi tra -60° e -45°, collocandosi nell'angolo inferiore sinistro di un grafico di Ramachandran. Ogni giro completo dell'elica corrisponde ad una distanza di 5,4 [Å] (Angstrom) lungo l'asse immaginario, il che implica che vengano coinvolti circa 4 amminoacidi ogni giro. Le alfa

eliche possono presentarsi sia in configurazione destrorsa che sinistrorsa, anche se le prime sono più comuni. Il legame chimico principale che interviene nella formazione di strutture secondarie è il ponte ad idrogeno, garante della stabilità dell'alfa elica: ogni giro d'elica è infatti unito a quelli adiacenti da 3-4 legami idrogeno.

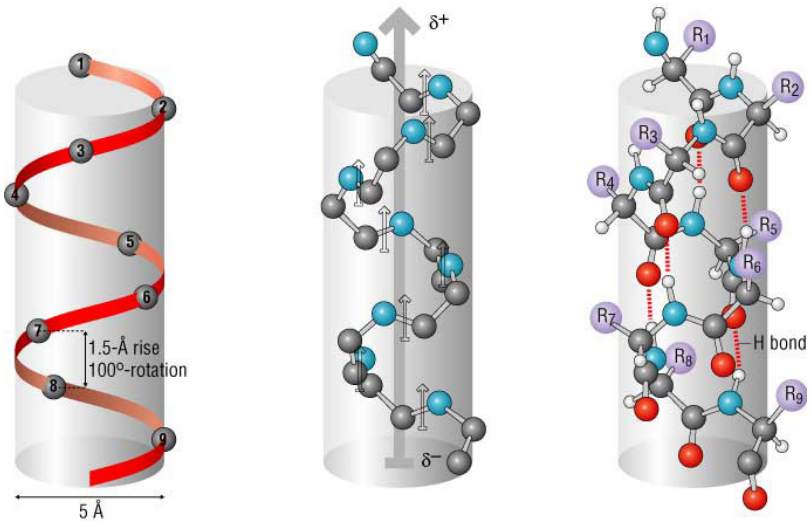


Figura 7: La struttura secondaria ad alfa elica: è evidenziato l'avvolgimento assieme alle interazioni presenti tra le molecole

Alcuni amminoacidi (alanina, acido glutammico, leucina e metionina), infine, partecipano attivamente e con particolare affinità a formare α-eliche, mentre altri (glicina, prolina, tirosina e serina) sono praticamente assenti o poco incidenti.

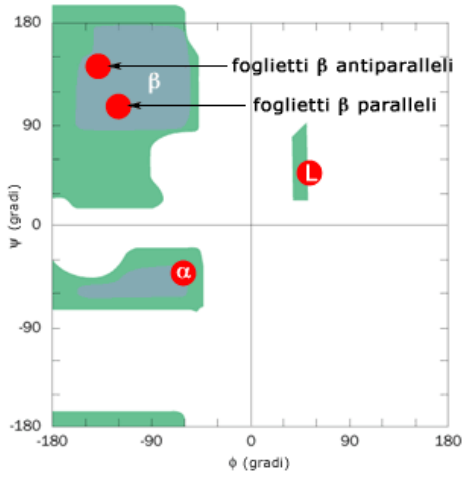


Figura 8: Grafico di Ramachandran. Le zone in verde indicano indicano gli angoli ϕ e ψ stericamente consentiti.

Strutture Beta

Anche riportate come *β-sheet,* è la seconda forma strutturale più diffusa per la struttura secondaria delle proteine, ed è generata dall'affiancamento di più filamenti β, collegati fra loro tramite solitamente 3 o più legami ad idrogeno. Il filamento beta (o beta strand) è una sequenza di 5-10 residui lineare, in grado di favorire la comparsa di ponti ad idrogeno, in modo tale da rendere la struttura beta una conformazione complessivamente planare (anche se leggermente incurvata) molto compatta. Per quanto riguarda la non perfetta planarità, gli angoli torsionali, $(\varphi, \psi) = (-135^{o}, 135^{o})$, corrispondenti grossomodo alla regione in alto a sinistra del grafico di Ramachandran, sono differenti da quelli che caratterizzerebbero una conformazione completamente planare, $(\varphi, \psi) = (-180^{o}, 180^{o})$. La disposizione dei filamenti beta può avvenire in senso parallelo, antiparallelo o misto.

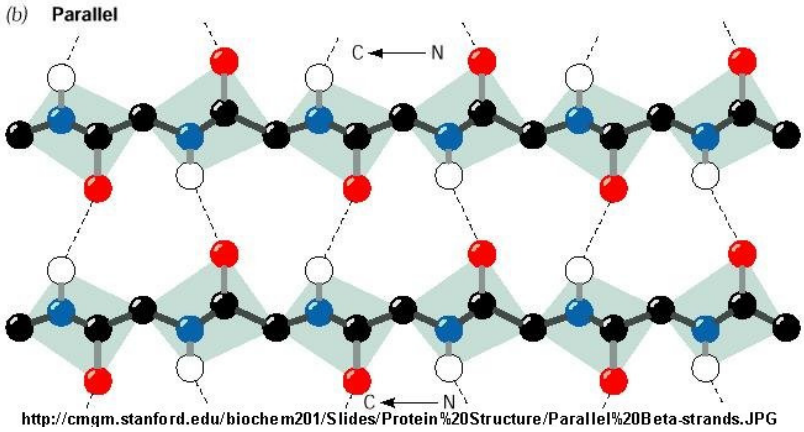


Figura 9: Beta strands paralleli qui mostrati in rappresentazione Ball-and-Stick

Loops

Oltre alle due regolari strutture appena introdotte, nelle proteine sono presenti tratti di catena apparentemente disorganizzati, di lunghezza molto variabile e più o meno convoluti. Questi tratti, definiti loop, collegano eliche e filamenti-β ed hanno un ruolo importante nell'organizzazione 3D della macromolecola. Sono infatti relativamente flessibili e consentono cambi di direzione, anche repentini, alle sequenze a conformazione α e β. Generalmente i loop si trovano esposti sulla superficie della proteina e i residui che ne fanno parte non formano ponti idrogeno fra loro ma con molecole del solvente. In queste regioni è costante la presenza della glicina e della prolina.

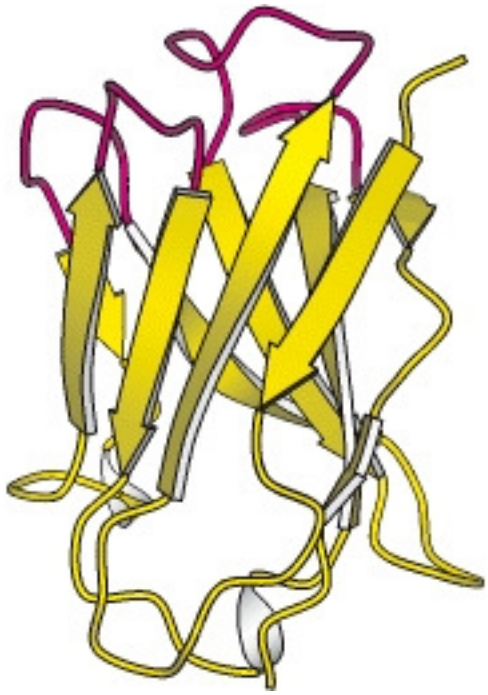


Figura 10: Raffigurazionedi una proteina composta da Beta Strands e da Loops

Motivi strutturali

I motivi strutturali più ricorrenti sono:

- elica-loop-elica
- β-turn: due filamenti-β consecutivi orientati in modo antiparallelo e collegati da un breve loop di 2-5 residui.
- chiave greca: quattro filamenti-β, due brevi loop e un loop più lungo. La caratteristica del motivo a chiave greca è il diverso ordine dei filamenti-β antiparalleli componenti la struttura rispetto alla posizione nella catena peptidica.
- β-α-β: due filamenti-β paralleli intercalati da un'α-elica. L'asse dell'elica è parallelo a quello dei filamenti-β. I due loop di collegamento possono avere lunghezze molto variabili (da 2 fino a 100 residui) e funzioni specifiche diverse. Nella quasi totalità delle proteine conosciute, gli SSE si combinano in particolari motivi strutturali di struttura super-secondaria. Spesso è anche possibile associare alcuni motivi strutturali, o più propriamente la loro organizzazione in domini, a particolari funzioni di una proteina.

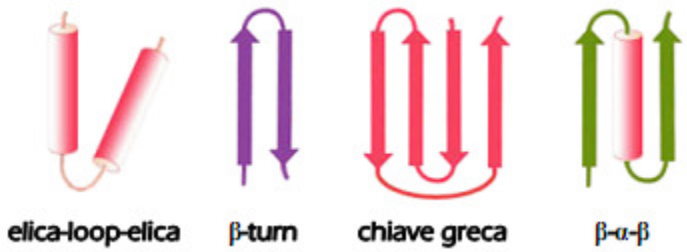


Figura 11: i motivi strutturali principali

1.4 Struttura terziaria

La struttura terziaria è la corretta conformazione tridimensionale assunta da una proteina ed è indispensabile per la sua attività biologica. La struttura terziaria è stabilizzata principalmente da legami non covalenti come ponti idrogeno, interazioni idrofobiche tra amminoacidi non polari e legami ionici. Inoltre, la struttura terziaria può coinvolgere anche legami covalenti, sotto forma di ponti disolfuro fra due cisteine. Le interazioni che si instaurano a livello tridimensionale coinvolgono amminoacidi non necessariamente vicini nella struttura primaria.

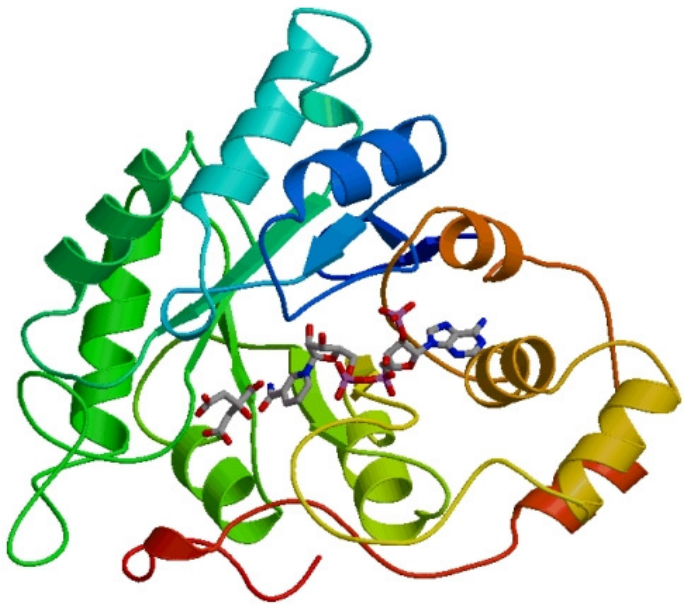


Figura 12: raffigurazione di una proteina in completa struttura terziaria

Quando queste interazioni vengono meno, per esempio in presenza di elevate temperature, di pH non ottimale o di detergenti, la struttura tridimensionale viene persa, e la proteina va incontro a denaturazione, perdendo la sua attività biologica. In molti casi la denaturazione è un processo reversibile, e, allontanando l'agente denaturante, la proteina riprende spontaneamente la sua conformazione tridimensionale (che è dettata dalla struttura primaria). Ogni proteina tende ad assumere una sola struttura terziaria e proteine differenti assumono conformazioni differenti. Talvolta le proteine possono assumere anche una struttura quaternaria.

1.5 Struttura quaternaria

Quando una proteina è costituita da due o più polipeptidi ha una struttura detta quaternaria. Molte proteine con funzione regolatoria sono costituite da più catene polipeptidiche, che consentono di svolgere al meglio il proprio compito. In altri casi invece, l'associazione di più polipeptidi è presente in proteine con funzione strutturale, come ad esempio nel capside che riveste i virus. L'associazione fra le diverse catene polipeptidiche può essere covalente, cioè mediata da ponti disolfuro, come negli anticorpi o non covalente, come nelle proteine G eterotrimeriche associate ai GPCR. Una proteina costituita da molte subunità polipeptidiche è detta multimero. Un multimero con poche subunità viene definito oligomero.

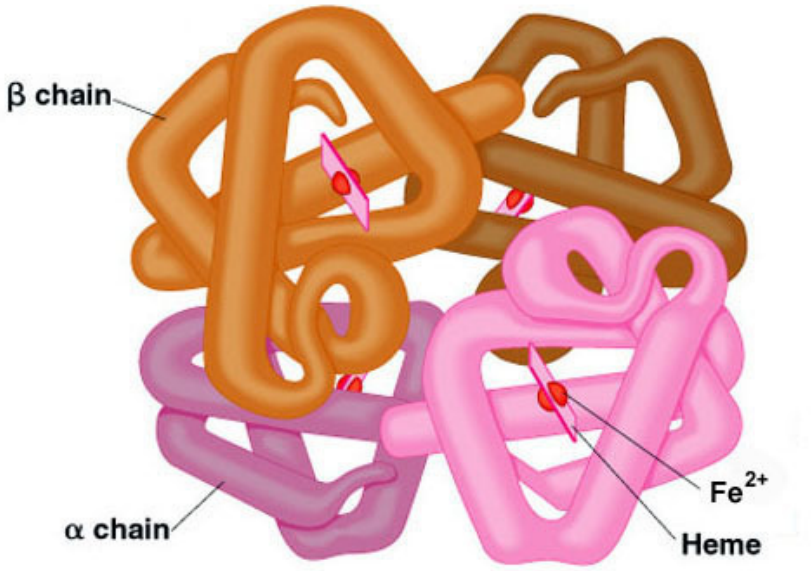


Figura 13: la struttura quaternaria dell'emoglobina

La prima struttura oligomerica studiata è stata l'emoglobina (Hb), la proteina che trasporta l'ossigeno dai polmoni ai tessuti e la CO_2, formatasi in seguito alla respirazione cellulare, nel percorso inverso. L'emoglobina di un individuo adulto è costituita da quattro catene polipeptidiche che coordinano ciascuna un gruppo prostetico detto eme, che contiene un atomo di ferro ed è indispensabile per la funzione di questa proteina. Le quattro catene sono organizzate in due coppie simmetriche.

CAPITOLO II
Le proteine solenoidi

Le proteine solenoidi o ripetute sono una classe di proteine caratterizzate da ripetizione sia nella sequenza (struttura primaria) che nel fold (struttura terziaria). Nei genomi eucarioti e procarioti sono state identificate grandi quantità di sequenze di DNA ripetute, ed è provato che esse siano coinvolte in almeno cinque malattie neurodegenerative (malattia di Huntington, atassia spinocerebellare di tipo 1, atrofia di Dentatorubral-pallidoluysian, malattia di Macado-Joseph e Atrofia muscolare bulbo-spinale. Dal punto di vista evolutivo, è interessante visionare alcune caratteristiche delle proteine ripetute. È noto che le ripetizioni nella sequenza proteica sono dovute ad un errore nel processo di duplicazione, che avviene con probabilità maggiore rispetto a quello delle normali mutazioni. Ciò suggerisce una maggiore velocità di evoluzione per le proteine ripetute, anche se solitamente la loro presenza si registra in regioni del genoma non codificanti.

Le ripetizioni delle proteine variano da un singolo aminoacido alle ripetizioni di domini composti da 100 o più residui, con strutture e funzioni eterogenee. Recenti analisi sulle proteine ripetute hanno evidenziato la loro presenza principalmente in organismi eucarioti, rispetto agli organismi procarioti, e presentano basse similarità con le proteine degli organismi più antichi. Le proteine solenoidi, quindi, sembrano essere un meccanismo di tipo evolutivo relativamente recente.

Metodi come la cristallografia a raggi X e la risonanza magnetica nucleare (NMR) non riescono a fornire strutture esaustive per la descrizione di queste macromolecole. Per questo motivo è necessario affidarsi a metodi predittivi, cercando cioè teoricamente, partendo dalla struttura primaria, di descrivere la struttura tridimensionale delle molecole.

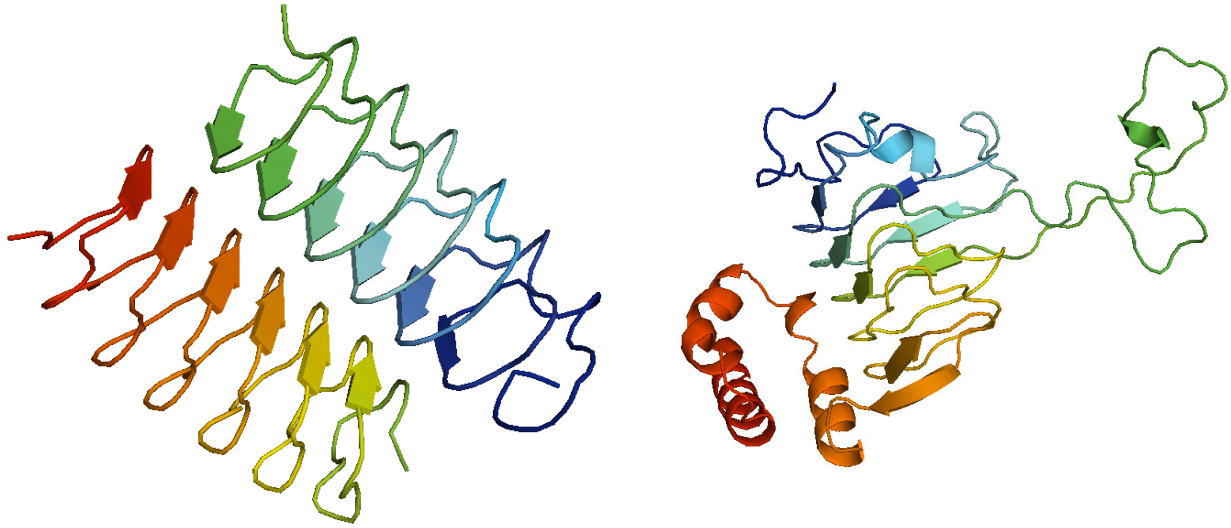


Figura 14: due esempi di proteine solenoidi. La prima, a sinistra, presenta una ripetizione costante, senza inserzioni. A destra, esempio di una proteina ripetuta con grande presenza di inserzioni

2.1 La struttura delle proteine con ripetizioni

La prima domanda che ci si pone quando si analizza una proteina è se sia possibile o meno predirre la struttura 3D che la caratterizza, partendo da alcune semplici informazioni. La maggior parte delle proteine solitamente si ripiega in una struttura univoca, che cambia in dipendenza della sequenza di amminoacidi di cui sono composte. Ciò si verifica particolarmente per proteine con sequenze non periodiche, che si ripiegano in strutture stabili, mentre le zone disordinate sono localizzate solamente nei loops o collegamenti tra domini. Per le proteine ripetute, l'unicità della struttura non è sempre una condizione verificata. Ad esempio, studi sperimentali non sono riusciti a rintracciare una struttura 3D unica per l'elastina. La particolarità che abbiamo appena introdotto è dovuta alle ripetizioni, che formano una struttura elastica e flessibile. Nonostante questo, le proteine solenoidi interagiscono con altre proteine con ripetizioni, poiché esse presentano stabilità solo nel sito di interazione.

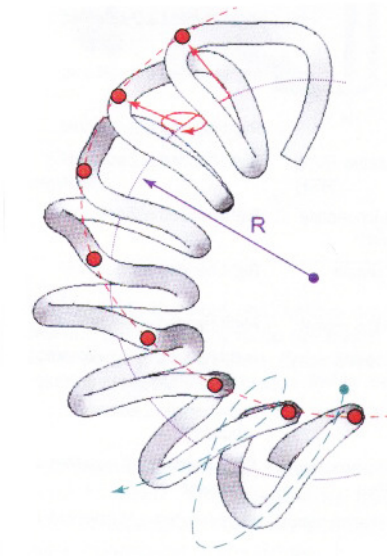


Figura 15: una schematica rappresentazione di una proteina solenoide: sono evidenziate la curvatura (magenta) l'orientazione (verde) e la torsione (rosso)

Lo stato disordinato di una proteina può essere causato, ad esempio, da un eccesso di residui di prolina (> 15-20%). Possiamo generalmente dire che le proteine con un contenuto maggiore del 20% di prolina hanno una elevata probabilità di essere disordinate. La prolina ha infatti la capacità di inibire la libertà conformazionale che portano alla formazione di alfa-eliche e beta-strands. Nelle proteine disordinate vi è inoltre una bassa percentuale di residui non polari, che non possono

probabilmente generare un core idrofobico in grado di stabilizzare la struttura. È possibile porre una soglia del 30% per quanto riguarda i residui non polari: al di sotto di questa soglia, le proteine non si strutturano stabilmente. Una eccezione alle proteine con presenza di prolina, sono quelle con ripetizioni di collagene. Ancora, un'altra eccezione a queste regole è la seguente: se le ripetizioni sono localizzate in posizioni con residui polari (Ser, Thr, Asn, Asp, His e Cys), le catene laterali inducono legami ad idrogeno e covalenti che consentono una struttura stabile.

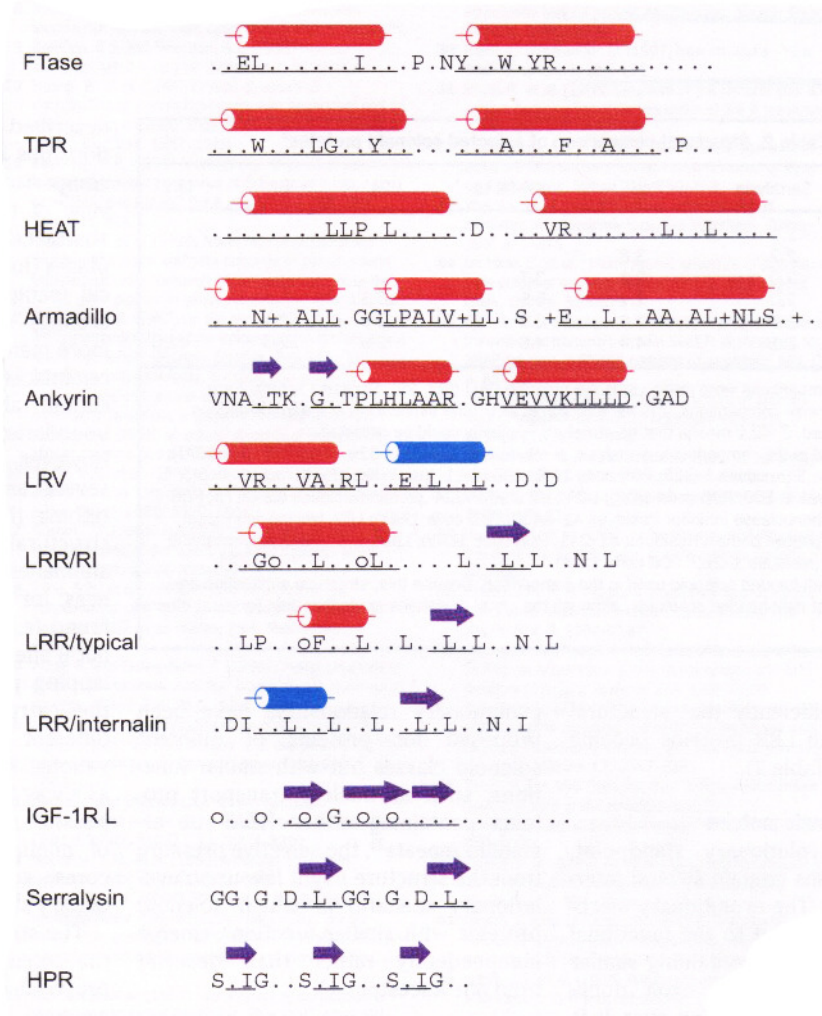


Figura 16: sequenze consensus di repeats presenti nelle proteine solenoidi. Sono evidenziati gli amminoacidi e i pattern più frequenti nelle proetine ripetute.

2.2 Lunghezze e strutture

La lunghezza delle ripetizioni, se conosciuta, può aiutarci a identificare la possibile struttura 3D delle proteine ripetute, nonché il loro stato di oligomerizzazione. Con una accurata analisi, le proteine solenoidi sono classificate in quattro classi principali:

- Classe I : lunghezza di 1-2 residui (peptidi)
- Classe II: lunghezza di 3-4 residui (proteine fibrose)
- Classe III: lunghezza di 5-40 residui (proteine solenoidi)
- Classe IV: lunghezza >30 residui, fibre "*beads-on-a-string*" , non utili per fornire informazioni atte a semplificare la predizione del ripiegamento.

Si presume che, con l'aumento della lunghezza delle ripetizioni, si formino nuove classi con struttura regolare. Attualmente, sono conosciute poche strutture di proteine solenoidi, sempre composte da un numero di residui ripetuti pari a 5-40.

Queste sono alcune proprietà delle proteine solenoidi conosciute:

- lunghezza del repeat: 16 – 40 amminoacidi
- strutture di base: formate da 1 a 4 strutture secondarie (alfa-eliche, beta strands)

2.3 Classificazione legate ai repeats del DNA

Altre proprietà possono emergere quando la classificazione si basa sulle ripetizioni del DNA:

- microsatelliti, con lunghezza < 6 nucleotidi (classe I)
- minisatelliti, 10-100 nucleotidi: classe II
- satelliti, lunghezza >100 nucleotidi, classe IV

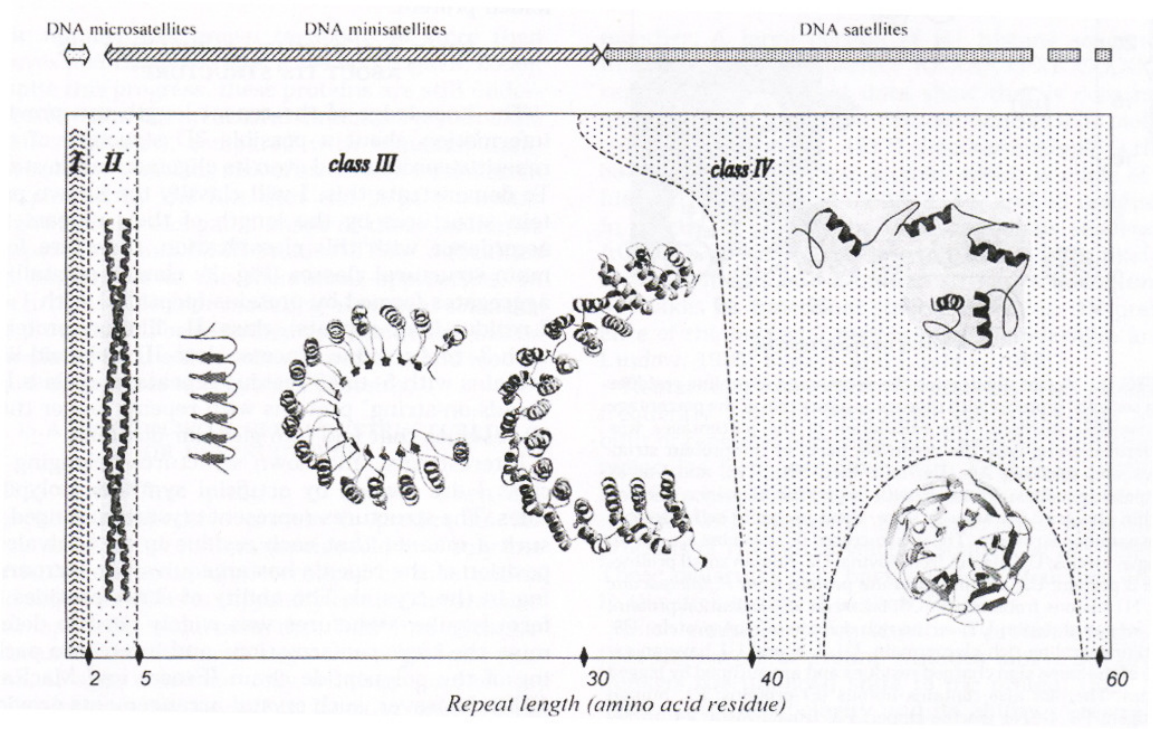


Figura 17: Classificazione strutturale delleproteine ripetute in base alla lunghezza dei repeats

La coincidenza dovuta alle lunghezze del repeat nel DNA non è ancora stata spiegata.

Mediante l'utilizzo di dati sperimentali, stato di oligomerizzazione e orientamento delle alfa-eliche, è possibile generare una predizione per le alfa eliche strutturate a spirale. Ulteriori affinamenti possono essere introdotti con tecniche di dinamica molecolare e minimizzazione dell'energia.

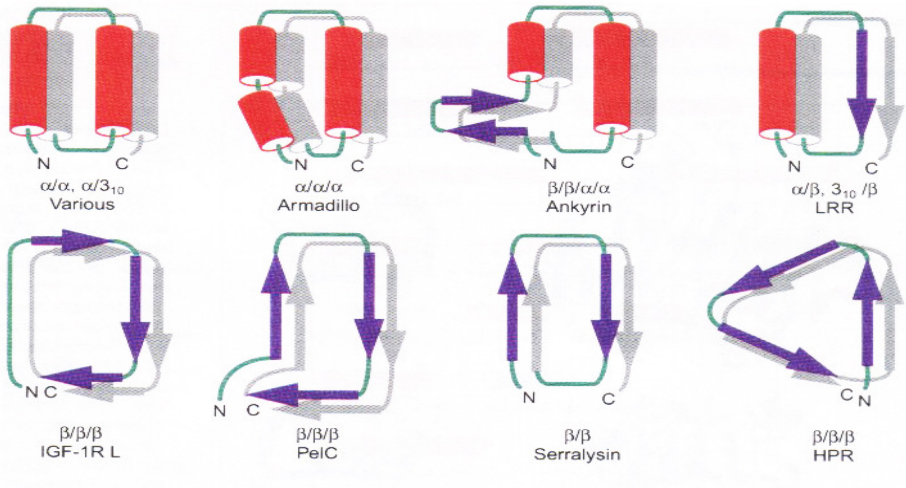


Figura 18: rappresentazione schematica delle principali unità di ripetizione nelle proteine solenoidi

2.4 Predizione e modeling di proteine solenoidi

Alcuni tentativi di predizione per le proteine ripetute hanno fatto largo uso di informazioni sulla presenza di particolari amminoacidi (per esempio di leucina) o di pentapeptidi, all'interno della sequenza primaria. La conformazione più comune per le proteine ripetute (Classe III) è una superelica composta da uno o più elementi di struttura secondaria (alfa eliche e foglietti beta).

Complessivamente, l'importanza della determinazione dei repeat di un solenoide è riassumibile in tre principali motivazioni:

1. può essere utile alla ricostruzione cristallografica a raggi X
2. può aiutare a determinare i confini tra due domini di una proteina
3. può essere determinante alla costruzione di nuovi ripiegamenti che sfruttano la duplicazione

Esistono tre classi di metodologie per determinare le ripetizioni in una sequenza. La prima è deputata a rilevare proteine fibrose, ovvero che non permettono le inserzioni nelle unità ripetute (Coward and Drablos, 1998) o tra le unità ripetute (Gruber et al., 2005; Lupas et al., 1991; McLachlan and Stewart, 1976; Newman and Cooper, 2007). La seconda metodologia utilizza informazioni a priori, ovvero sfruttano dati riguardanti unità ripetute in forma di profilo di molte strutture note. I profili vengono comparati con la sequenza di ingresso e, in caso di riscontro multiplo, è necessario conoscere la famiglia di origine della sequenza. Tra questi metodi vi sono HMMER/Pfam (Eddy, 1998; Sonnhammer et al., 1998), REP (Andrade et al., 2000) e Mocca (Notredame, 2001). La terza metodologia si basa sul concetto di identificazione in base alla scoperta di simmetrie interne. Metodologie di questo tipo sono ad esempio: internal repeat finder (Sonnhammer et al., 1998), PROSPERO (Mott, 2000), REPRO (Heringa e Argos, 1993), RADAR (Heger e Holm, 2000), TRUST [successore di REPRO, Szklarczyk and Heringa (2004)], e il server HHrep (Söding et al., 2006).

Una considerazione a parte sulle metodologie adottate va dedicata a REPETITA, un software sviluppato da BioComputing UP, in grado di ottenere prestazioni elevate utilizzando la trasformata di Fourier, o meglio, la Discrete Fourier Transform. REPETITA è stato progettato partendo dalle medesime assunzioni impiegate per Ouroboros, e scopo di questa tesi è capire se il criterio usato per REPETITA sia migliorabile oppure non abbia margini di miglioramento.

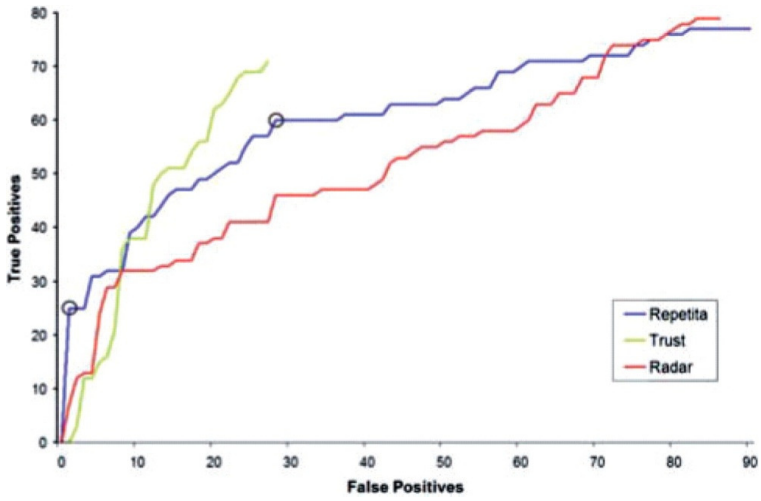


Figura 19: Le prestazioni di REPETITA rispetto ad altri software. Il numero di falsi positivi (asse x) è rappresentato in funzione del numero dei veri positivi (asse y). Le prime 25 predizioni di REPETITA sono veri positivi.

CAPITOLO III

Materiali e metodi

In questo capitolo si analizzano le metodologie adottate per estrarre i segnali utili all'analisi di Ouroboros, nonché vengono presentati i software che hanno permesso di ricavare queste informazioni. Si tratta per lo più di software molto noti (Psi-Blast su tutti) e di metodologie per l'analisi delle proteine di nuova generazione, mutuate da alcune pubblicazioni all'avanguardia del gruppo BioComputing. Viene descritta brevemente, inoltre, la libreria VICTOR, una libreria molto vasta con classi essenziali per l'analisi bioinformatica, e che ha permesso una facile intereazione di Ouroboros con i metodi appena introdotti.

3.1 Allineamento di sequenze

Acidi nucleici e proteine sono costituiti da catene di quattro possibili residui nucleotidici e di venti possibili residui amminoacidici, rispettivamente. Essendo queste catene delle stringhe di caratteri, possono essere analizzate con metodi informatici in modo da identificare pattern particolari o effettuare allineamenti tra sequenze. L'allineamento di due sequenze, siano esse acidi nucleici o proteine, è importante per ottenere un confronto diretto tra le sequenze in esame. La procedura di

allineamento, inoltre, è il presupposto per analisi più complesse come la ricerca in banca dati per similarità o la costruzione di alberi filogenetici o l'identificazione di motivi funzionali.

Spesso i termini similarità ed omologia vengono utilizzati come sinonimi anche se non è così. Il termine omologia indica che due entità condividono una stessa origine filogenetica, da cui si sono poi evolute differenziandosi l'una dall'altra. Il termine similarità ha un significato più generale che indica somiglianza prescindendo dalle ragioni che l'hanno determinata. La similarità biologica può essere spesso dovuta a omologia ma può essere anche generata dal caso o da fenomeni di convergenza adattativa sia a livello morfologico (analogia: ala di uccello e pipistrello) che a livello molecolare. L'omologia è una caratteristica qualitativa, la similarità è una caratteristica quantitativa (dipende dal criterio scelto per valutare la somiglianza).

Due sequenze omologhe, che condividono la stessa origine evolutiva, possono differenziarsi per le modalità con cui si sono evolute rispetto al comune ancestore. Per tale motivo distinguiamo i seguenti casi di sequenze omologhe:

- sequenze ortologhe: che derivano da un processo di speciazione

- sequenze paraloghe: che derivano da duplicazione genica

- sequenze xenologhe: che derivano per trasferimento orizzontale

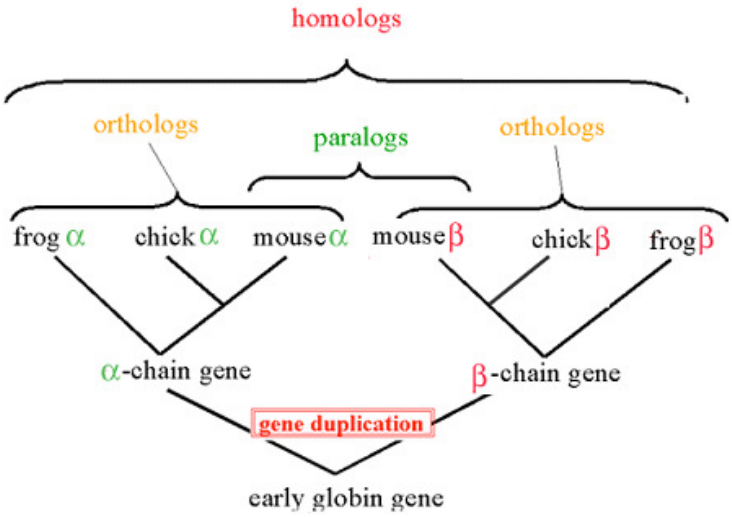


Figura 20: Albero di classificazione delle sequenze globiniche

Mediante l'allineamento dovremmo essere in grado di determinare le regioni comuni tra due sequenze. Il problema non è però così semplice in quanto la somiglianza tra due sequenze è basata sul concetto di similarità che è abbastanza generico. Ne consegue che non possiamo allineare due sequenze se non definiamo un criterio per valutare la similarità. D'altra parte se si vuole valutare la similarità tra due sequenze bisogna allinearle. Similarità ed allineamento sono, quindi, intimamente associate tra loro.

Per allineare due sequenze, una volta definito un criterio di similarità, bisogna disporre di un metodo o algoritmo. Nel caso più semplice supponiamo che il criterio di similarità scelto sia quello di valutare il numero di lettere che si appaiano esattamente tra due sequenze. In questo caso l'algoritmo più semplice è quello di considerare lo scorrimento di una sequenza sull'altra fin quando la sovrapposizione tra le sequenze è massima.

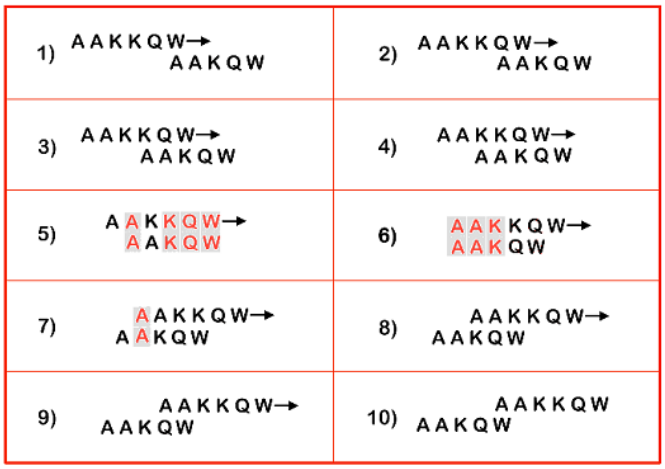


Figura 21: Il principio base del raffronto tra sequenze per ottenere un allineamento

Normalmente, le sequenze biologiche che si confrontano non hanno la stessa lunghezza e bisogna anche pensare al fatto che durante l'evoluzione le sequenze non hanno solo subito sostituzioni nucleotidiche ma anche inserzioni e/o delezioni (Indels). Tali indels devono essere inseriti nell'allineamento come gap.

L'inserimento dei gap permette di migliorare l'allineamento come nell'esempio sottostante. Nel primo caso abbiamo 10 appaiamenti esatti, l'inserimento di due gap nel secondo caso, invece, incrementa il numero di appaiamenti esatti a 25. I gap, tuttavia, complicano il problema dell'allineamento sia perché è necessario introdurre un altro criterio per valutare la similarità e sia

perché l'algoritmo di scorrimento dovrebbe effettuare un numero di confronti molto maggiore per ogni gap aggiunto.

```
   IPLMTRWDQEQESDFGHKLPIYTREWCTRG
             ||||||||||
CHKIPLMTRWDQQESDFGHKLPVIYTREW

   IPLMTRWDQEQESDFGHKLP-IYTREWCTRG
   ||||||||| |||||||||| ||||||
CHKIPLMTRWDQ-QESDFGHKLPVIYTREW
```

Figura 22: Un esempio di allineamento e allineamento che tiene conto di gaps

Per tener conto dei gap, uno dei modi è quello di assegnare una penalità ogni volta che un gap è inserito. E' possibile inserire una penalità anche ogni qual volta un gap viene esteso (gap opening - gap extension).

$$Score = \sum_{i=1}^{L} s(a_i, b_i) - \sum_{j=1}^{G} \left(\gamma + \delta\left(len(j) - 1\right)\right)$$

Dove L= lunghezza dell'allineamento, S(ai,bi) = score per l'appaiamento, G = numero di gap, γ = penalità per l'apertura del gap, δ = penalità per l'estensione del gap.

3.2 BLAST

BLAST (Basic Local Alignment Search Tool) è un software per la generazione di allineamenti di sequenze sviluppato nel 1990 dall'NCBI (National Center for Biotechnology Information) con sede a Bethesda, negli Stati Uniti.

L'algoritmo che produce come risultato l'allineamento è suddiviso in tre parti:

- nella prima fase viene generata una lista di parole leggendo una ad una tutte le parole di lunghezza W (word) di una sequenza di ingresso (query). Per ciascuna di queste parole della *query*, viene compilata una lista di parole affini (chiamate W-mers) che danno uno score maggiore di una soglia (T) se allineate con la parola della sequenza *query*.

- nella seconda fase si analizzano tutte le sequenze della banca dati ricercando la presenza dei W-mers corrispondenti alle parole della lista prodotta dall'indicizzazione della sequenza *query*. Ogni corrispondenza trovata (*hit*) potrebbe rappresentare una porzione di un allineamento più ampio e dovrà pertanto essere approfondita in seguito.
- nella terza fase si verifica se e quanto sia possibile aumentare ogni hit. Questo processo avviene tentando di allungare l'allineamento in entrambe le direzioni senza considerare l'inserimento di gap. Lavorando in questo modo, si ottiene un segmento di allineamento locale non ulteriormente estendibile (*High-scoring Segment Pair, HSP*). Il parametro S definisce una soglia di score sopra la quale un HSP è ritenuto degno di attenzione. Il parametro X (misurato in termini di perdita di score) stabilisce quanto il programma debba insistere nel cercare di incrementare gli HSP in presenza di score negativi.

I parametri utilizzati da BLAST sono quattro: W, T, S e X. I primi due sono particolarmente importanti perché determinano l'ampiezza della lista di parole affini: per ogni valore di W, più il valore di T è basso, più esteso sarà l'elenco di W-mers (con un conseguente incremento del tempo di esecuzione del programma). Viceversa, alti valori di T aumentano il rischio di non identificare alcuni HSP. Anche il parametro X influenza le prestazioni del metodo. La massimizzazione del valore di X accresce il tempo di esecuzione perché l'intorno di ogni hit viene esplorato a maggiore profondità. Nel settaggio dei parametri, bisogna dunque trovare un giusto compromesso fra velocità di esecuzione e sensibilità della ricerca.

Le prime versioni di BLAST impiegavano mediamente più del 90% del tempo di esecuzione nel tentativo di allungare gli hits; per ovviare a questo inconveniente, le versioni più recenti hanno introdotto una variazione all'algoritmo originale denominata two-hits method. L'idea è di considerare solo i casi in cui almeno due hits indipendenti si verificano in vicinanza fra loro, a una distanza non superiore ad A, sulla stessa diagonale (cioè allineabili senza gap sulle due sequenze). Impostando opportunamente i parametri, il nuovo algoritmo raggiunge sensibilità maggiori con tempi di esecuzione minori rispetto alle versioni precedenti. Un aspetto fondamentale di BLAST è la sua base statistica estremamente solida che consente di produrre una stima accurata del significato di ogni allineamento. Il parametro T è generalmente impostato automaticamente dal programma. Ad esempio, per allineamenti di proteine con W=3 e con la matrice BLOSUM62, l'algoritmo originale adotta T=13 (la variante two-hits, T=11 e A=40).

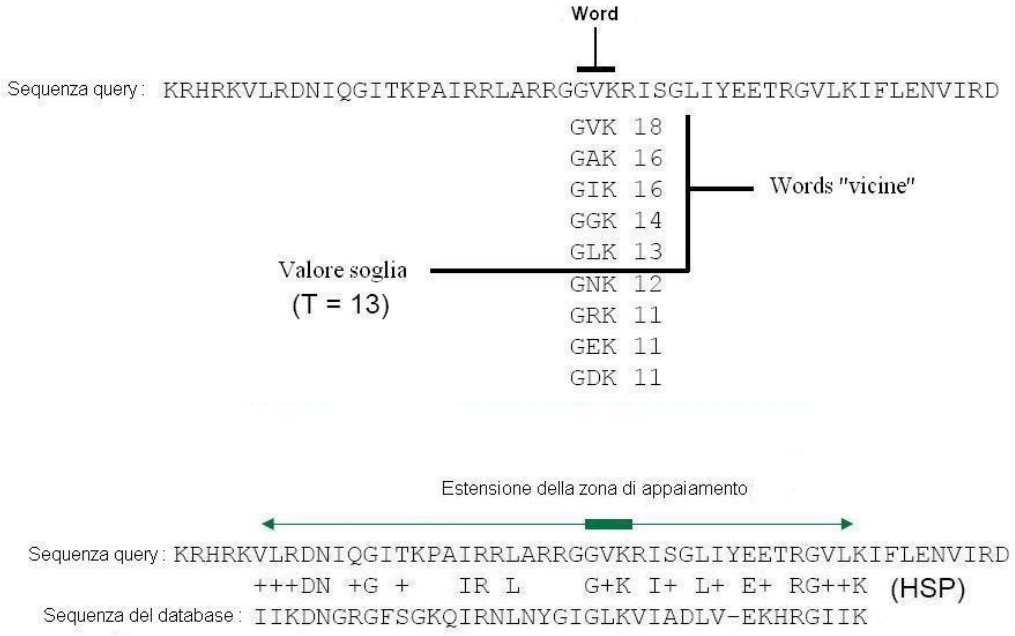


Figura 23: schema del funzionamento di BLAST

Con questi parametri, la possibilità di un HSP con score alto ma senza hits si riduce a livelli minimi. La statistica di BLAST permette inoltre di rapportare il valore di S (score) con il numero atteso (E o Expected) di HSP che raggiungono tale soglia in una banca dati di sequenze casuali della stessa grandezza di quella considerata. Le variabili S ed E sono legate da una relazione di proporzionalità inversa secondo una complessa legge statistica. Dato un valore di E, quindi, il valore dello score S è ricavato sfruttando questa relazione di proporzionalità. Se il valore di E non viene definito, BLAST lo imposta automaticamente a 10. Se, per esempio, E viene impostato al valore 0.1, lo score S sarà maggiore, e di conseguenza HSP con score bassi (al d sotto della soglia) non saranno considerati.

Sono state sviluppate diverse versioni del programma BLAST che consentono di ricercare sequenze a livello di acidi nucleici o di proteine. La lista seguente riassume le varianti esistenti e le loro funzioni specifiche:

- *blastp* ricerca similarità in una banca di sequenze amminoacidiche a partire da una sequenza query di amminoacidi.
- *blastn* ricerca similarità in una banca di sequenze nucleotidiche a partire da una sequenza query di nucleotidi.

- *blastx* ricerca similarità in una banca di sequenze amminoacidiche a partire da una sequenza query di nucleotidi, dopo aver tradotto automaticamente la query in amminoacidi usando tutte le possibili fasi di lettura.
- *tblastn* ricerca similarità in una banca di sequenze nucleotidiche a partire da una sequenza query di amminoacidi, traducendo automaticamente ogni sequenza della banca dati in tutte le possibili fasi di lettura.
- tblastx cerca similarità in una banca di sequenze nucleotidiche a partire da una sequenza query di nucleotidi, traducendo automaticamente sia la sequenza query sia le sequenze subject in amminoacidi in tutte le possibili fasi di lettura.

Negli ultimi anni, per rispondere a particolari esigenze bioinformatiche, sono stati implementati numerosi programmi basati su BLAST:

- PSI-BLAST (Position Specific Iterated BLAST) sfrutta una ricerca iterativa in cui le sequenze trovate ad ogni ciclo sono usate per costruire un modello di punteggio per il ciclo successivo. In questo modo, è possibile determinare i profili che caratterizzano le sequenze conservate nell'ambito di un particolare dominio funzionale.
- PHI-BLAST (Pattern Hit Initiated BLAST) combina PSI-BLAST con la capacità di identificare pattern regolari. Con questo programma, si possono cercare sequenze simili che, allo stesso tempo, contengano uno specifico pattern in vicinanza della regione di similarità.
- BL2SEQ (BLAST two SEQuences) è una versione di BLAST dedicata all'allineamento pairwise (due sequenze) con le stesse opzioni elencate sopra.
- MegaBLAST concatena molte queries fra loro per ottenere tempi di esecuzione più veloci.

MegaBLAST è ottimizzato per allineare sequenze molto simili fra loro e ha tempi di esecuzione fino a dieci volte inferiori rispetto alla versione normale di BLAST, ma necessita di maggiori risorse di memoria. Questa applicazione è particolarmente adatta per il confronto incrociato di due banche dati di sequenze. BLAST è disponibile gratuitamente per l'installazione su piattaforme locali di ogni tipo o per l'esecuzione remota sul potente web server dell'NCBI. Presso il sito, è inoltre consultabile un tutorial che introduce all'uso del programma, al funzionamento dell'algoritmo e all'analisi statistica ad esso associata

3.3 Il problema metrico delle proteine

Un notevole ostacolo a una rigorosa analisi statistica dei dati di sequenza biologico è il cosiddetto problema della metrica delle sequenze cioè l'utilizzo delle sequenze di simboli alfabetici per caratterizzare le sequenze. Codici Lettera mancanza di una metrica naturale sottostante per il confronto. Ad esempio, la leucina (L) è più simile nelle sue proprietà fisico-chimiche alla valina (V) di quanto sia rispetto all'alanina (A). Tuttavia, l'uso di queste lettere dell'alfabeto non è utile a definire queste particolari proprietà. Singole lettere usate come variabili nelle analisi di sequenze generano una significativa perdita di risoluzione e di informazioni sulle proprietà fisico-chimiche degli aminoacidi.

Il problema da risolvere è dunque quello di definire una o più proprietà (siano esse strutturali o chimiche o fisiche) a partire dalla semplice struttura primaria. Attraverso l'analisi statistica multivariata di un gran numero di aminoacidi, William Atchley e lo staff del Bioinformatics Research Center della North Carolina, si è tentato di dare una risposta a questo problema metrico per le sequenze. L'analisi fattoriale è usata per ricavare un piccolo insieme di valori numerici che riassumono componenti di grandi dimensioni ed interpretabili di variazione di aminoacidi. L'approccio ha molti aspetti positivi e facilita notevolmente l'analisi statistica dei dati delle sequenze, in quanto gli score numerici prodotti in questo modo sono di utilità generale e possono essere impiegati in molti tipi di analisi. Gli obbiettivi dell'individuazione di questo insieme di valori numerici è quello di rivelare, ad esempio, la struttura latente di dati multidimensionali dai singoli amminoacidi e dai loro attributi, descrivere i pattern principali di co-varianze tra questi attributi, di esplorare quali siano le cause di variazione degli attributi. L'approccio facilita la (i) comprensione dell'entità dimensionale delle sequenze multivariate, (ii) la rilevazione di pattern multidimensionali negli attributi degli amminoacidi, (iii) la comprensione tra le interrelazioni variazione di sequenza, strutturali e funzionali, (iv) la standardizzazione dei dati di ingresso per molti diversi tipi di analisi, (v) in decomposizione variazione di sequenza nelle sue componenti di base evolutive, strutturali e funzionali, e (vi) la modellazione dinamica della variabilità delle proteine.

Un indice di aminoacidi è una serie di 20 valori numerici che rappresentano caratteristiche fisico-chimiche e biologiche diverse. I dati analizzati da Atchley sono stati ottenuti da un database on-line contenente 494 valori per gli amminoacidi (www.genome.ad.jp/dbget/aaindex.html). Questi valori sono descrivono attributi generali, quali il volume o le dimensioni molecolari, l'idrofobicità e la carica

elettrostatica, oltre a misure più specifiche, come la quantità di energia libera per atomi o per angolo di orientamento di catene laterali.

Amino acid attribute	F I	F II	F III	F IV	F V	Com.
Average nonbonded energy per atom	1.028	0.074	0.152	0.047	−0.079	0.982
Percentage of exposed residues	1.024	0.016	0.194	0.095	0.025	0.965
Average accessible surface area	1.005	−0.034	0.159	0.059	0.153	0.994
Residue accessible surface area in folded protein	0.950	0.098	0.178	0.039	0.237	0.961
No. of hydrogen bond donors	0.809	0.021	0.122	0.021	0.357	0.808
Polarity	0.790	−0.044	−0.388	0.027	−0.092	0.956
Hydrophilicity value	0.779	−0.153	−0.333	0.213	0.023	0.862
Polar requirement	0.775	−0.128	−0.335	−0.020	−0.245	0.939
Long range nonbonded energy per atom	0.725	−0.024	−0.394	0.189	−0.104	0.905
Negative charge	0.451	−0.218	−0.024	−0.052	−0.714	**0.737**
Positive charge	0.442	−0.246	−0.225	−0.085	0.708	**0.730**
Size	0.440	−0.112	0.811	−0.144	0.108	0.915
Normalized relative frequency of bend	0.435	0.674	−0.225	0.082	−0.118	0.912
Normalized frequency of β-turn	0.416	0.648	−0.346	−0.019	−0.079	0.969
Molecular weight	0.363	−0.091	0.657	−0.504	−0.047	0.923
Relative mutability	0.337	−0.172	−0.183	0.297	−0.296	**0.416**
Normalized frequency of coil	0.271	0.863	0.028	0.123	0.073	0.860
Average volume of buried residue	0.269	−0.153	0.766	−0.340	0.016	0.928
Conformational parameter of β-turn	0.243	0.693	−0.185	−0.439	0.078	0.837
Residue volume	0.225	−0.172	0.794	−0.292	0.036	0.946
Isoelectric point	0.224	−0.060	−0.049	0.163	0.967	0.955
Optimized propensity to form reverse turn	0.224	−0.005	−0.433	0.319	−0.194	**0.563**
Chou–Fasman parameter of coil conformation	0.201	0.780	−0.338	−0.052	0.048	0.948
Information measure for loop	0.196	0.786	−0.193	−0.335	0.181	0.908
Free energy in β-strand region	0.189	0.447	−0.125	0.127	−0.150	**0.369**
Side chain volume	0.181	−0.201	0.754	−0.299	0.088	0.948
Amino acid composition of total proteins	0.155	−0.163	−0.042	0.963	0.040	0.931
Average relative probability of helix	0.150	−1.004	−0.163	−0.068	−0.040	0.977
α-Helix indices	0.136	−0.939	−0.183	−0.219	0.014	0.893
Relative frequency of occurrence	0.111	−0.122	−0.079	0.931	−0.005	0.897
Helix–coil equilibrium constant	0.106	−0.724	0.368	−0.112	0.053	0.854
Amino acid composition	0.101	−0.024	−0.245	0.852	0.048	0.873
No. of codon(s)	0.079	0.133	0.087	0.867	0.294	**0.778**
Net charge	0.078	0.041	−0.004	0.147	0.967	0.932
Normalized frequency of turn	0.075	0.831	−0.088	−0.393	−0.051	0.859
Relative frequency in α-helix	0.061	−0.987	−0.270	−0.215	0.024	0.945
Average nonbonded energy per residue	0.042	0.376	0.001	−0.507	−0.295	**0.428**
Bulkiness	−0.036	−0.105	0.988	0.059	−0.244	0.897
Normalized relative frequency of coil	−0.047	0.353	−0.582	−0.082	0.135	**0.494**
Refractivity	−0.049	−0.061	0.471	−0.621	0.095	0.854
Normalized frequency of left-handed α-helix	−0.079	0.366	−0.641	−0.075	0.273	**0.558**
Heat capacity	−0.163	−0.366	0.152	−0.656	0.006	**0.721**
Free energy in α-helical region	−0.178	0.858	−0.002	−0.096	−0.101	**0.750**
Hydrophobicity factor	−0.224	0.200	0.833	−0.008	−0.098	**0.728**
Normalized frequency of extended structure	−0.390	0.335	0.706	0.152	0.054	**0.779**
Normalized frequency of β-sheet, unweighted	−0.460	0.108	0.611	0.121	0.040	**0.711**
Normalized frequency of β-sheet	−0.506	0.021	0.580	0.021	0.110	**0.795**
Information measure for pleated-sheet	−0.522	−0.132	0.438	0.069	0.179	**0.724**
Hydropathy index	−0.856	−0.171	0.131	0.221	−0.028	0.950
Eisenberg hydrophobic index	−0.864	0.008	0.175	0.004	−0.268	0.911
Average side chain orientation angle	−0.896	−0.160	0.000	−0.113	0.187	0.858
Average interactions per side chain atom	−0.928	−0.127	−0.141	0.062	0.135	0.842
Transfer free energy	−1.003	−0.027	−0.116	−0.114	−0.137	0.982
Percentage of buried residues	−1.017	−0.125	−0.169	−0.074	0.044	0.967

Figura 24: I 54 attributi da cui Atchley et Al. hanno estratto la matrice di sostituzione di Atchley. Come si può evincere soffermandoci sugli attributi, la ridondanza di alcune informazioni hanno reso necessaria una semplificazione, di cui la matrice di Atchley è il risultato.

L'analisi fattoriale è stata quindi utilizzata per creare un sottoinsieme di valori numerici, che possano riassumere con pochi valori le proprietà fisico-chimiche degli amminoacidi. Le 494 proprietà, a seguito dell'analisi fattoriale, sono risultati altamente ridondanti. Un sottogruppo di 54 caratteristiche è stato prima selezionato in relazione all'ampiezza relativa dei coefficienti fattoriali, alle proprietà statistiche di distribuzione, alla relativa facilità di interpretazione, e alla riconosciuta importanza strutturale. L'analisi ulteriore di queste 54 parole di aminoacidi ha generato cinque "cluster", o modelli di variabili fisico-chimiche, altamente intercorrelati: una riduzione della dimensioni di due ordini di grandezza rispetto agli originali 494 attributi.

AA	F1	F2	F3	F4	F5
A	-0.591	-1.302	-0.733	1.570	-0.146
C	-1.343	0.465	-0.862	-1.020	-0.255
D	1.050	0.302	-3.656	-0.259	-3.242
E	1.357	-1.453	1.477	0.113	-0.837
F	-1.006	-0.590	1.891	-0.397	0.412
G	-0.384	1.652	1.330	1.045	2.064
H	0.336	-0.417	-1.673	-1.474	-0.078
I	-1.239	-0.547	2.131	0.393	0.816
K	1.831	-0.561	0.533	-0.277	1.648
L	-1.019	-0.987	-1.505	1.266	-0.912
M	-0.663	-1.524	2.219	-1.005	1.212
N	0.945	0.828	1.299	-0.169	0.933
P	0.189	2.081	-1.628	0.421	-1.392
Q	0.931	-0.179	-3.005	-0.503	-1.853
R	1.538	-0.055	1.502	0.440	2.897
S	-0.228	1.399	-4.760	0.670	-2.647
T	-0.032	0.326	2.213	0.908	1.313
V	-1.337	-0.279	-0.544	1.242	-1.262
W	-0.595	0.009	0.672	-2.128	-0.184
Y	0.260	0.830	3.097	-0.838	1.512

Tabella 1: Tabella con i valori per gli attributi di Atchley. Sono 5 valori per ogni amminoacido, che descrivono rispettivamente: Polarità, struttura secondaria, dimensioni molecolari, differenze tra codoni ed elettrostaticità.

Di seguito, una descrizione di questi 5 "cluster", in base alle loro caratteristiche principali:

- Fattore I: è una variabile bipolare (grandi coefficienti positivi e negativi) che riflette covarianza simultanea nelle porzioni dei residui esposti rispetto ai residui sepolti, energia di non legame rispetto all'energia libera, il numero dei donatori di legame a idrogeno, la polarità in confronto alla apolarità, e l'idrofobia paragonata all'idrofilia. Per semplicità, indicheremo il Fattore I come un indice di polarità
- Fattore II: è un fattore che riflette proprietà riguardanti la struttura secondaria. C'è una relazione inversa di propensione relativa per vari aminoacidi in diverse configurazioni strutturali secondarie, come ad esempio un avvolgimento, una girata, o un ripiegamento rispetto alla frequenza di un α-elica.
- Fattore III : riguarda le dimensioni molecolari o il volume molecolare, con elevati coefficienti per la voluminosità, volume di un residuo, volume medio di un residuo nascosto, volume delle catene laterali, e per il peso molecolare. Un fattore con grande peso negativo si ha per α-eliche sinistrorse.
- Fattore IV: riflette la relativa composizione degli amminoacidi in varie proteine, il numero di codoni che codificano per un aminoacido, e la composizione degli aminoacidi. Questi attributi variano inversamente con rifrangenza e la capacità termica.
- Fattore V: è un coefficiente che descrive la carica elettrostatica, con alti valori per il punto isoelettrico e per la carica netta. Come si può supporre, vi è una relazione inversa tra carica positiva e negativa.

Il valore previsto per ogni amminoacido di ciascuno dei fattori può essere tradotto dalla sequenza al valore di attributo: ogni amminoacido può essere rappresentato in cinque variabili numeriche, altamente versatili e interpretabili secondo le esigenze. Banche dati di sequenze allineate possono essere convertite in cinque distinte matrici di valori numerici per facilitare le analisi statistiche di omologhi. Anche le tradizionali analisi statistiche possono essere effettuate con questa nuova matrice per la trasformazione delle sequenze amminoacidiche.

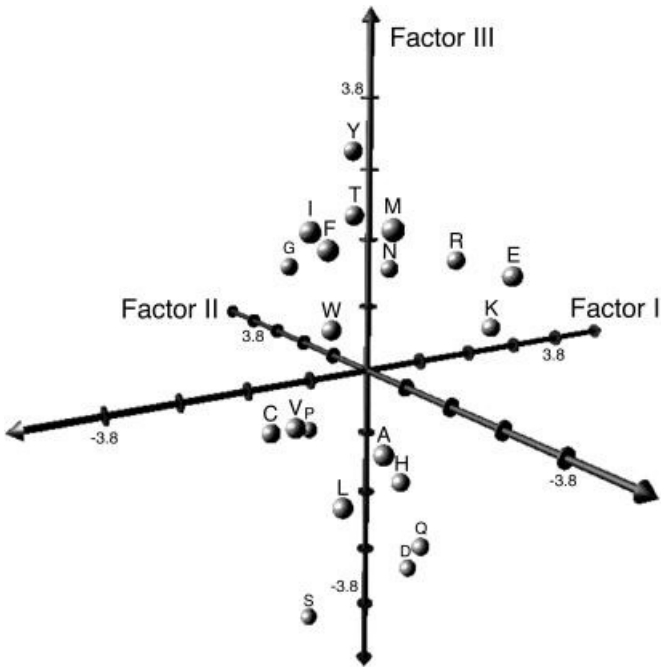


Figura 25: I fattori o attributi di Atchley visualizzati nello spazio 3D

3.4 Dataset

Il server TESE (Sirocco & Tosatto, 2008) è stato utilizzato per ricercare le proteine solenoidi, partendo da un iniziale set di 32 proteine. TESE genera una set di proteine non ridondante, con struttura nota, rifacendosi alla classificazione CATH. Il numero totale di proteni solenoidi ricavate è di 105 elementi. Il set delle proteine non solenoidi è invece stato ricavato dal server scegliendo in maniera random, scegliendo strutture con differenti topologie e senza una precisa similarità di sequenza. In totale, il dataset globulare contiene 247 proteine. Il dataset, a sua volta, è stato sudiviso in due set, uno di 50 solenoidi e 119 non-solenoidi, l'altro di 55 solenoidi e 128 non solenoidi.

3.5 La libreria VICTOR

La libreria VICTOR (VIrtual Construction TOol for pRotein design), sviluppata dal Prof. Silvio C.E. Tosatto come progetto di dottorato, è un pacchetto di tools sviluppato in linguaggio C++, il cui scopo spazia dalla predizione, all'analisi e alla modellazione delle strutture proteiche.

In questo progetto, è stata particolarmente prezioso l'ausilio della sottolibreria ALIGN, in cui sono implementate tecniche e metodologie di allineamento di sequenze proteiche. L'integrazione del software Ouroboros all'interno della libreria VICTOR ha permesso l'utilizzo di classi in grado di leggere e interpretare le informazioni provenienti da allineamenti per utilizzarle allo scopo di aumentare le nostre conoscenze sul dato iniziale, attraverso un preventivo filtraggio e una ottimizzazione rispetto a quantità "spurie" che avrebbero altrimenti compromesso notevolmente l'analisi successiva.

La libreria ALIGN, mediante le sue funzioni, consente in particolare:

- la produzione di allineamenti sequenza contro sequenza, profilo contro sequenza e profilo contro profilo.
- la produzione di allineamenti globali, locali e semi-globali.
- l'utilizzo di funzioni per la penalizzazione dei gap lineari, affini e variabili.
- l'utilizzo di *weighting schemes* nella costruzione dei profili.
- l'utilizzo di *scoring functions* negli allineamenti profilo contro profilo.
- l'impiego di informazioni strutturali di varia natura.

Anche se i campi di impiego di ALIGN, come si è visto, sono molto eterogenei, per Ouroboros l'utilizzo della libreria è limitato alle funzioni di *parsing* (analisi di uno *stream* continuo di input) per dati provenienti da allineamenti ottenuti con BLAST, e alla estrazione di informazioni utili alla creazione di un segnale di ingresso pulito.

CAPITOLO IV

Struttura del programma Ouroboros

Il software Ouroboros, sviluppato come progetto di tesi sperimentale presso il laboratorio BioComputing UP, rappresenta un tentativo di indagine nell'ambito delle proteine solenoidi, ambito di recente sviluppo e di continua crescita. Le problematiche più consistenti, quando si considerano le proteine solenoid, riguardano principalmente l'inadeguatezza dei metodi classici di indagine nel fornire una descrizione completa per queste strutture proteiche. Il compito dei software di predizione, di cui Ouroboros fa parte, è quello di sopperire a queste mancanze, introducendo per l'appunto l'analisi predittiva e cercando di ottenere le informazioni necessarie a completare il quadro rappresentativo per questa famiglia di proteine. Le due investigazioni principali per le strutture solenoidi concernono l'individuazione e il distinguo rispetto alle proteine globulari e la determinazione dei periodi di ripetizione. Il software che viene qui descritto tenta un nuovo approccio per tentare di dare una risposta a questi interrogativi.

Ouroboros è un programma scritto in linguaggio C++, ed integrato con la libreria VICTOR, grazie alle cui classi preesistenti è in grado di produrre un allineamento di sequenze e generare quindi un profilo. Le successive strutture sono state aggiunte per modificare ed elaborare i dati riguardanti le

sequenze proteiche. In questo capitolo, si descrive la struttura di Ouroboros nelle sue parti principali, facendo uso, ove possibile, anche di illustrazioni didascaliche e frammenti di pseudo-codice.

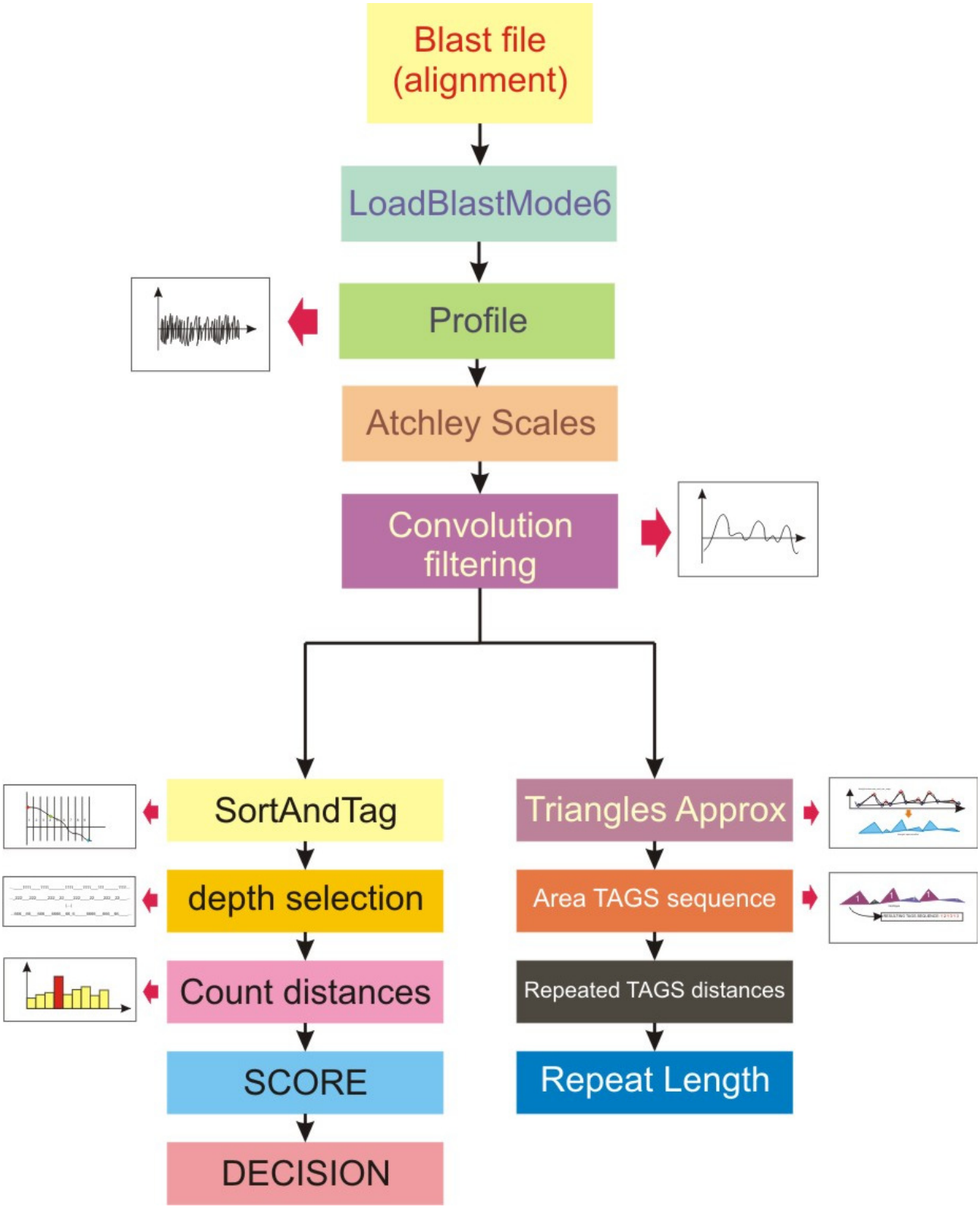


Figura 26: schema della struttura di Ouroboros

Lo schema generale per la struttura di Ouroboros è visualizzato in figura 26. Nel capitolo verranno trattate nel dettaglio le operazioni descritte nel diagramma a blocchi, seguendo prima lo schema per la decisione sulla appartenenza di una proteina al gruppo delle solenoidi oppure all'insieme delle proteine globulari, poi trattando l'argomento dei periodi di ripetizione. La prima parte, in cui si descrive la maniera in cui il segnale viene estratto da un allineamento Blast, è comune sia alla sezione in cui avviene la separazione tra le due classi di proteine trattate, sia alla sezione sulle lunghezze di ripetizione. Quando si parla di segnale elaborato dai vari tool creati per Ouroboros, si assume quindi un segnale pulito, estratto con le metodologie indicate nel blocco principale, ovvero un profilo modificato mediante i fattori di Atchley e filtrato per mezzo di una convoluzione. Il blocco a sinistra, indicante la fase di identificazione, viene presentato espandendo le sottosezioni, dalla funzione SortAndTag, all'uso di un parametro di profondità per residui con tag, al conteggio finale di uno score decisionale. Il blocco a destra, riguarda l'elaborazione effettuata sul segnale, mediante approssimazione per triangoli, calcolo di aree e assegnazione di tags, utilizzato per individuare la dimensione dei periodi di ripetizione.

4.1 Estrazione del segnale utile

Dal dataset, composto in totale di 247 proteine, suddiviso in training set e test set, sono generati degli allineamenti multipli di sequenze con PSI-BLAST. La classe di Victor deputata alla lettura di output blast, `loadBlastMode6` di Align, permette l'inserimento come input di questi allineamenti, mentre la classe `Profile`, chiamata successivamente, rende possibile la creazione di un profilo. Un profilo non è altro che una matrice, di dimensione 20*L, ove L è la lunghezza della sequenza, che include nelle sue colonne le probabilità per ognuno dei 20 amminoacidi di comparire in quella posizione. Alla colonna i, pertanto, avremmo 20 valori di probabilità, compresi cioè tra 0 e 1, e la cui somma è pari a 1, rappresentanti quante volte nell'allineamento, ciascuno dei 20 amminoacidi compare in i in alternativa all'attuale amminoacido di sequenza. La matrice del profilo così ricavata, con i valori di probabilità appena descritti, viene quindi combinata con la matrice di Atchley, di

dimensione 20*5, in modo da ottenere una matrice risultante di dimensioni 5*L, composta cioè dai 5 attributi di Atchley nelle righe, e dai residui nelle colonne.

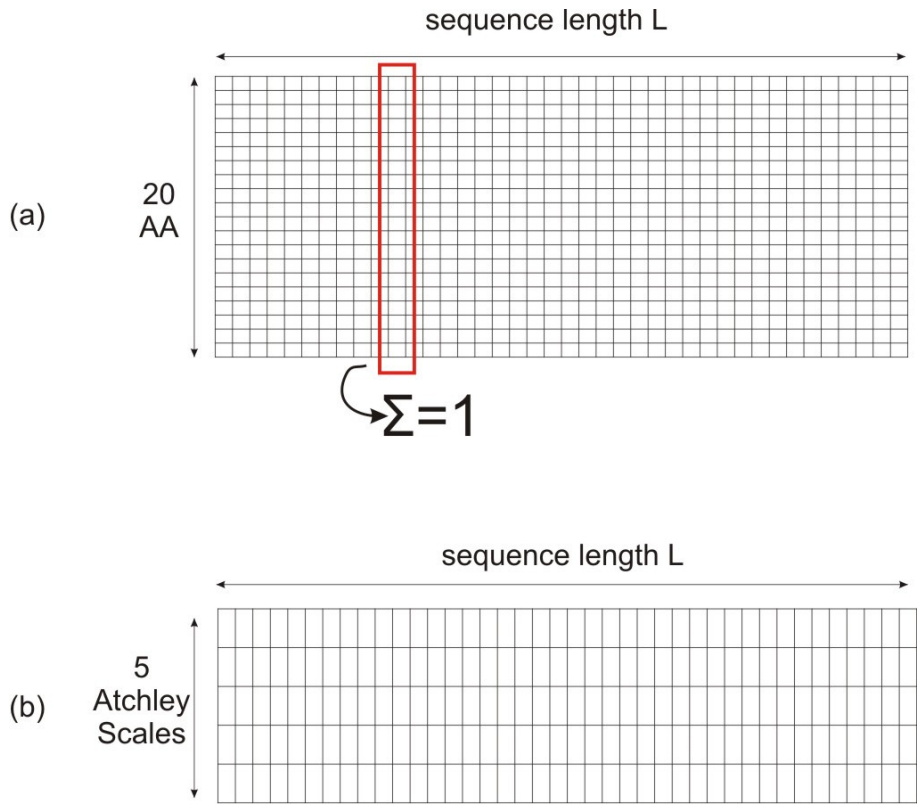


Figura 27: a) la matrice contenente un profilo ricavato da una allineamento Blast. b) La matrice risultante per ogni sequenza, che combina i fattori di Athcley con le probabilità per ogni singolo amminoacido

4.2 Filtraggio del segnale

Il segnale ottenuto da profilo e da attributi di Atchley si presenta nell'aspetto come un segnale rumoroso, in cui è difficile riconoscere delle componenti analizzabili alla prima ispezione visiva. Una operazione di filtraggio si è resa necessaria, occorrendo per le successive operazioni un segnale pulito.

Un metodo molto semplice per ottenere un filtraggio adeguato ai nostri scopi è effettuare una convoluzione del segnale con un kernel.

La convoluzione è una operazione tra funzioni, utilizzata spesso come un metodo di filtraggio, soprattutto per sistemi lineari tempo-invarianti.

Per funzioni discrete, siano esse f(n) e g(n) : R ⟶R, la convoluzione è definita da

$$(f * g)(m) \coloneqq \sum_{n} f(n)g(m-n)$$

Ma, poiché la lunghezza del segnale con la convoluzione normale varia, si è preferito usare la convoluzione circolare, definita da:

$$x_N[n] * h[n] = \sum_{m=-\infty}^{\infty} h[m] \cdot x_N[n-m] = \sum_{m=-\infty}^{\infty} (\, h[m] \cdot \sum_{k=-\infty}^{\infty} x[n-m-kN])$$

con x(n) e h(n) anch'esse discrete. Il risultato è un segnale filtrato della medesima lunghezza del segnale di partenza.

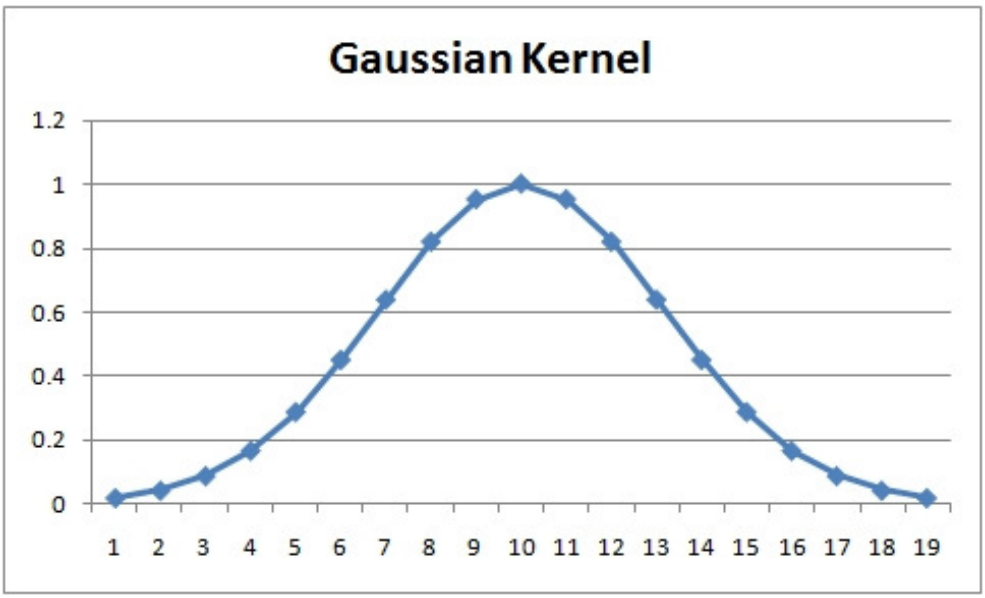


Figura 28: Kernel Gaussiano, simile a quello utilizzato per il filtraggio.

Vari tentativi sono stati impiegati, da una preventiva fase di ottimizzazione mediante grid-search, e un buon kernel è risultato essere quello gaussiano, con dimensione fissata pari a 11 unità, σ pari a 7, e l'utilizzo della convoluzione circolare, che permette come è stato in precedenza introdotto, di mantenere le dimensioni del segnale originario. I risultati di un segnale rumoroso e il filtraggio con convoluzione sono visibili in figura 30

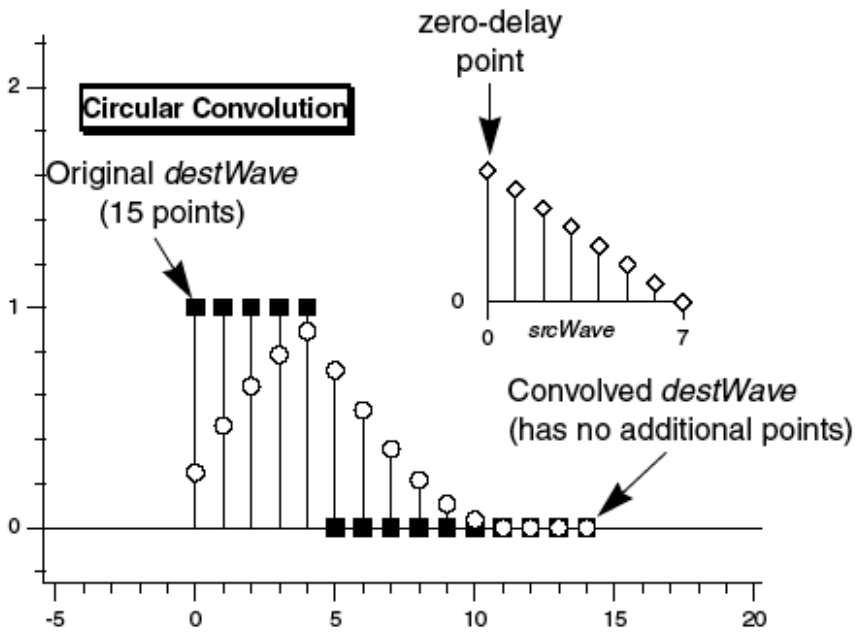


Figura 29: schema illustrante la convoluzione circolare

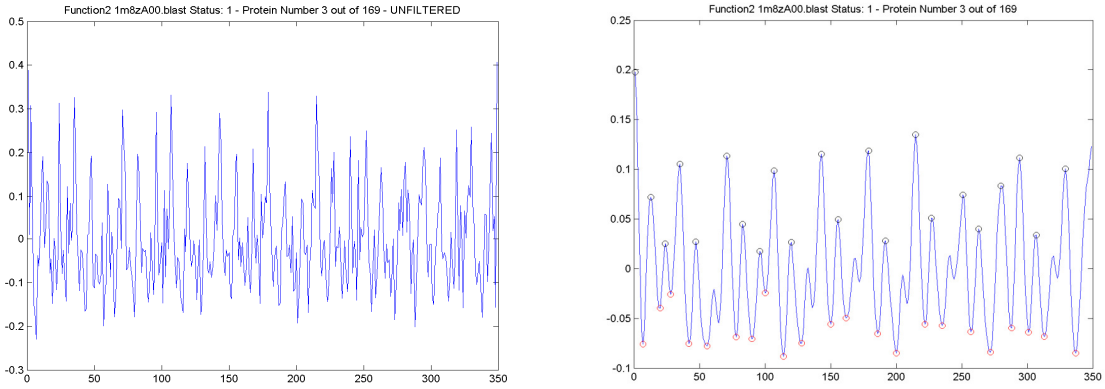


Figura 30: Risultato del filtraggio mediante convoluzione della funzione 2, proteina 1m8z del training set. A sinistra, funzione prima del filtraggio, con profilo rumoroso. A destra, risultato ottenuto convolvendo con un kernel gaussiano di dimensione 12.

4.3 Etichettatura dei residui

La funzione `SortAndTag` effettua la prima classificazione dei residui, come appartenenti ad un gruppo specifico, in base all'ampiezza del loro valore in scala di Atchley e come risultato del variare di un parametro chiamato "segment". Il primo parametro da ottimizzare è in effetti una distanza, che

serve a suddividere il risultato di un sort dei valori delle sequenze: una volta ordinati i valori attribuiti ad ogni residuo dal più elevato al più basso, mediante segment effettuiamo una separazione del sort.

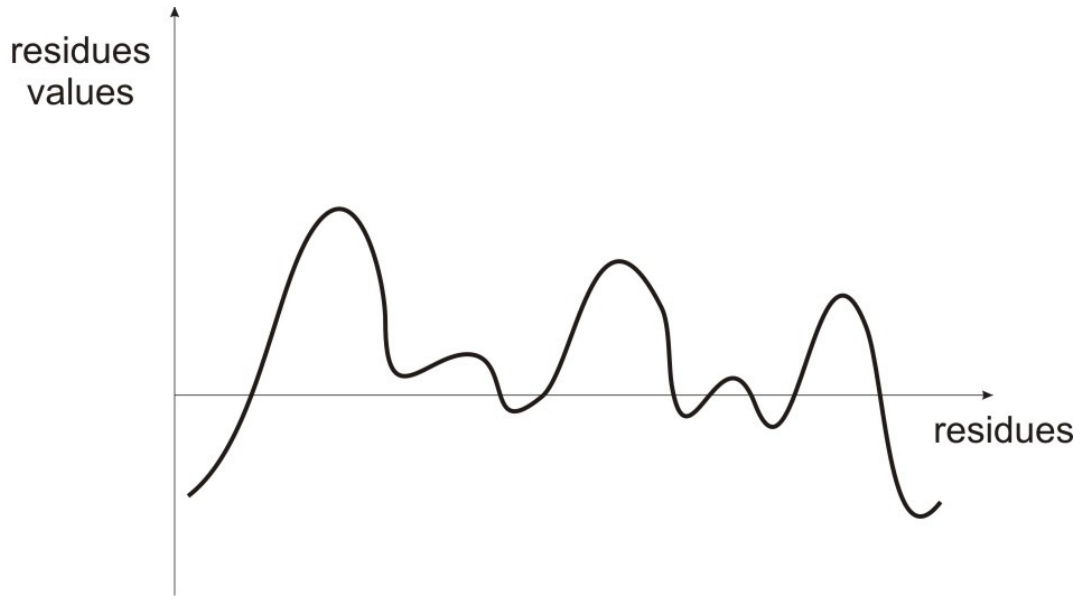


Figura 31: Rappresentazione sintetica del segnale di partenza, ricavato dai valori di Atchley

A ciascun residuo, che cade in un intervallo definito dalla lunghezza di "segment", viene assegnata una etichetta, o tag, in modo da renderlo afferibile ad un univoco insieme. Il tag assegna un simbolo (in questo caso si è adoperata una simbologia numerica) per ogni sottogruppo del sort.

La successiva operazione consiste nel ricostruire l'andamento precedente al sort, ma con i residui etichettati, in modo da ottenere una nuova sequenza, della stessa lunghezza originaria, ma composta da valori di tag in ordinata.

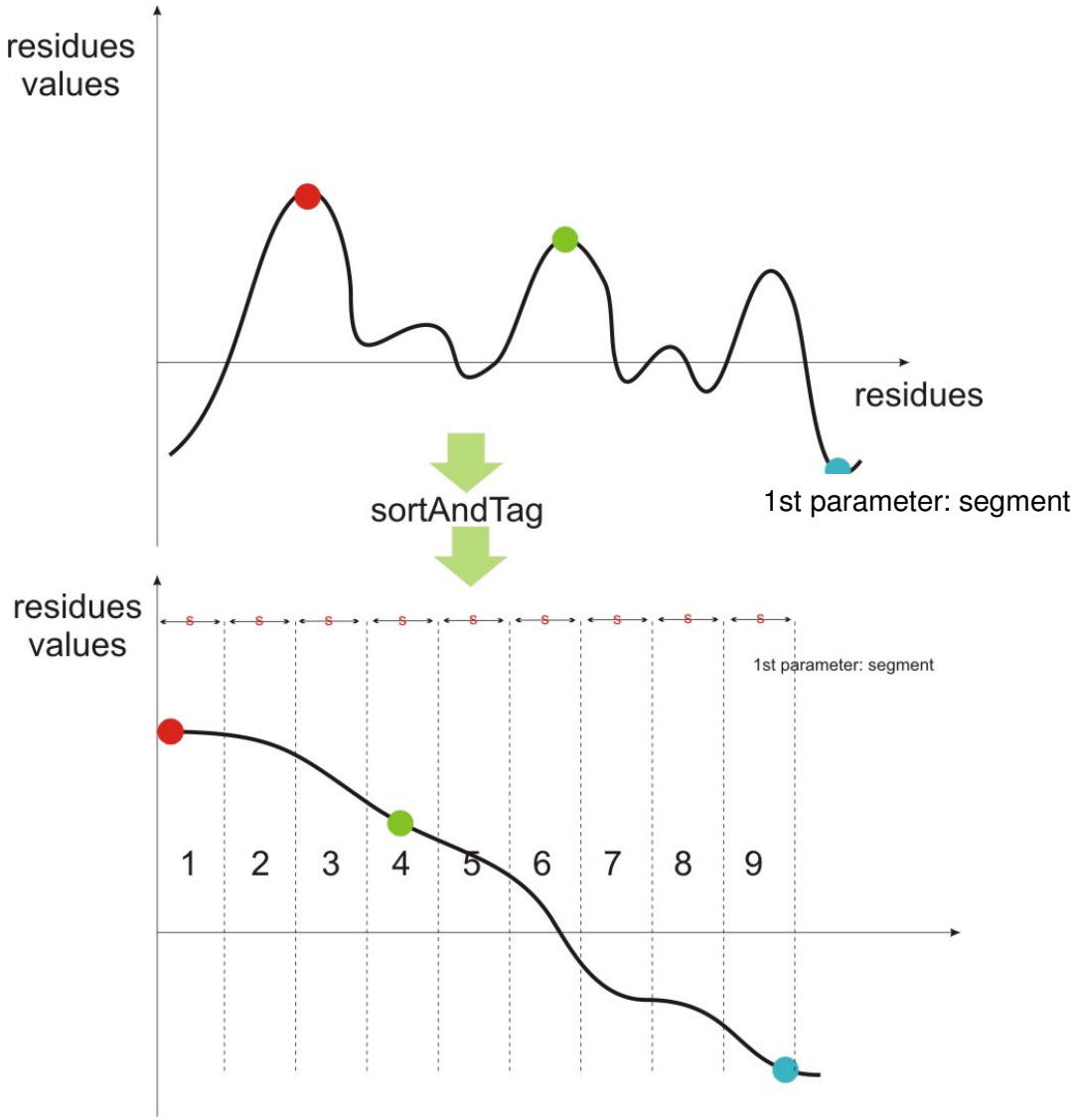


Figura 32: funzionamento di SortAndTag: dopo la "destrutturazione" della funzione in una sequenza ordinata, i residui vengono etichettati come indicato dal parametro segment, ovvero separati in intervalli uguali e forniti di TAG univoco.

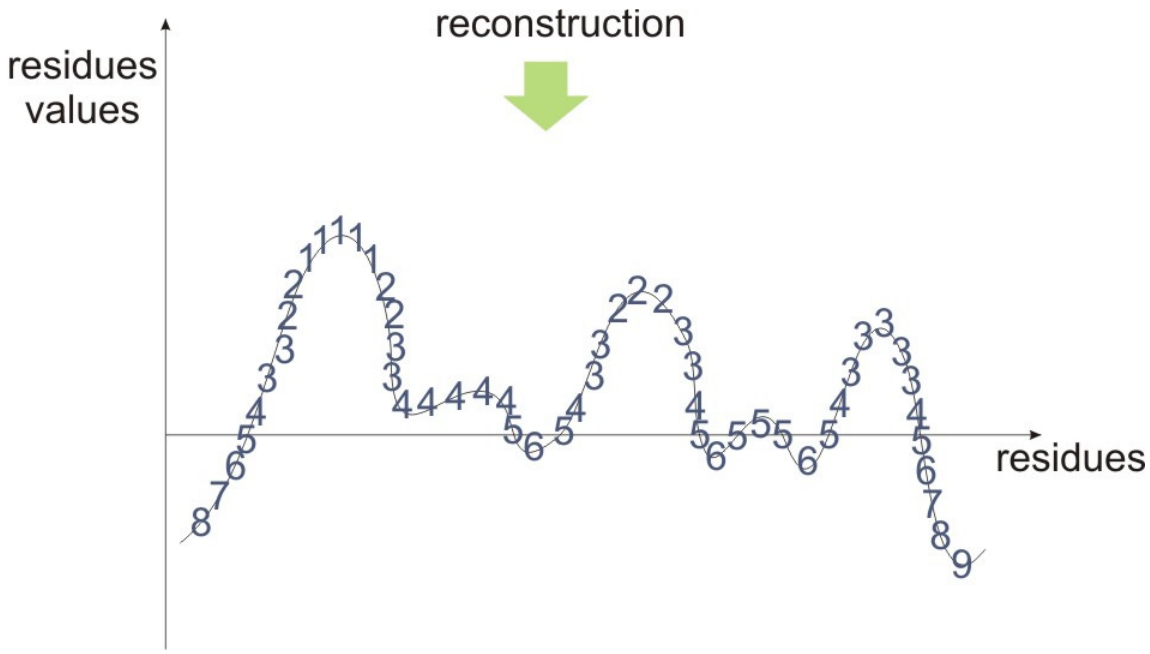


Figura 33: Ricostruzione della funzione con memoria dei TAGS. La ricostruzione è eseguita dopo l'operazione di etichettatura, e "ristruttura" la curva della funzione associata alla proteina.

4.4 Il parametro profondità

Ottenute le sequenze (in totale 5, una per ogni serie di attributi di Atchley) convertite in tags, Ouroboros utilizza un parametro depth, o profondità, in grado di decidere quale sia il valore di tag più rappresentativo per una proteina ripetuta.

...____1111____1111_____1111____1111___111_____1111...

..._222___222_____222__22____222____22____222__22___...

(...)

...666__66___666___6666__66_6_____6666___666__66____...

Figura 34: sequenze di TAGS isolate

In questo passo dell'algoritmo, il software non fa altro che selezionare coppie di sequenze di tag, selezionate da depth: alla profondità 1, Ouroboros seleziona il prima e l'ultimo insieme di tag

ottenuto. A profondità 2, selezionerà il secondo gruppo di tag e il penultimo, fino ad esaurire il numero massimo di tag disponibile.

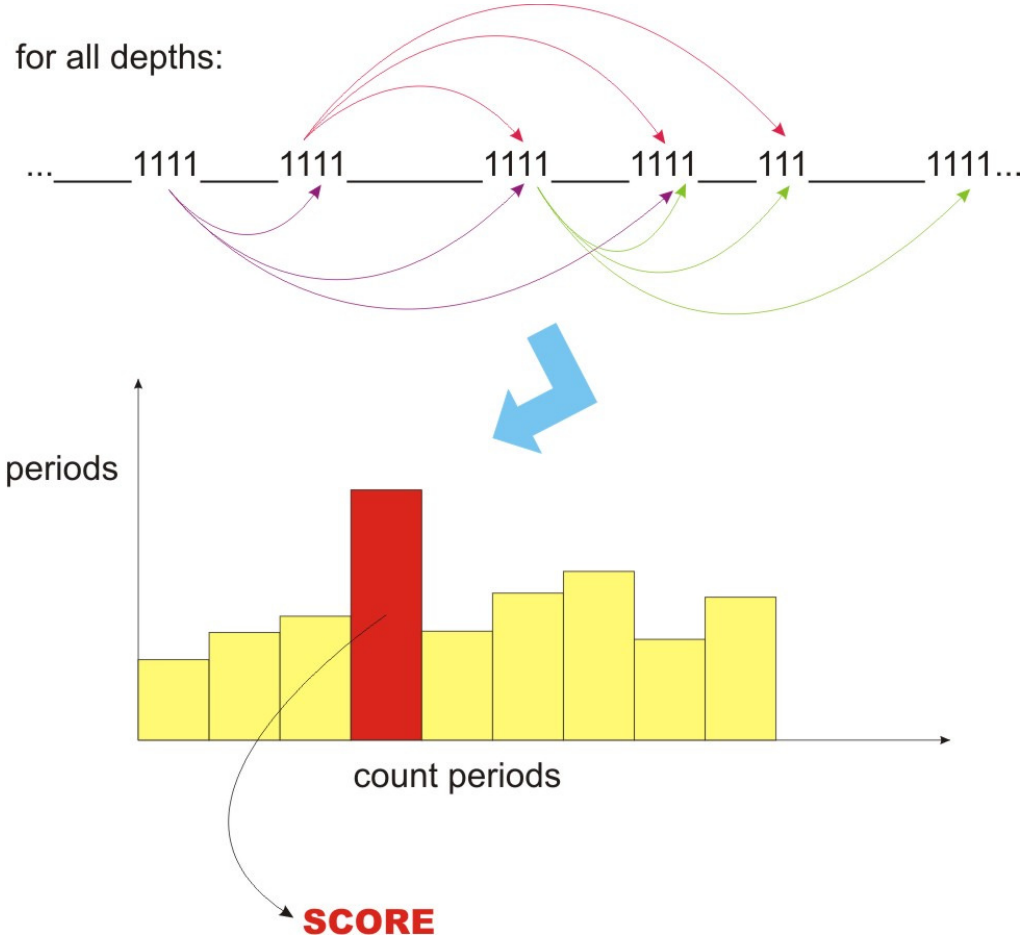


Figura 35: conteggio delle frequenze dei periodi e score per ciascuna proteina. Viene eseguito per ogni profondità, considerando, cioè ad ogni passo, due sequenze simultaneamente. In esempio è riportata la generica situazione con sequenze di TAG 1.

Per tutte le profondità rilevate, si computano le distanze (maggiori di una unità) che separano tag uguali. Viene infine creata una rappresentazione della distribuzione delle distanze, dalla quale si estrae uno score finale, pari alla frequenza della distanza che compare più volte nella distribuzione.

4.5 Soglia di decisione e ottimizzazione

L'ultimo parametro utilizzato in fase di machine training, per ottimizzare il riconoscimento di Ouroboros, è una soglia decisionale, o decision threshold, argomento per cui uno score ricavato dall'algoritmo può assegnare l'appartenenza di una proteina all'insieme delle proteine solenoid o meno. L'intero procedimento di ottimizzazione è stato condotto con la strategia del Simulated Annealing, un procedimento euristico per la ricerca di ottimo globale.

4.6 Simulated Annealing

E' una strategia euristica risolutiva per problemi di ottimizzazione: date tutte le soluzioni di un problema, l'obbiettivo è quello di trovare la soluzione migliore. In presenza di minimi locali, ovvero soluzioni possibili, il simulated annealing mira a trovare il minimo globale: esso rappresenterà la soluzione ottima. Il nome è mutuato da una procedura di trattamento termico per metalli, e indica il processo di eliminazione di difetti reticolari da cristalli mediante riscaldamento a temperature inferiori a quella di fusione, seguito da una fase di lento raffreddamento. Il metodo è stato descritto in maniera indipendente da Scott Kirkpatrick, C. Daniel Gelatt and Mario P. Vecchi nel 1983, nonché da Vlado Černý in 1985. Rappresenta un adattamento dell'algoritmo Metropolis-Hastings, un metodo Monte Carlo per generare campioni di stato di sistemi termodinamici. Nel metodo SA, ogni punto dello spazio di ricerca è analogo a uno stato per un sistema fisico, e la funzione E(s), da minimizzare, è analoga a una energia interna del sistema in quello stato. Lo scopo è quello di portare il sistema da un arbitrario stato iniziale allo stato con la minima possibile energia.

La probabilità di eseguire una transizione dallo stato corrente (s) al nuovo stato (s') è specificata da una funzione di probabilità, P(e,e',T), che dipende da e=E(s) e e'=E(s'), energie dei due stati, e da una variabile globale tempo variante T, chiamata, sempre in relazione alla procedura di *annealing (ricottura metallurgica)*, *temperature*.

Gli step dell'algoritmo SA possono essere schematizzati come segue:

1. Viene scelta una T iniziale, arbitraria:
    1. Si ricercano le soluzioni per il problema (da 50 a 100 soluzioni)
    2. Per ogni possibile soluzione se ne calcola il costo

3. Si prende il ΔE_{max} = e – e'
4. A questo punto si prende una $T_{iniziale} > \Delta E_{max}$ ma dello stesso ordine di grandezza;

2. Viene abbassata la temperatura fino a raggiungere un valore prossimo allo 0;
3. In prossimità del minimo valore di T si troverà un minimo (di energia) abbastanza forte;
4. Ripetendo questo ciclo la possibilità di trovare la stessa soluzione è tendente a 0. Se vengono trovate due soluzioni uguali per due prove diverse dello stesso problema significa che, molto probabilmente, qualcosa non funziona correttamente.

La temperatura della rete viene definita in modo che:

1. Se T è elevata: Ci si può permettere di fare salti alti e quando si trova un minimo si può provare a proseguire per scoprire se si trattava solamente di un minimo locale;
2. Se T è bassa: Si possono ancora fare dei salti alti ma con minore probabilità quindi si procede a passi più corti;
3. Riduzione veloce di T: Implica il congelamento di alcune fluttuazioni termiche;
4. Riduzione molto lenta di T: Può implicare il non raggiungimento della conclusione del calcolo e quindi il non trovare un minimo globale.
5.

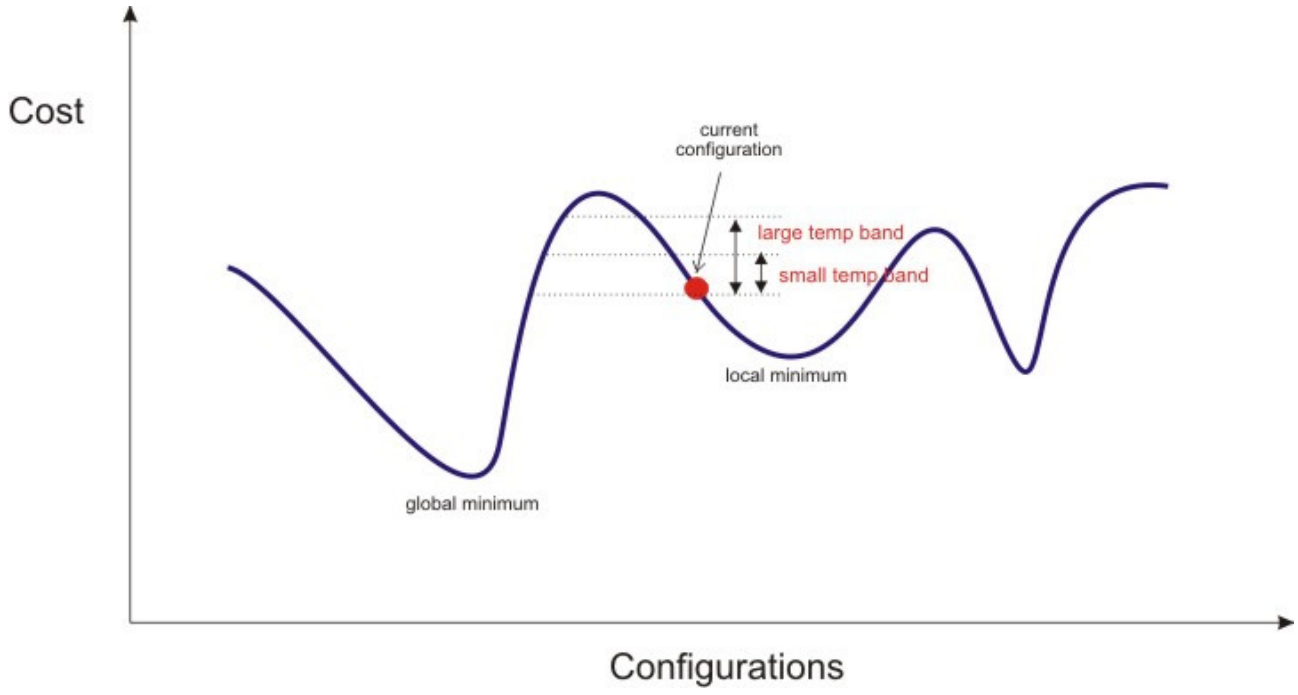


Figura 36: Simulated Annealing per una configurazione monodimensionale

```
simulated_annealing_alg(Tstart, Tend, Tstep, Istep)
{
 for (T=Tstart; T > Tend; T -= T*Tstep)
    for (I = 1; I < Istep; I = I + 1) {
       trans_data = select_transform();
       if (test_transform(trans_data)) {
          delta_E = estimate_transform(trans_data, cost_function);
          if (delta_E >= 0 || exp(delta_E/T) > rand())
             perform_transform();
          }
       }
}
```

Figura 37: Pseudocodice per Simulated Annealing

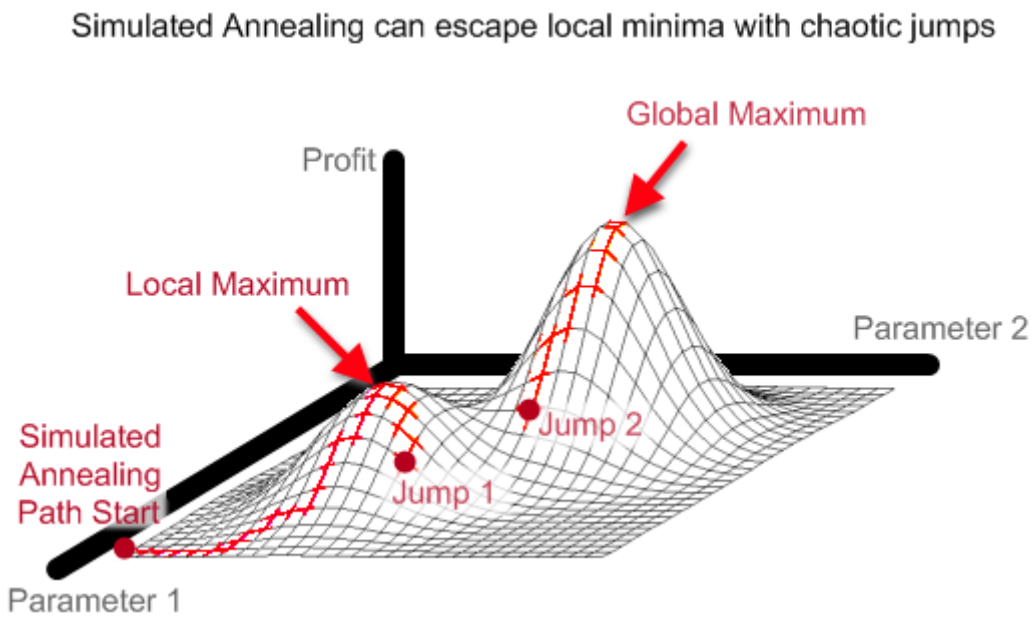


Figura 38: Illustrazione raffigurante un porcesso di Simulated Annealing su due dimensioni. L'algoritmo compie letteralmente dei "salti" tra valori massimi (o minimi) locali, fino ad ottenere l'ottimo globale. E' un tipo di ottimizzazione "hill climbing".

4.7 Periodi di ripetizione

Per quanto riguarda il periodo delle ripetizioni delle proteine, le funzioni ricavate nella fase di estrazione del segnale sono state impiegate da un diverso punto di vista. Per le sequenze filtrate, sempre con il metodo della convoluzione circolare, vengono evidenziati i picchi (massimi e minimi locali), mediante un tool appositamente scritto in C++, peakdet, che permette di decidere anche l'altezza minima dei picchi da selezionare. La selezione dei picchi è tarata sulla standard deviation del segnale, in modo da ottenere durante una fase di training la migliore combinazione di estremi locali, per poter procedere alla seconda fase di analisi. Il primo parametro, quindi, rappresenta un coefficiente moltiplicativo per standard deviation.

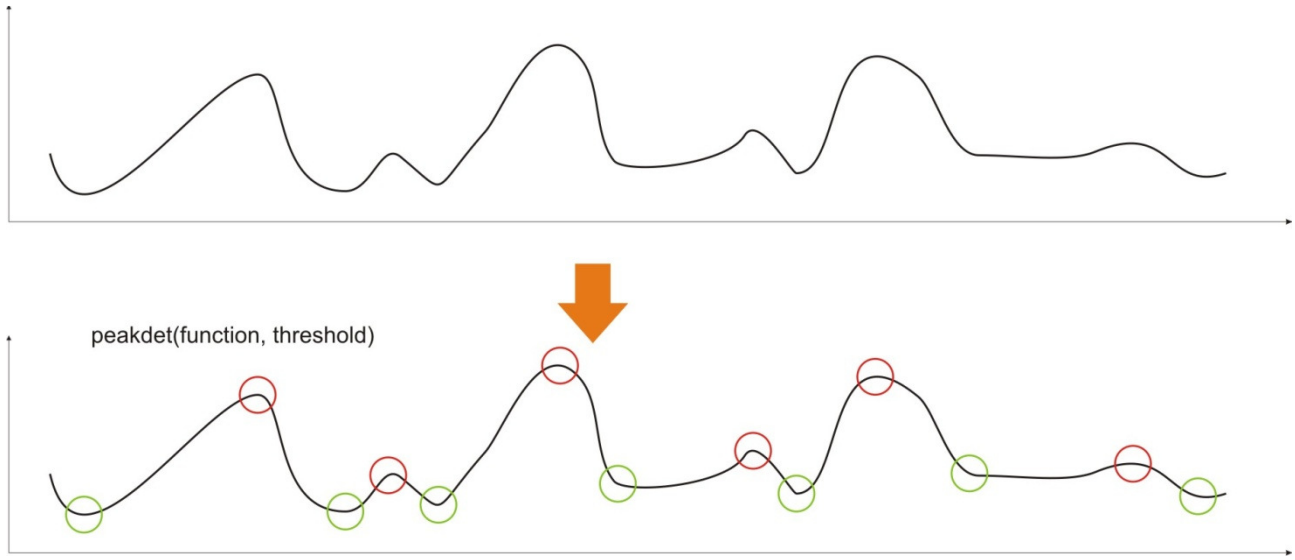


Figura 39: rilevazione dei massimi e minimi con peakdet. Peakdet rivela massimi e minimi a meno di un valore di soglia specificato. Ciò permette di selezionare solo i picchi che risultano avere una distanza maggiore o uguale alla soglia rispetto al picco precedente.

Una volta calcolati i massimi e i minimi locali, si passa alla generazione di una "mappa" in cui sono contenute solo le posizioni e i valori corrispondenti a questi estremali relativi, e dalla mappa viene riprodotta una approssimazione del segnale mediante triangoli (ciascuno compreso tra due minimi, con un punto di massimo locale come vertice). In questo modo, è possibile ottenere una versione semplificata del segnale. Il codice che permette la traduzione dei punti di estremo locale in triangoli si chiama Trinity.

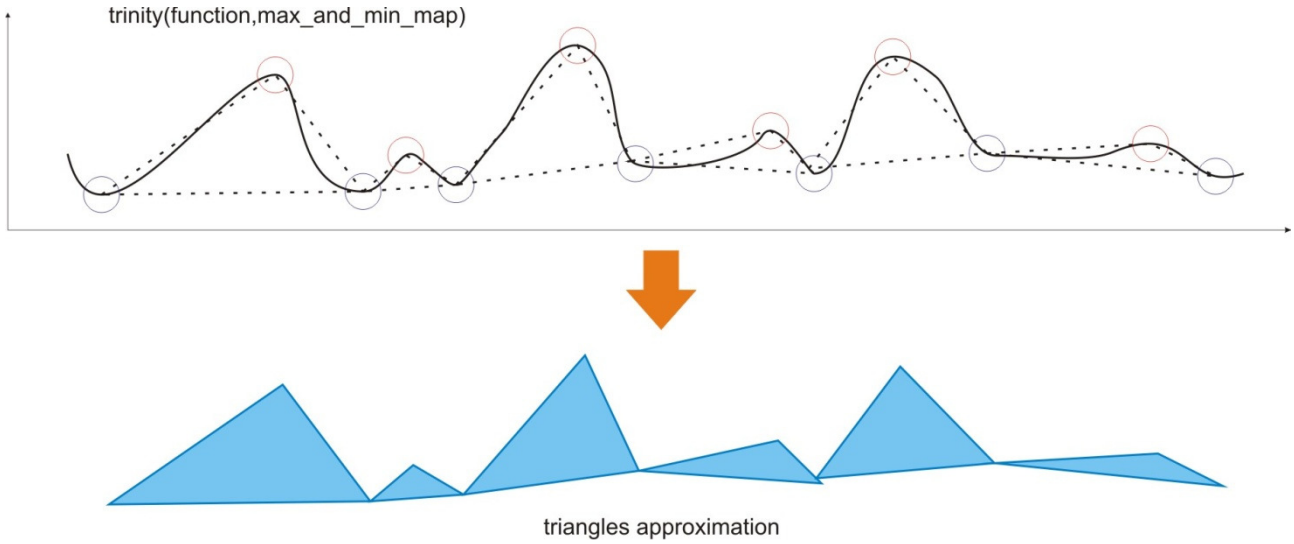


Figura 40: Approssimazione di una curva di funzione mediante triangoli. Trinity esegue questa operazione, e calcola le aree relative ai triangoli. Una approssimazione di aree di questo tipo serve a valutare la presenza di comportamenti simili all'interno della sequenza.

Solo dopo avere ottenuto i triangoli, una semplice funzione, denominata Xanthippe, effettua il calcolo dell'area di ciascun triangolo, quindi li compara e, a meno di una soglia, etichetta con tags uguali le aree uguali. Il secondo parametro ottimizzando è questa soglia che permette di determinare se due areee sono somiglianti.

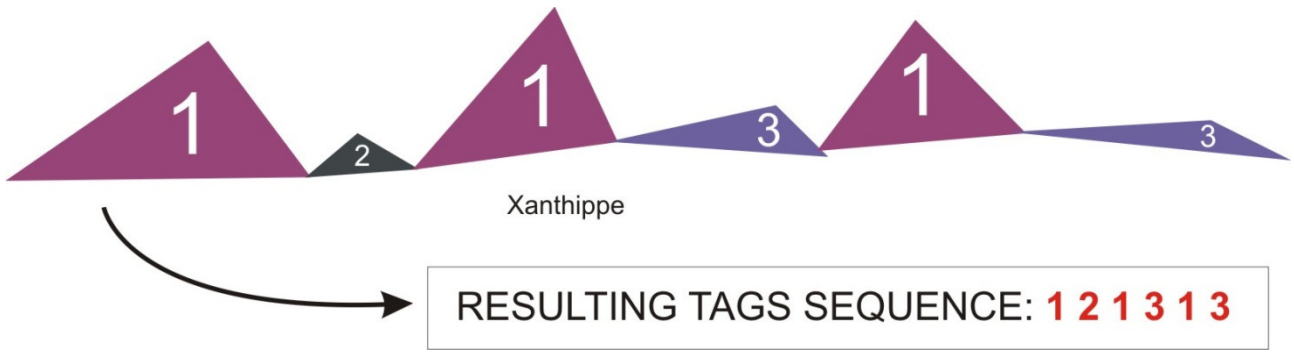


Figura 41: Etichettatura delle aree con Xanthippe. Ad aree uguali corrisponderanno TAG uguali, a meno di un valore di soglia (tolleranza per la somiglianza fra le aree)

Infine, una utility chiamata Radix permette semplicemente di risalire dalle aree ai residui estremali (residui che in origine erano i massimi e minimi locali). Selezionando l'etichetta associata alle aree che compare più frequentemente, e calcolando una media delle distanze che separano aree uguali, questa dovrebbe essere uguale o molto vicina al periodo di ripetizione corretto. In caso di inserzioni, quindi di comportamenti anomali tra i periodi di repeat, dovute a comportamenti strutturali che deviano dai pattern di ripetizione, si cerca di ricostruire la struttura originaria del repeat, coprendo cioè i buchi di informazione con valori fittizi simili ai valori ottenuti per tutte le situazioni regolari. Il software è allenato su un training set di 50 solenoidi, già presenti all'interno del training set per la discriminazione tra proteine ripetute e non ripetute, e successivamente testato su un test set di 55 proteine solenoidi (ovviamente, anche queste sono le proteine solenoidi afferenti, questa volta, al gruppo del test set precedentemente utilizzato).

In totale si andranno a valutare, per ciascuna proteina e al variare dei parametri sopra elencati, quanti periodi corretti si riscontreranno, quanti mezzi periodi verranno invece identificati e anche quanti periodi con il doppio del valore del vero repeat saranno ottenuti. Un terzo parametro, delta, serve a allargare la forchetta per i risultati ottenuti, assumendo che si possa erroneamente identificare il periodo con uno scarto di alcuni residui.

CAPITOLO V

Risultati e discussione

Il benchmarking di Ouroboros è stato condotto su un dataset totale di 247 proteine, suddiviso in un training set di 169 proteine (50 solenoidi e 119 non solenoidi), e un test set di 183 elementi (55 solenoidi e 128 non solenoidi). I test sono stati condotti principalmente sulla rete del Biocomputing UP, data le lunghezze delle tempistiche di esecuzione per testare tutti i risultati possibili. In totale, la fase di ottimizzazione di Ouroboros ha richiesto un tempo pari al tempo di progettazione e di programmazione. Molte delle intuizioni che hanno reso possibile i buoni risultati di Ouroboros sono state sviluppate quando la fase di machine learning era in progresso: mano a mano che le problematiche più ostiche venivano individuate, si introducevano soluzioni che potessero correggerle. Scopo di questo capitolo è intuire questi problemi, tipici delle fasi di benchmarking e presentare i risultati finali, anche se, al momento della stesura di questa tesi, i test su Ouroboros continuano per cercare di ottenere la migliore configurazione.

5.1 Ottimizzazione e variabili

Nella fase di machine learning e di testing, si è preferito massimizzare un coefficiente denominato *Matthew's correlation coefficient*, ovvero un valore utilizzato largamente in machine-learning per misurare la qualità di una classificazione binaria (a due classi). Il *Matthew's correlation coefficient* considera veri positivi, veri negativi, falsi positivi e negativi. Varia dal valore −1 a +1, dove +1 rappresenta una perfetta predizione, 0 una predizione media random, cioè non correlata con i dati osservati, e −1 una predizione inversa. E' utilizzato soprattutto quando due classi hanno dimenzioni molto differenti, ove un criterio basato sull'*accuracy* (predizioni corrette) non sarebbe abbastanza rappresentativo. MCC può essere ricavato mediante la formula:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Dove TP è il numero dei veri positivi, TN veri negativi, FP falsi positivi e FN falsi negativi.

Accanto alla valutazione per l'MCC, si sono registrati i corrispettivi valori di sensitività e specificità. La sensitività è così descritta:

$$Sensitivity = \frac{number\ of\ TRUE\ POSITIVES}{number\ of\ TRUE\ POSITIVES + number\ of\ FALSE\ NEGATIVES}$$

Una sensitività del 100% significa la perfetta identificazione dei reali valori positivi, mentre per la specificità si assume questa definizione:

$$Specificity = \frac{number\ of\ TRUE\ NEGATIVES}{number\ of\ TRUE\ NEGATIVES + number\ of\ FALSE\ POSITIVES}$$

Se la specificità raggiungesse il 100%, si avrebbe una perfetta identificazione per tutti i valori negativi del dataset.

In un primo momento si è proceduto ad una ricerca esaustiva dei parametri mediante grid-search, ovvero testando le varie combinazioni di parametri in modo da garantire l'ottimalità della scelta. La presenza di molti minimi locali nel range di ricerca dei parametri ha però reso necessario l'utilizzo di un criterio come il *Simulated Annealing* per cercare di ricavare la migliore combinazione parametrica.

5.2 Discriminazione solenoide – non solenoide

Le cinque funzioni di Atchley sono state ottimizzate separatamente, ottenendo valori eterogenei per i termini di confronto usati. In particolare, si può notare un sostanziale sbilanciamento tra le funzioni di Atchley per quanto riguarda il risultato. Le funzioni 1 e 2, nel processo di identificazione delle proteine ripetute, sembrano essere molto più informative rispetto alle funzioni 3 e 4 e 5.

	Training Set	Test Set	Parameters Values
F1	MCC: 0.66 SENS: 0.64 SPEC: 0.95	MCC: 0.69 SENS: 0.66 SPEC: 0.97	Dec: 103 Segment: 0.0456 Depth: 2
F2	MCC: 0.65 SENS: 0.60 SPEC: 0.97	MCC: 0.49 SENS: 0.53 SPEC: 0.91	Dec: 55 Segment: 0.0282 Depth: 3
F3	MCC: 0.35 SENS: 0.40 SPEC: 0.90	MCC: 0.43 SENS: 0.38 SPEC: 0.95	Dec: 36 Segment: 0.0189 Depth: 3
F4	MCC: 0.45 SENS: 0.32 SPEC: 0.98	MCC: 0.36 SENS: 0.31 SPEC: 0.95	Dec: 255 Segment: 0.0388 Depth: 3
F5	MCC: 0.30 SENS: 0.42 SPEC: 0.86	MCC: 0.34 SENS: 0.51 SPEC: 0.82	Dec: 161 Segment: 0.0285 Depth: 4

Tabella 2: risultati di Ouroboros per l'identificazione delle proteine solenoidi. Sono valutate le prestazioni in base al Matthew's correlation coefficient, sensitività e specificità.

Ciò rappresenta un comportamento che comunque era atteso: ricordiamo che le funzioni 3, 4 e 5 rappresentano rispettivamente dimensione molecolare, differenze nei codoni e carica elettrostatica, informazioni che apparentemente poco concorrono alla descrizione di una struttura terziaria. La funzione 1, che rappresenta la polarità, e la funzione 2 ovvero struttura secondaria, sembrano conservare e riprodurre meglio queste caratteristiche tipiche delle proteine solenoidi, e individuano una più netta soglia di discriminazione di queste noi confronti delle proteine globulari. Alcuni dei risultati ottenuti nella fase di *machine learning*, non solo sono interamente riprodotti nel test set, ma vengono migliorati: è il caso, ad esempio, della funzione 1, che sembra ottenere ottime stime per le

proteine solenoidi. Un'altra osservazione obbligatoria è lo sbilanciamento tra sensitività e specificità. In tutte e 5 le funzioni ottenute, la specificità non scende mai al di sotto del 90%, ciò dimostra una grande precisione nell'identificare come globulari proteine che in effetti lo sono, mentre la sensitività è molto variabile, ed è questo che genera queste differenze tra le stesse funzioni. Il *machine learning*, come metodo automatico, pone infatti una soglia decisionale ottima che massimizza significativamente il *Matthew's correlation coefficient*, ma non tiene conto del rapporto tra sensitività e specificità.

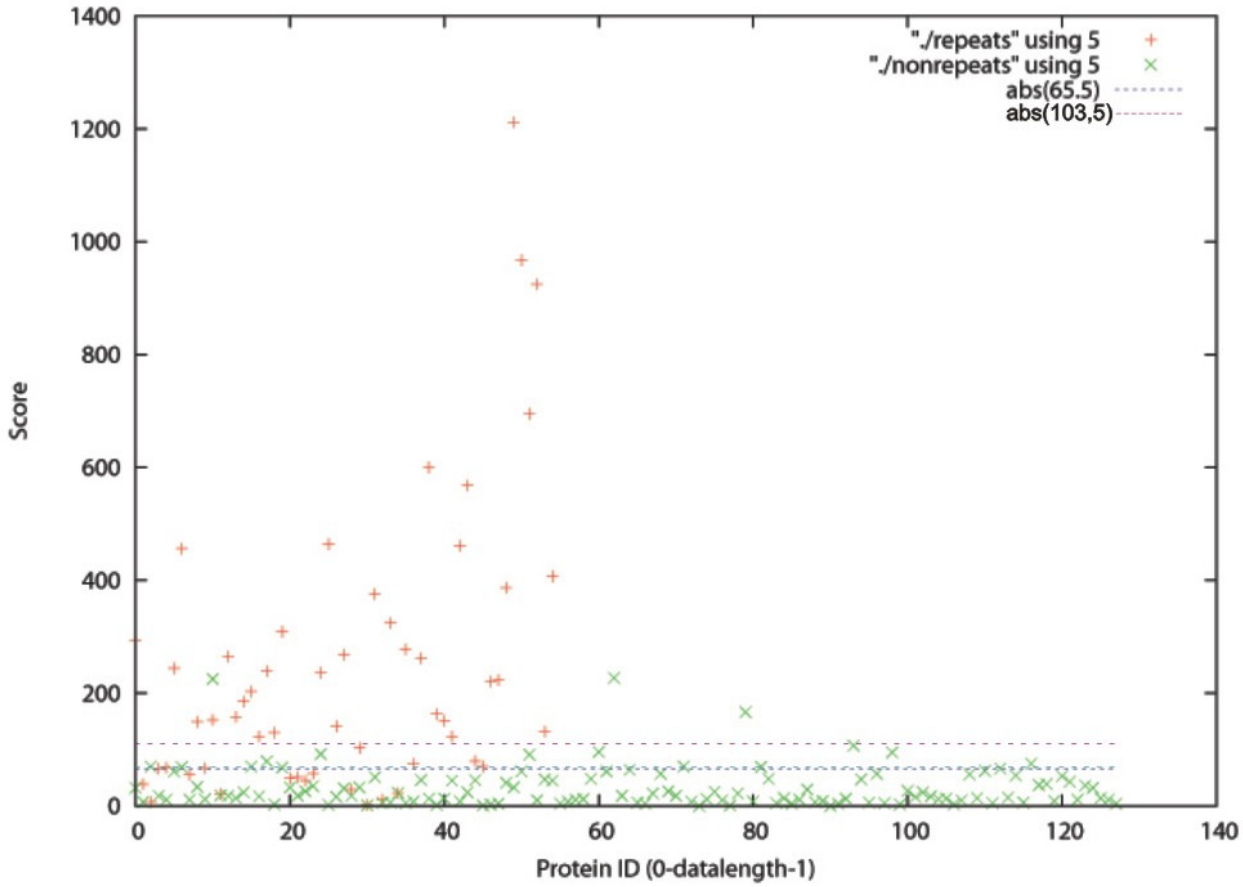


Figura 42: plot dei valori di score ottenuti per il training set. E' evidenziata la soglia di 103, suggerita dal benchmarking. E' inoltre presente una soglia più conveniente, pari a 65.5, che consente di bilanciare sensitività e specificità.

Nella illustrazione precedente, ottenuta con *Gnuplot*, è visualizzata la distribuzione dei valori di score con uno *scatterplot*, per la funzione 1 ed estratta dal training set. Una soglia al valore di 103 (unità di score) è evidenziata in viola, mentre la soglia più bassa è paria a 65.5. I valori ripetuti sono presentati con un + rosso, mentre le proteine non ripetute con una x verde. La separazione è netta, specie se

guardiamo alla altissima concentrazione di valori verdi sul fondo del grafico. Essi sono anche uniformemente distribuiti, mentre i valori rossi sembrano occupare di più la parte sinistra del grafico, e avere una distribuzione a nuvola, con valori di score mediamente più alti. La soglia di discriminazione fissata dal *machine learning* sul training set, ovvero 103, è abbastanza adatta ad una discriminazione tra le due classi, ma se dovessimo considerare un valore pari alla seconda soglia, cioè 65.5, riusciremmo ad alzare la sensitività, catalogando cioè alcune proteine che nella realtà sono solenoidi come davvero solenoidi, al costo di perdere alcune delle proteine globulari che sono quasi perfettamente riconosciute. Il problema dell'ottimizzazione è quindi qualcosa che va valutato al di là di un semplice risultato ottimale, può essere necessaria una correzione per ottenere i valori più convenienti, piuttosto che i valori ottimi.

Un altro problema affrontato, e parzialmente risolto grazie al preventivo filtraggio, è il comportamento "anomalo" di alcune proteine solenoidi. Questo comportamento, che può certamente inficiare la classificazione corretta delle proteine, può essere meglio compreso con qualche esempio. Nella figura 43, sono visualizzati due esempi di funzione 2 (struttura) per proteine ripetute: le proteine denominate con il codice PDB `1k1a` e `1vbl`. Si nota che la prima curva presenta chiaramente un comportamento periodico, mentre per la seconda proteina questa assunzione non può essere fatta altrettanto spontaneamente. La ragione più probabile per queste anomalie è da ricercarsi nella presenza di grandi zone di inserzione visibili nella struttura, come dimostra l'illustrazione precedente. Il problema delle inserzioni è un evidente ostacolo alla distinzione tra solenoidi e non solenoidi.

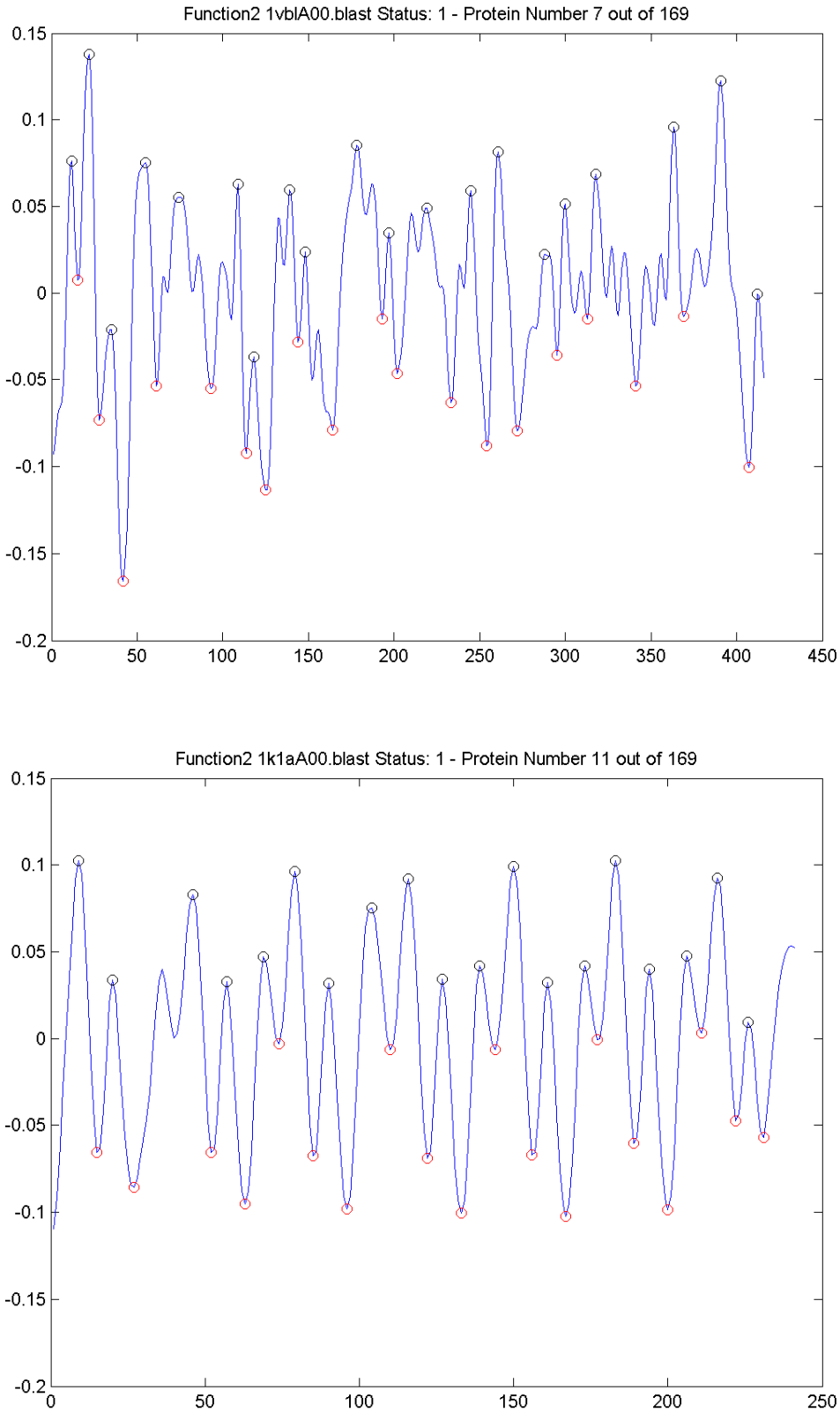


Figura 43: due plot di funzione 2 di proteine ripetute. In alto, il comportamento sinusoidale non è immediatamente riconoscibile. In basso, comportamento periodico

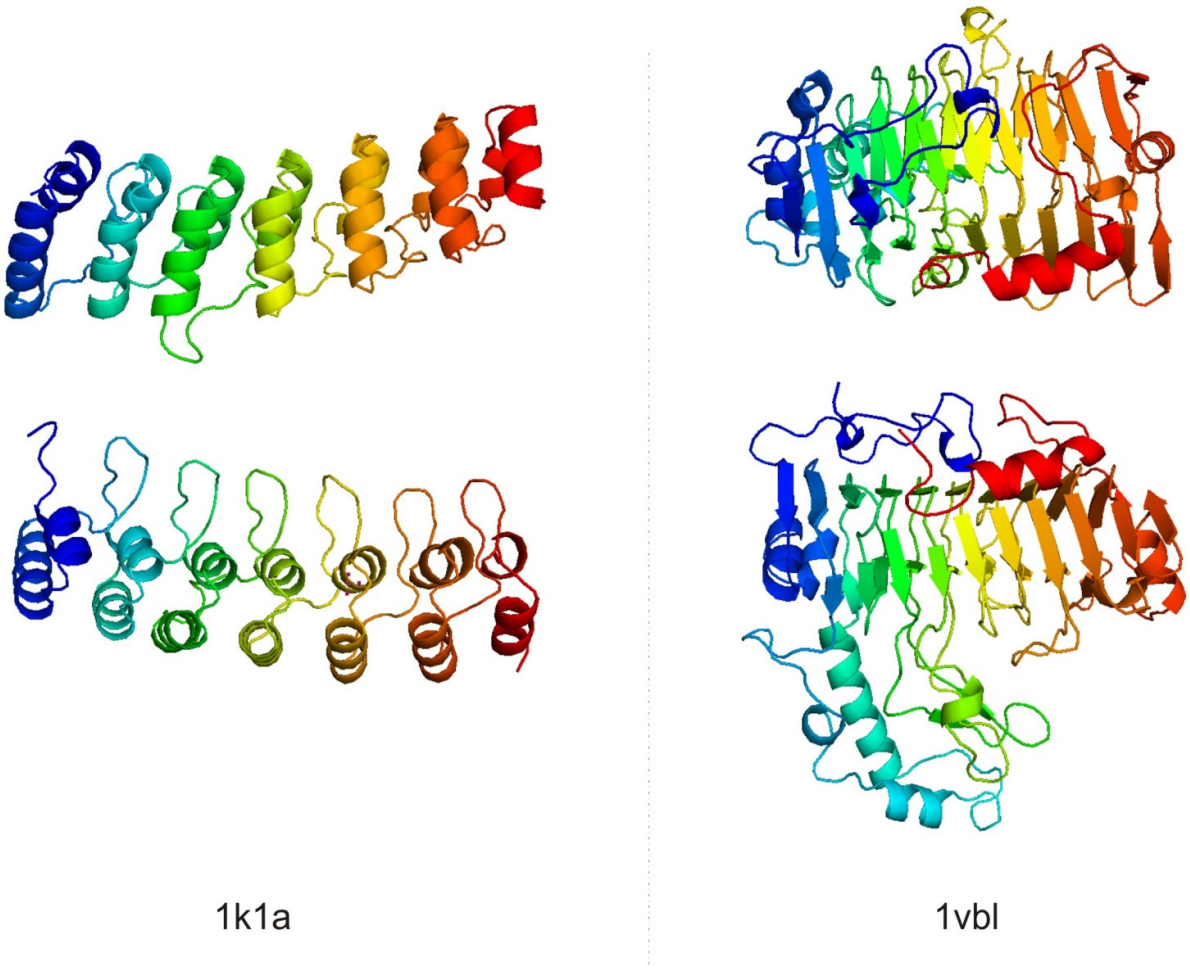


Figura 44: le due proteine prima visualizzate tramite le loro funzioni 2. la proteina con andamento periodico è decisamente più regolare anche nella struttura 3D. Vi sono importanti inserzioni per l'altra proteina, che comunque presenta comportamento solenoide (ripetizioni di Beta-strands)

Per quanto riguarda la comparazione di Ouroboros con altri metodi allo stato dell'arte, possiamo ad esempio prendere come termine di paragone Repetita, software già menzionato nel capitolo II. I risultati per quanto riguarda la discriminazione ripetuta-non ripetuta sono i seguenti (per Ouroboros è stata scelta la funzione 1, questa volta con la modifica della soglia decisionale a 65.5) :

	Training Set	Test Set
OUROBOROS	MCC: 0.58 SENS: 0.72 SPEC: 0.85	MCC: 0.64 SENS: 0.78 SPEC: 0.87
REPETITA	MCC: 0.54 SENS: 0.85 SPEC: 0.70	MCC: 0.51 SENS: 0.83 SPEC: 0.69

Tabella 3: risultati di Ouroboros (Funzione 1, soglia modificata) paragonati a quelli di Repetita. E' sempre privilegiato il confronto di MCC, ma è riportato comunque il valore di sensitività e di specificità.

Si tenga presente che Repetita è uno dei migliori esempi di software per l'individuazione delle proteine, in grado di surclassare nei risultati tutti i metodi che in precedenza erano stati proposti (Trust, RADAR, HHrepID). Inoltre, risulta facile confrontare Ouroboros con Repetita in quanto il dataset utilizzato per i due programmi è il medesimo. Come si può vedere, Repetita viene battuta in alcuni casi da Ouroboros. In termini di prestazioni, in assoluto la funzione 1 è la migliore, seguita dalla funzione 2.

5.3 Predizioni del periodo di ripetizione

Per quanto riguarda la lunghezza delle ripetizioni, e il periodo di ripetizione, Ouroboros ottiene i risultati seguenti:

	Training Set	Test Set
OUROBOROS - REPEATS	0.61	0.32

Tabella 4: risultati di Ouroboros per l'identificazione del repeat length

Sono considerate come corrette valori corrispondenti a ripetizioni intere, valori che identificano metà periodo e valori che ne considerano il doppio. Ciò che si può immediatamente considerare, osservando i dati numerici, è che il risultato ottenuto per il training set (con parametri: peakdet threshold = 1.9*STD(function), xanthippe threshold = 20) non si conserva per il test set.

Se confrontati con Repetita (che ottiene circa un 80% di valori corretti su test set) appare evidente che Ouroboros non determina sufficientemente bene la lunghezza del periodo di repeat. Ciò può essere dovuto a molteplici fattori. Sicuramente, il comportamento anomalo, che è stato già illustrato, corrompe notevolmente un criterio di ricerca delle lunghezze del repeat che si basa su un concetto di regolarità per proteina ripetuta. Un'altra causa di questo risultato può essere la notevole diminuzione di informazione comportata dall'approssimazione delle funzioni mediante triangoli.

Sarebbe utile comprendere, soprattutto per le versioni future di Ouroboros, se valutare non solo coppie di aree simili, ma pattern che si ripetono nella sequenza, porti a determinare una più corretta identificazione dei repeat, ciò potrebbe portare Ouroboros dalla situazione attuale, cioè un buon metodo discriminatorio per proteine solenoidi, ad un metodo di identificazione completo.

CAPITOLO VI

Conclusioni

In questa tesi è stato presentato un software programmato in linguaggio C++, che ha come scopo principale l'individuazione e la classificazione delle proteine solenoidi. Il progetto è partito come idea per il miglioramento e l'affinamento di REPETITA, precedentemente sviluppato dallo stesso laboratorio Biocomputing UP, presso il dipartimento di Biologia dell'Università di Padova. La fase di progettazione del software ha richiesto l'impiego di alcune delle metodologie più utilizzate per l'analisi sulle proteine, accanto a tools e funzioni di indipendente ideazione e programmazione.

Le soluzioni alle problematiche sono frutto di una attenta analisi dei dati, e spesso sono state adottate dopo aver testato più varianti, in modo da avere sempre un quadro chiaro sull'efficacia delle strategie risolutive. Alcune di esse, come ad esempio il filtraggio del segnale e la scelta del tipo di score, sono scelte fatte in seguito allo studio e alla comprensione del comportamento delle proteine solenoidi quando sono viste non dalla prospettiva di una struttura tridimensionale già nota, bensì da una semplice sequenza di dati sostituita alla struttura primaria, ovvero, in una più concisa definizione, da un segnale monodimensionale.

L'informazione portata da un segnale ad una sola dimensione, la sua analisi e la traduzione di essa nella predizione per una struttura tridimensionale è stata la questione capitale in questo progetto.

Essa ha richiesto non pochi sforzi per far sì che fosse intuita la sua natura, e se ne potessero estrarre dati utili alla suddivisione delle proteine in due classi distinte. Naturalmente, si è ancora lontani dal maneggiare queste conoscenze in modo perfetto: ne è un chiaro esempio la difficoltà nel ricavare un risultato migliore per i periodi di ripetizione. Tuttavia, l'aver dimostrato che i dati in nostro possesso sono potenzialmente utilissimi, e si prestano a molteplici tipi di manipolazione per ottenere ottimi risultati, rappresenta un punto di partenza molto significativo. Con gli sviluppi futuri, e l'adeguato tempo che serve all'indagine e al miglioramento di questo software, è certo che gli esperimenti confermeranno questa asserzione.

Da un altro punto di vista, Ouroboros ha rappresentato, per la mia esperienza personale e per il lavoro che gli è stato dedicato, una magnifica occasione per addentrarmi nel mondo dello studio e della ricerca scientifica. A partire dalla fase di preparazione, fino ai passaggi più importanti della programmazione informatica ed infine dei test eseguiti per valutare la validità scientifica del metodo adottato, ho appreso i fondamentali e imparato a gestire le basi di una architettura software: un'ottima occasione di formazione.

6.1 Miglioramenti futuri

La versione attuale di Ouroboros può definirsi "beta", in quanto le problematiche indagate non sono state risolte in modo completo: necessita, quindi, di apporti e migliorie che lo rendano stabile ed efficiente. Per prima cosa va riconsiderato il metodo di identificazione dei periodi di repeat. Per questa versione sono stati testati molte varianti per produrre una soluzione consistente, tuttavia non si può dire di avere sempre incrociato e testato ogni combinazione tra metodologie di separazione solenoidi/non-solenoidi e ricerca dei periodi. Alcuni test vanno quindi eseguiti in questo senso, almeno per valutare se la soluzione non sia a portata di mano, senza addurre ulteriori criteri di indagine. L'ottimizzazione del programma, al secondo punto, è stata effettuata considerando i 5 fattori di Atchley in maniera distinta. Bisognerebbe trovare una combinazione dei 5 fattori in modo tale da produrre compensazione laddove ve ne sia bisogno: una funzione potrebbe contenere informazioni che un'altra non possiede, sfruttare in questa maniera i dati potrebbe essere una soluzione più conveniente, e forse andrebbe anche a compensare i comportamenti irregolari di alcune proteine. Infine, le lunghezze dei repeats sono sicuramente di difficile derivazione in quanto sono

corrotte dalle inserzioni presenti nelle strutture delle proteine. Le inserzioni sono dei disturbi alla regolarità del periodo. Trovare un metodo efficace per scovare le inserzioni, e magari eliminarle dalla valutazione complessiva, potrebbe essere la chiave di volta per un software dalle prestazioni notevoli.

APPENDICE A

Risultati del Benchmarking

Training set, 50 Solenoidi, 119 Non-Solenoidi

Protein File	Actual Status	Prediction	SCORE
2grs001.blast	NON-REPEAT	REPEAT	319
1murA02.blast	NON-REPEAT	REPEAT	99
1m8zA00.blast	REPEAT	NON-REPEAT	25
1gw5A00.blast	REPEAT	REPEAT	812
1p5qA02.blast	REPEAT	NON-REPEAT	1
1r6bX04.blast	NON-REPEAT	REPEAT	448
1vblA00.blast	REPEAT	NON-REPEAT	43
2ca6A00.blast	REPEAT	NON-REPEAT	8
1kh3C02.blast	NON-REPEAT	NON-REPEAT	44
1bh5D00.blast	NON-REPEAT	NON-REPEAT	1
1k1aA00.blast	REPEAT	REPEAT	391
1n2kA01.blast	NON-REPEAT	NON-REPEAT	19
2bm3A00.blast	NON-REPEAT	REPEAT	190
1n11A00.blast	REPEAT	NON-REPEAT	3
1j2zA01.blast	REPEAT	NON-REPEAT	40
3pcnN00.blast	NON-REPEAT	REPEAT	830
1xkuA00.blast	REPEAT	REPEAT	136
1og2A00.blast	NON-REPEAT	REPEAT	82
1rmg000.blast	REPEAT	REPEAT	1025
1qcxA00.blast	REPEAT	NON-REPEAT	4
1v1mA00.blast	NON-REPEAT	REPEAT	640
1q22A00.blast	NON-REPEAT	REPEAT	492
1sat001.blast	REPEAT	NON-REPEAT	14
1s70B01.blast	REPEAT	NON-REPEAT	39
1ap7000.blast	REPEAT	REPEAT	413
1bhe000.blast	REPEAT	NON-REPEAT	64
1mt5C00.blast	NON-REPEAT	REPEAT	664
1pzlA00.blast	NON-REPEAT	NON-REPEAT	44
1qqeA00.blast	REPEAT	NON-REPEAT	44
1oflA00.blast	REPEAT	NON-REPEAT	5
1tndB01.blast	NON-REPEAT	REPEAT	378
1lrv000.blast	REPEAT	NON-REPEAT	0
1jz0C05.blast	NON-REPEAT	REPEAT	412
1io0A00.blast	REPEAT	REPEAT	106
1lshA02.blast	REPEAT	REPEAT	67
1qsaA01.blast	REPEAT	NON-REPEAT	48
1czfA00.blast	REPEAT	REPEAT	82
1oznA00.blast	REPEAT	REPEAT	104

Protein File	Actual Status	Prediction	SCORE
1gpmB02.blast	NON-REPEAT	REPEAT	1751
1tc0B00.blast	NON-REPEAT	NON-REPEAT	5
1hfvA00.blast	NON-REPEAT	NON-REPEAT	22
1ei5A01.blast	NON-REPEAT	NON-REPEAT	12
1wa5B00.blast	REPEAT	REPEAT	164
2behI01.blast	NON-REPEAT	REPEAT	209
1sdmA00.blast	NON-REPEAT	NON-REPEAT	25
1g5cE00.blast	NON-REPEAT	NON-REPEAT	9
2ie4A00.blast	REPEAT	NON-REPEAT	2
1rwbA00.blast	NON-REPEAT	NON-REPEAT	17
1kohD00.blast	REPEAT	REPEAT	233
1m56G00.blast	NON-REPEAT	NON-REPEAT	38
1a74B00.blast	NON-REPEAT	NON-REPEAT	10
1dabA00.blast	REPEAT	REPEAT	482
1ky5D01.blast	NON-REPEAT	NON-REPEAT	30
155l000.blast	NON-REPEAT	NON-REPEAT	17
2c0iA04.blast	NON-REPEAT	NON-REPEAT	0
1rlmD01.blast	NON-REPEAT	NON-REPEAT	29
1t7pA03.blast	NON-REPEAT	REPEAT	87
2bf7A00.blast	NON-REPEAT	REPEAT	106
1rvxG02.blast	NON-REPEAT	NON-REPEAT	5
1jm6A02.blast	NON-REPEAT	REPEAT	98
1hz4A00.blast	REPEAT	REPEAT	112
2bp7F01.blast	NON-REPEAT	NON-REPEAT	52
1w36E05.blast	NON-REPEAT	NON-REPEAT	57
1ru4A00.blast	REPEAT	REPEAT	482
1pgvA00.blast	REPEAT	REPEAT	106
1t34A02.blast	NON-REPEAT	NON-REPEAT	3
1qnyA00.blast	NON-REPEAT	NON-REPEAT	53
2ao6A00.blast	NON-REPEAT	NON-REPEAT	31
1tys000.blast	NON-REPEAT	NON-REPEAT	27
1cx8F01.blast	NON-REPEAT	NON-REPEAT	44
1m6bA01.blast	REPEAT	REPEAT	83
1czzA00.blast	NON-REPEAT	NON-REPEAT	6
1dcnC02.blast	NON-REPEAT	NON-REPEAT	32
1hyoA02.blast	NON-REPEAT	NON-REPEAT	29
1oelB03.blast	NON-REPEAT	NON-REPEAT	5
1g57B00.blast	NON-REPEAT	NON-REPEAT	2
1cal000.blast	NON-REPEAT	REPEAT	82
1qyaA02.blast	NON-REPEAT	NON-REPEAT	53
1rypE00.blast	NON-REPEAT	NON-REPEAT	4
1igrA01.blast	REPEAT	REPEAT	127
1q80A00.blast	NON-REPEAT	NON-REPEAT	31
1ms1B02.blast	NON-REPEAT	NON-REPEAT	10

Protein File	Actual Status	Prediction	SCORE
1vyhE00.blast	NON-REPEAT	NON-REPEAT	2
2c4gA02.blast	NON-REPEAT	NON-REPEAT	6
1po5A00.blast	NON-REPEAT	NON-REPEAT	23
1gt7Q00.blast	NON-REPEAT	REPEAT	85
2fakL00.blast	NON-REPEAT	NON-REPEAT	31
1oyzA00.blast	REPEAT	REPEAT	169
1qbqA00.blast	REPEAT	REPEAT	117
2iegA02.blast	NON-REPEAT	NON-REPEAT	32
1njfD01.blast	NON-REPEAT	NON-REPEAT	40
1v4vB01.blast	NON-REPEAT	NON-REPEAT	41
1tgvA00.blast	NON-REPEAT	NON-REPEAT	22
1bjoB01.blast	NON-REPEAT	REPEAT	79
1n5nB00.blast	NON-REPEAT	NON-REPEAT	18
1k5gD00.blast	NON-REPEAT	NON-REPEAT	7
1v3wA00.blast	REPEAT	NON-REPEAT	40
1oxmA00.blast	NON-REPEAT	NON-REPEAT	4
1ofgA02.blast	NON-REPEAT	REPEAT	102
1xhdA00.blast	REPEAT	NON-REPEAT	29
2b6oA00.blast	NON-REPEAT	NON-REPEAT	23
1fs2A00.blast	REPEAT	REPEAT	509
1vaoA03.blast	NON-REPEAT	NON-REPEAT	11
2ay7B02.blast	NON-REPEAT	NON-REPEAT	25
1chwB01.blast	NON-REPEAT	NON-REPEAT	4
2hpzA00.blast	NON-REPEAT	NON-REPEAT	6
1ogoX02.blast	REPEAT	REPEAT	162
1g1lF00.blast	NON-REPEAT	NON-REPEAT	28
1ofeB04.blast	REPEAT	NON-REPEAT	1
1mb2A01.blast	NON-REPEAT	NON-REPEAT	12
2agvB04.blast	NON-REPEAT	NON-REPEAT	13
1xurA00.blast	NON-REPEAT	NON-REPEAT	13
2bamA00.blast	NON-REPEAT	NON-REPEAT	0
1sqxB02.blast	NON-REPEAT	NON-REPEAT	37
1k5cA00.blast	REPEAT	REPEAT	92
1koqB00.blast	NON-REPEAT	NON-REPEAT	37
1scwA00.blast	NON-REPEAT	REPEAT	169
1s4eF02.blast	NON-REPEAT	NON-REPEAT	1
1hu3A00.blast	REPEAT	NON-REPEAT	17
1xu9A00.blast	NON-REPEAT	NON-REPEAT	9
1eu1A02.blast	NON-REPEAT	NON-REPEAT	35
1k94B00.blast	NON-REPEAT	NON-REPEAT	2
1nptO01.blast	NON-REPEAT	NON-REPEAT	53
1mkqA01.blast	NON-REPEAT	NON-REPEAT	7
1r4bA00.blast	NON-REPEAT	NON-REPEAT	33
2cizA00.blast	NON-REPEAT	NON-REPEAT	48

Protein File	Actual Status	Prediction	SCORE
1sqvA01.blast	NON-REPEAT	NON-REPEAT	4
1ykpK00.blast	NON-REPEAT	NON-REPEAT	55
1h0mB01.blast	NON-REPEAT	NON-REPEAT	40
1aq6B01.blast	NON-REPEAT	NON-REPEAT	27
1n5lB00.blast	NON-REPEAT	NON-REPEAT	2
1qjy200.blast	NON-REPEAT	NON-REPEAT	24
1r44D00.blast	NON-REPEAT	NON-REPEAT	52
1c7nD01.blast	NON-REPEAT	NON-REPEAT	23
1mczK02.blast	NON-REPEAT	NON-REPEAT	36
1y7hD00.blast	NON-REPEAT	NON-REPEAT	15
1l9mA02.blast	NON-REPEAT	NON-REPEAT	30
1ff3B00.blast	NON-REPEAT	NON-REPEAT	6
1sbzC00.blast	NON-REPEAT	NON-REPEAT	20
1m9lA00.blast	REPEAT	REPEAT	272
1w9cA00.blast	REPEAT	REPEAT	101
1p9hA00.blast	REPEAT	REPEAT	112
1tvbD01.blast	NON-REPEAT	NON-REPEAT	9
2c42A01.blast	NON-REPEAT	NON-REPEAT	51
2c1dG01.blast	NON-REPEAT	NON-REPEAT	47
1n2cG00.blast	NON-REPEAT	REPEAT	75
1ot3G00.blast	NON-REPEAT	NON-REPEAT	13
1q5pA00.blast	NON-REPEAT	REPEAT	73
3eugA00.blast	NON-REPEAT	NON-REPEAT	24
1g8yE00.blast	NON-REPEAT	NON-REPEAT	12
1xat000.blast	REPEAT	NON-REPEAT	18
1g2oC00.blast	NON-REPEAT	NON-REPEAT	7
1hghF00.blast	NON-REPEAT	NON-REPEAT	14
1isqA00.blast	NON-REPEAT	REPEAT	99
1h38D05.blast	NON-REPEAT	NON-REPEAT	29
1jgcB00.blast	NON-REPEAT	REPEAT	112
1hlpB01.blast	NON-REPEAT	NON-REPEAT	13
1ee6A00.blast	REPEAT	REPEAT	188
1m6bA03.blast	REPEAT	NON-REPEAT	8
2bs2C00.blast	NON-REPEAT	NON-REPEAT	1
1i43J01.blast	NON-REPEAT	NON-REPEAT	19
1xx7B00.blast	NON-REPEAT	NON-REPEAT	7
1a88B00.blast	NON-REPEAT	NON-REPEAT	45
2hr7A03.blast	REPEAT	REPEAT	115
1jdbH02.blast	NON-REPEAT	NON-REPEAT	11
1qmiC01.blast	NON-REPEAT	REPEAT	94
1nhcA00.blast	REPEAT	REPEAT	281
1mr7C00.blast	REPEAT	NON-REPEAT	14
1dvmC01.blast	NON-REPEAT	NON-REPEAT	9

Test set, 55 Solenoidi, 128 Non-Solenoidi

Protein File	Actual Status	Prediction	SCORE
1ialA00.blast	REPEAT	REPEAT	294
1j9rC02.blast	NON-REPEAT	NON-REPEAT	32
1r49A04.blast	NON-REPEAT	NON-REPEAT	7
1qsgH00.blast	NON-REPEAT	REPEAT	70
1g63F00.blast	NON-REPEAT	NON-REPEAT	18
1gw5B00.blast	REPEAT	NON-REPEAT	39
1ho8A01.blast	REPEAT	NON-REPEAT	7
1qgrA00.blast	REPEAT	NON-REPEAT	66
1qbkB00.blast	REPEAT	REPEAT	68
1oukA02.blast	NON-REPEAT	NON-REPEAT	11
1wnzA00.blast	NON-REPEAT	NON-REPEAT	61
1odsB00.blast	NON-REPEAT	REPEAT	69
1e2wA01.blast	NON-REPEAT	NON-REPEAT	11
2nsiC01.blast	NON-REPEAT	NON-REPEAT	34
1i7wA00.blast	REPEAT	REPEAT	244
1q5qM00.blast	NON-REPEAT	NON-REPEAT	12
1tqwA04.blast	NON-REPEAT	REPEAT	225
1rrlA01.blast	NON-REPEAT	NON-REPEAT	19
9pap000.blast	NON-REPEAT	NON-REPEAT	18
1u6gC00.blast	REPEAT	REPEAT	456
1iw8F00.blast	NON-REPEAT	NON-REPEAT	14
1itkB03.blast	NON-REPEAT	NON-REPEAT	24
1nx2A00.blast	NON-REPEAT	REPEAT	70
1ixfA00.blast	NON-REPEAT	NON-REPEAT	17
1wm5A00.blast	REPEAT	NON-REPEAT	55
1xnfA00.blast	REPEAT	REPEAT	149
1ouvA00.blast	REPEAT	REPEAT	67
1fchA00.blast	REPEAT	REPEAT	153
1gqcB00.blast	NON-REPEAT	REPEAT	79
2bbwA00.blast	NON-REPEAT	NON-REPEAT	3
1gadO01.blast	NON-REPEAT	REPEAT	68
1b89A00.blast	REPEAT	NON-REPEAT	21
2f8xK00.blast	REPEAT	REPEAT	265
1vqpM00.blast	NON-REPEAT	NON-REPEAT	33
1zy8H01.blast	NON-REPEAT	NON-REPEAT	18
1vezA00.blast	NON-REPEAT	NON-REPEAT	26
1nlvA01.blast	NON-REPEAT	NON-REPEAT	35
1gpl001.blast	NON-REPEAT	REPEAT	92
1qbgB00.blast	NON-REPEAT	NON-REPEAT	2
1dbzA01.blast	NON-REPEAT	NON-REPEAT	16
1dovA00.blast	NON-REPEAT	NON-REPEAT	31
1b15A00.blast	NON-REPEAT	NON-REPEAT	18

Protein File	Actual Status	Prediction	SCORE
1uhkB00.blast	NON-REPEAT	NON-REPEAT	33
1ireA00.blast	NON-REPEAT	NON-REPEAT	1
1iad000.blast	NON-REPEAT	NON-REPEAT	51
1bu9A00.blast	REPEAT	REPEAT	158
2fg7X01.blast	NON-REPEAT	NON-REPEAT	6
1j31C00.blast	NON-REPEAT	NON-REPEAT	6
1k3zD00.blast	REPEAT	REPEAT	185
1o9eA00.blast	NON-REPEAT	NON-REPEAT	23
1l3sA01.blast	NON-REPEAT	NON-REPEAT	5
1xn4A00.blast	NON-REPEAT	NON-REPEAT	8
1yrqJ00.blast	NON-REPEAT	NON-REPEAT	46
2jagA00.blast	NON-REPEAT	NON-REPEAT	13
1fj4A01.blast	NON-REPEAT	NON-REPEAT	2
1e6yD03.blast	NON-REPEAT	NON-REPEAT	13
1hqvA00.blast	NON-REPEAT	NON-REPEAT	45
1qymA00.blast	REPEAT	REPEAT	203
1iknD00.blast	REPEAT	REPEAT	123
1ixvA00.blast	REPEAT	REPEAT	239
1j2eB02.blast	NON-REPEAT	NON-REPEAT	8
1dzwP00.blast	NON-REPEAT	NON-REPEAT	24
2d5jB00.blast	NON-REPEAT	NON-REPEAT	46
1nz7A00.blast	NON-REPEAT	NON-REPEAT	1
1sw6A00.blast	REPEAT	REPEAT	130
1wdyA00.blast	REPEAT	REPEAT	309
2a4zA03.blast	REPEAT	NON-REPEAT	49
1e2fA00.blast	NON-REPEAT	NON-REPEAT	3
1dceA01.blast	REPEAT	NON-REPEAT	51
1f0jB00.blast	NON-REPEAT	NON-REPEAT	4
1k1eD00.blast	NON-REPEAT	NON-REPEAT	41
1gk3A01.blast	NON-REPEAT	NON-REPEAT	33
1w27A01.blast	NON-REPEAT	NON-REPEAT	61
1otkB00.blast	NON-REPEAT	REPEAT	91
1p7mA00.blast	NON-REPEAT	NON-REPEAT	10
1dmwA00.blast	NON-REPEAT	NON-REPEAT	47
1ecgA01.blast	NON-REPEAT	NON-REPEAT	45
1n52A03.blast	REPEAT	NON-REPEAT	45
1h2vC02.blast	REPEAT	NON-REPEAT	58
1tw1A00.blast	NON-REPEAT	NON-REPEAT	4
1rvuA01.blast	NON-REPEAT	NON-REPEAT	8
1t34A01.blast	NON-REPEAT	NON-REPEAT	11
1umtA00.blast	NON-REPEAT	NON-REPEAT	12
2d3sC00.blast	NON-REPEAT	NON-REPEAT	48
1poiC01.blast	NON-REPEAT	REPEAT	96
1go8P01.blast	REPEAT	REPEAT	236

Protein File	Actual Status	Prediction	SCORE
1hyt002.blast	NON-REPEAT	NON-REPEAT	61
1ia7A00.blast	NON-REPEAT	REPEAT	227
1ngsA01.blast	NON-REPEAT	NON-REPEAT	18
1m6xB02.blast	NON-REPEAT	NON-REPEAT	64
1wuiS01.blast	NON-REPEAT	NON-REPEAT	6
1kapP01.blast	REPEAT	REPEAT	464
1hm9A02.blast	REPEAT	REPEAT	141
1hv9A02.blast	REPEAT	REPEAT	268
1ypwD03.blast	NON-REPEAT	NON-REPEAT	4
1pmaU00.blast	NON-REPEAT	NON-REPEAT	22
1pl6B01.blast	NON-REPEAT	NON-REPEAT	57
1dofB01.blast	NON-REPEAT	NON-REPEAT	26
1qrlA00.blast	REPEAT	NON-REPEAT	29
1xkzA00.blast	NON-REPEAT	NON-REPEAT	18
1fvfA01.blast	NON-REPEAT	REPEAT	70
1hqyA00.blast	NON-REPEAT	NON-REPEAT	8
1efcB01.blast	NON-REPEAT	NON-REPEAT	0
1ciy001.blast	NON-REPEAT	NON-REPEAT	12
1o01G01.blast	NON-REPEAT	NON-REPEAT	25
1uhoA00.blast	NON-REPEAT	NON-REPEAT	10
1esiA00.blast	NON-REPEAT	NON-REPEAT	1
2aq9A01.blast	REPEAT	REPEAT	103
1tdtA02.blast	REPEAT	NON-REPEAT	2
1auiA00.blast	NON-REPEAT	NON-REPEAT	22
1jykA00.blast	NON-REPEAT	REPEAT	166
1dubC01.blast	NON-REPEAT	NON-REPEAT	8
1hg8A00.blast	REPEAT	REPEAT	376
1ls4A00.blast	NON-REPEAT	REPEAT	69
1lycA02.blast	NON-REPEAT	NON-REPEAT	48
1tf1A00.blast	NON-REPEAT	NON-REPEAT	5
1oxoA02.blast	NON-REPEAT	NON-REPEAT	14
2f4dA00.blast	NON-REPEAT	NON-REPEAT	4
2fny200.blast	NON-REPEAT	NON-REPEAT	11
1mczF03.blast	NON-REPEAT	NON-REPEAT	29
1krrA00.blast	REPEAT	NON-REPEAT	11
1pe9A00.blast	REPEAT	REPEAT	324
1s1mB01.blast	NON-REPEAT	NON-REPEAT	7
1ea0A04.blast	REPEAT	NON-REPEAT	23
1xkkA02.blast	NON-REPEAT	NON-REPEAT	9
1nstA00.blast	NON-REPEAT	NON-REPEAT	0
1txoB00.blast	NON-REPEAT	NON-REPEAT	4
1gjhA00.blast	NON-REPEAT	NON-REPEAT	13
1h80B00.blast	REPEAT	REPEAT	278
1o88A00.blast	REPEAT	REPEAT	75

Protein File	Actual Status	Prediction	SCORE
1rwrA00.blast	REPEAT	REPEAT	262
1n7pA02.blast	NON-REPEAT	REPEAT	107
1rvdA00.blast	NON-REPEAT	NON-REPEAT	47
1rj2D01.blast	NON-REPEAT	NON-REPEAT	6
1ia5A00.blast	REPEAT	REPEAT	600
1bn8A00.blast	REPEAT	REPEAT	163
1gq8A00.blast	REPEAT	REPEAT	151
1xg2A00.blast	REPEAT	REPEAT	123
1g98A02.blast	NON-REPEAT	NON-REPEAT	57
1a6o002.blast	NON-REPEAT	NON-REPEAT	5
1a16002.blast	NON-REPEAT	REPEAT	95
1w8aA00.blast	REPEAT	REPEAT	461
1z7xW00.blast	REPEAT	REPEAT	568
1igrA03.blast	REPEAT	REPEAT	80
2hr7A01.blast	REPEAT	REPEAT	70
1kjzA01.blast	NON-REPEAT	NON-REPEAT	3
1dl2A00.blast	NON-REPEAT	NON-REPEAT	27
1qjvA00.blast	REPEAT	REPEAT	221
1tsp000.blast	REPEAT	REPEAT	223
1x3lA02.blast	NON-REPEAT	NON-REPEAT	16
1e9rG01.blast	NON-REPEAT	NON-REPEAT	24
1umzA00.blast	NON-REPEAT	NON-REPEAT	17
3crxA01.blast	NON-REPEAT	NON-REPEAT	12
1oqhA02.blast	NON-REPEAT	NON-REPEAT	13
1oo5B00.blast	NON-REPEAT	NON-REPEAT	3
2fakY00.blast	NON-REPEAT	NON-REPEAT	10
1uu1C02.blast	NON-REPEAT	NON-REPEAT	56
1ookG00.blast	REPEAT	REPEAT	387
1ogqA00.blast	REPEAT	REPEAT	1212
1h6tA01.blast	REPEAT	REPEAT	967
1gh0C00.blast	NON-REPEAT	NON-REPEAT	14
1m44B00.blast	NON-REPEAT	NON-REPEAT	62
1nvfB01.blast	NON-REPEAT	NON-REPEAT	5
1t0qB00.blast	NON-REPEAT	NON-REPEAT	66
1c1dA02.blast	NON-REPEAT	NON-REPEAT	15
3stdC00.blast	NON-REPEAT	NON-REPEAT	54
2h7sC00.blast	NON-REPEAT	NON-REPEAT	6
1yvmA00.blast	NON-REPEAT	REPEAT	75
1t8wB02.blast	NON-REPEAT	NON-REPEAT	37
1xeuA01.blast	REPEAT	REPEAT	695
1upcE01.blast	NON-REPEAT	NON-REPEAT	39
1tyfM00.blast	NON-REPEAT	NON-REPEAT	9
1wd3A02.blast	NON-REPEAT	NON-REPEAT	53
1w36E03.blast	NON-REPEAT	NON-REPEAT	43

Protein File	Actual Status	Prediction	SCORE
1m90E00.blast	NON-REPEAT	NON-REPEAT	11
1hqhC00.blast	NON-REPEAT	NON-REPEAT	36
1qafB03.blast	NON-REPEAT	NON-REPEAT	32
1nq5A02.blast	NON-REPEAT	NON-REPEAT	14
1jl5A00.blast	REPEAT	REPEAT	925
1a9nC00.blast	REPEAT	REPEAT	132
1aihA00.blast	NON-REPEAT	NON-REPEAT	11
1q84A00.blast	NON-REPEAT	NON-REPEAT	4
2ft3A00.blast	REPEAT	REPEAT	407

Risultati ripetizioni, 110 Solenoidi

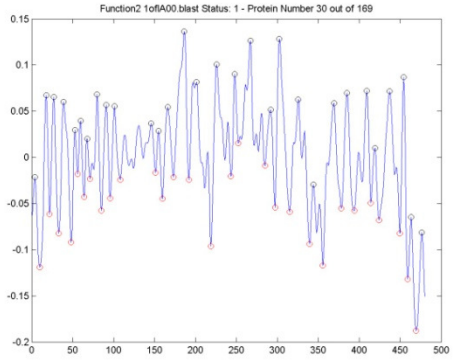
PROTEIN NAME	ACTUAL STATUS	PREDICTION	PROTEIN NAME	ACTUAL STATUS	PREDICTION
1i7wA00.blast	40	29	'1m8zA00.blast'	36	28
1ialA00.blast	40	31	1gw5A00.blast'	40	43
1gw5B00.blast	36	40	1p5qA02.blast'	34	15
1ho8A01.blast	43	30	'1vblA00.blast'	30	51
1qgrA00.blast	45	38	'2ca6A00.blast'	28	17
1qbkB00.blast	40	34	'1xkuA00.blast'	25	13
1u6gC00.blast	40	31	'1k1aA00.blast'	34	12
1wm5A00.blast	30	30	'1n11A00.blast'	34	11
1xnfA00.blast	33	16	'1j2zA01.blast'	18	10
1ouvA00.blast	35	22	'1rmg000.blast'	25	32
1fchA00.blast	34	13	'1qcxA00.blast'	22	45
1b89A00.blast	27	20	'1sat001.blast'	18	32
2f8xK00.blast	33	9	'1s70B01.blast'	32	22
1bu9A00.blast	32	14	'1ap7000.blast'	32	12
1k3zD00.blast	33	12	'1bhe000.blast'	25	27
1qymA00.blast	33	9	'1qqeA00.blast'	40	37
1iknD00.blast	36	12	'1oflA00.blast'	25	32
1ixvA00.blast	34	12	'1lrv000.blast'	25	13
1sw6A00.blast	42	16	'1io0A00.blast'	27	13
1wdyA00.blast	34	14	'1lshA02.blast'	40	21
2a4zA03.blast	34	14	'1qsaA01.blast'	30	20
1dceA01.blast	36	18	'1czfA00.blast'	25	18
1n52A03.blast	47	17	'1oznA00.blast'	24	12
1h2vC02.blast	41	25	'1m9lA00.blast'	23	13
1go8P01.blast	20	29	'1w9cA00.blast'	50	18
1kapP01.blast	18	33	'1p9hA00.blast'	15	15
1hm9A02.blast	17	17	'1fs2A00.blast'	26	23
1hv9A02.blast	17	29	'1oyzA00.blast'	38	18
1qrlA00.blast	17	14	'1qbqA00.blast'	34	31
2aq9A01.blast	18	10	'1m6bA01.blast'	25	13
1tdtA02.blast	18	26	'2ie4A00.blast'	40	35
1krrA00.blast	18	22	'1kohD00.blast'	25	17
1pe9A00.blast	30	23	'1dabA00.blast'	20	8
1hg8A00.blast	40	23	'1wa5B00.blast'	40	21
1ia5A00.blast	30	19	'1ru4A00.blast'	22	26
1bn8A00.blast	30	23	'1pgvA00.blast'	28	15
1gq8A00.blast	22	23	'1hu3A00.blast'	38	19
1xg2A00.blast	22	16	'1hz4A00.blast'	40	28
1h80B00.blast	30	15	'1k5cA00.blast'	30	27
1o88A00.blast	25	20	'1xat000.blast'	18	16
1rwrA00.blast	23	18	'1ogoX02.blast'	20	18
1qjvA00.blast	20	19	'2hr7A03.blast'	30	15

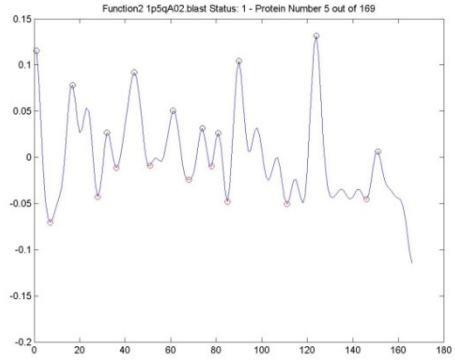
PROTEIN NAME	ACTUAL STATUS	PREDICTION	PROTEIN NAME	ACTUAL STATUS	PREDICTION
1tsp000.blast	30	27	'1ee6A00.blast'	22	22
1ea0A04.blast	20	25	'1m6bA03.blast'	25	22
1ookG00.blast	24	14	'1ofeB04.blast'	18	18
1ogqA00.blast	24	14	'1xhdA00.blast'	22	11
1h6tA01.blast	22	12	'1v3wA00.blast'	20	11
1xeuA01.blast	22	12	'1igrA01.blast'	30	20
2ft3A00.blast	24	13	'1nhcA00.blast'	25	18
1jl5A00.blast	22	13	'1mr7C00.blast'	18	23
1a9nC00.blast	22	14			
1w8aA00.blast	24	11			
1z7xW00.blast	30	17			
1igrA03.blast	30	16			
2hr7A01.blast	25	20			

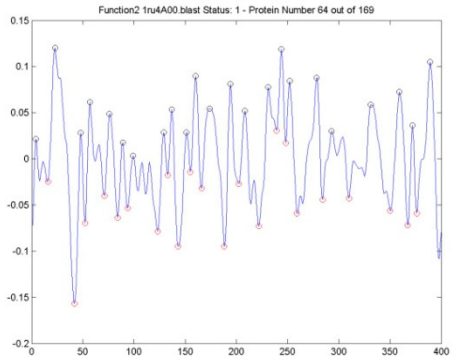
APPENDICE B

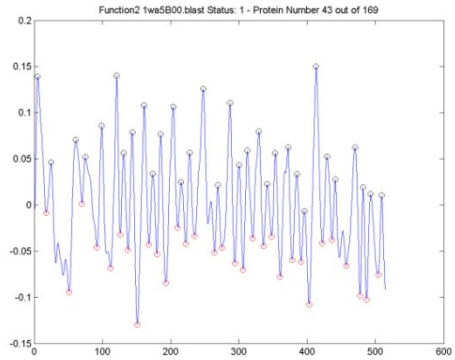
Proteine dal comportamento anomalo

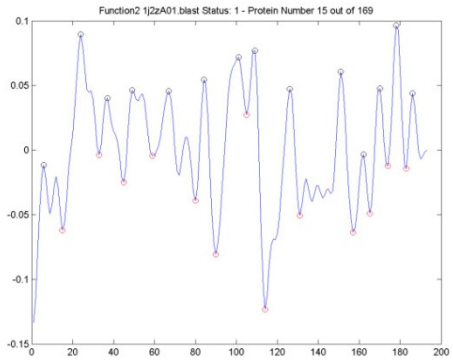
Proteine solenoidi con comportamento anomalo

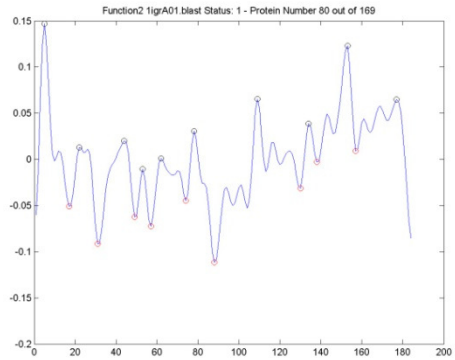


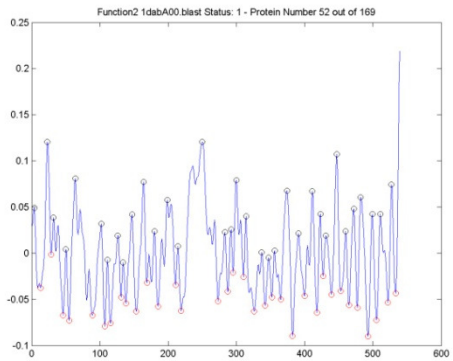


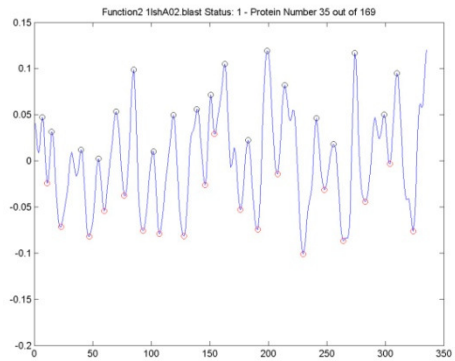


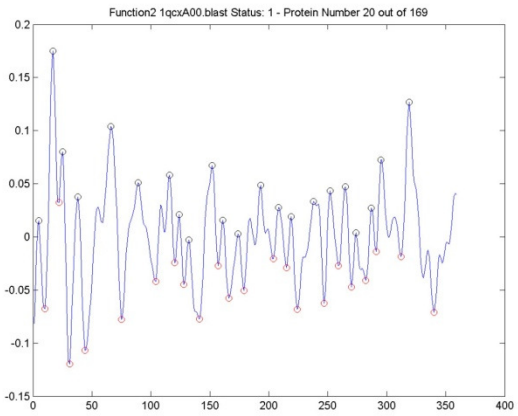


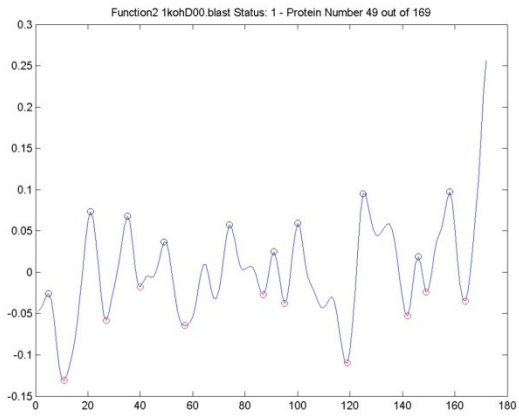


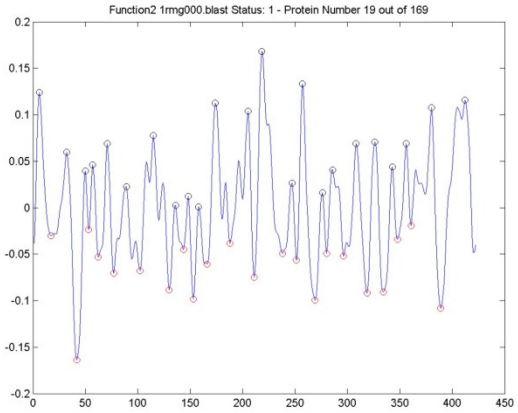


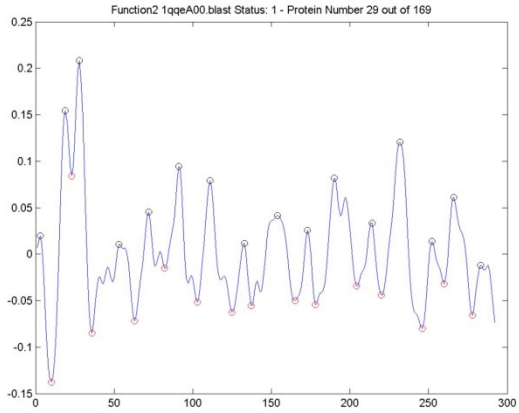


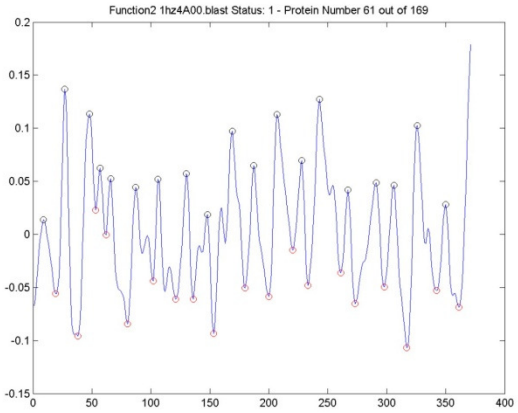


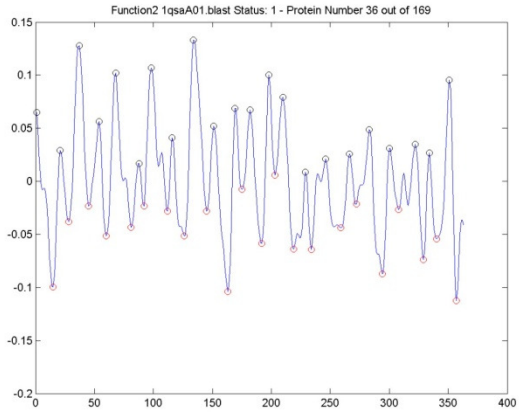

Function2 1qcxA00.blast Status: 1 - Protein Number 20 out of 169

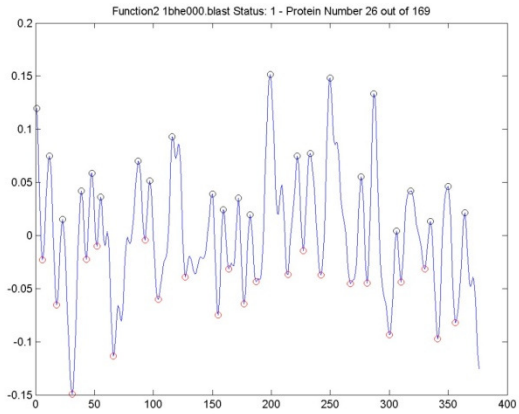

Function2 1kohD00.blast Status: 1 - Protein Number 49 out of 169


Function2 1rmg000.blast Status: 1 - Protein Number 19 out of 169

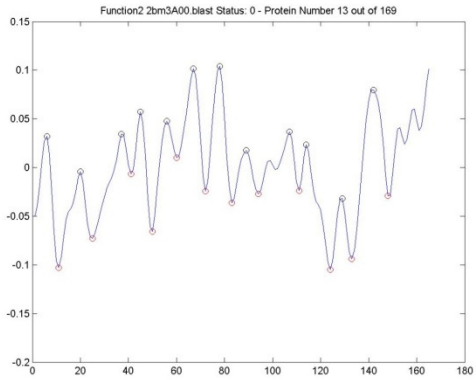

Function2 1qqeA00.blast Status: 1 - Protein Number 29 out of 169

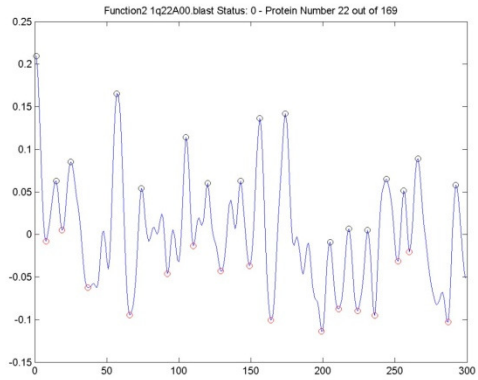

Function2 1hz4A00.blast Status: 1 - Protein Number 61 out of 169

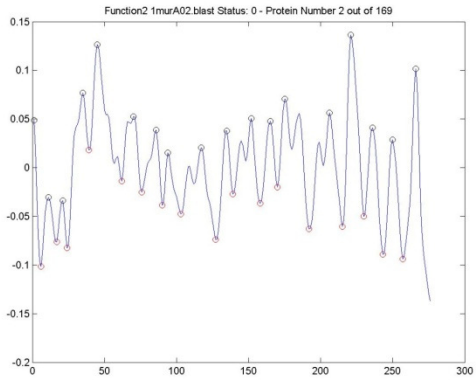

Function2 1qsaA01.blast Status: 1 - Protein Number 36 out of 169

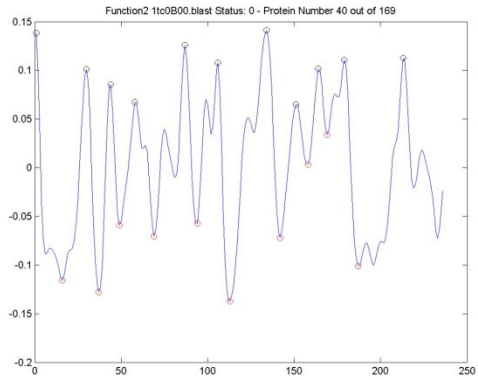

Function2 1bhe000.blast Status: 1 - Protein Number 26 out of 169

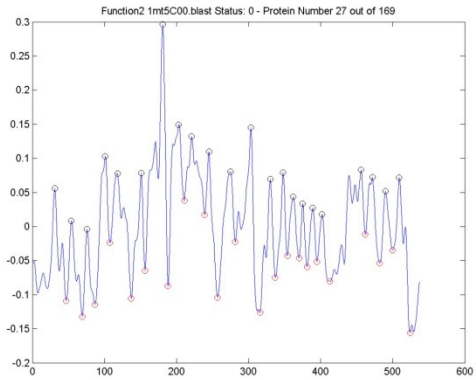
Proteine non-solenoidi con comportamento anomalo

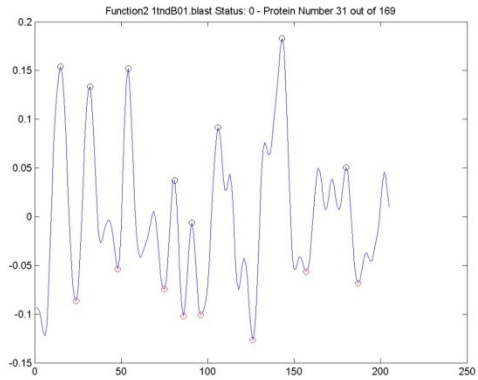


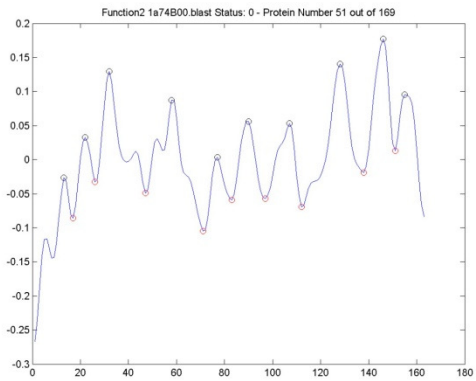


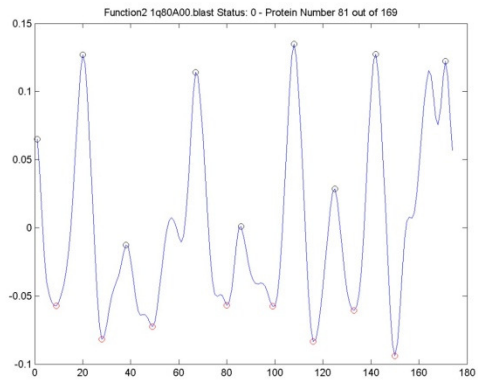


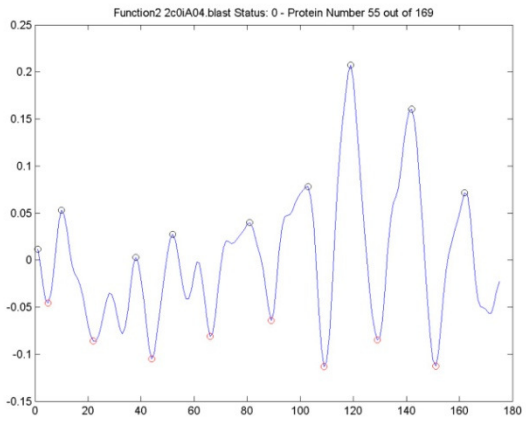


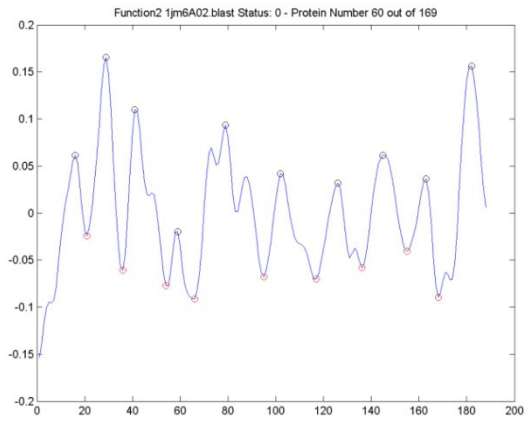


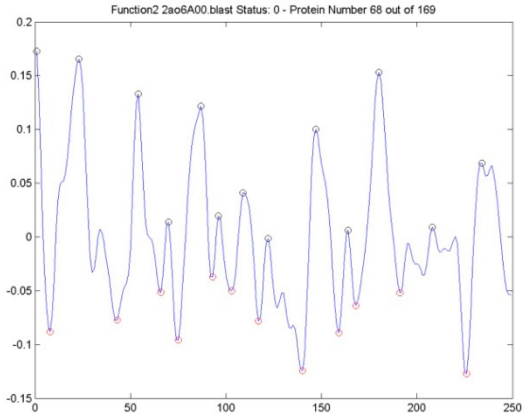


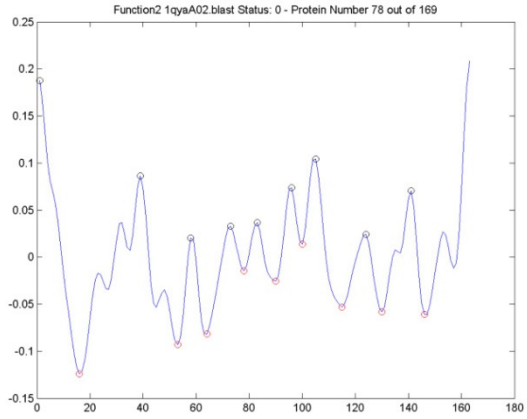


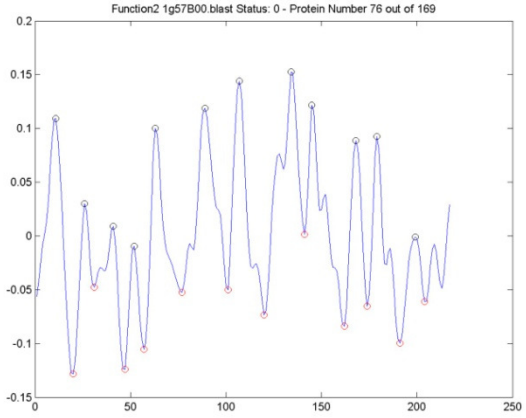


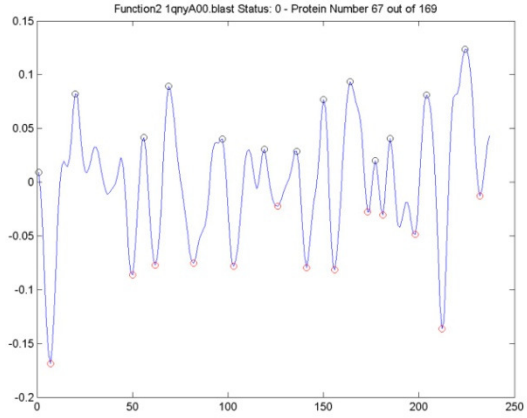

Function2 2c0iA04.blast Status: 0 - Protein Number 55 out of 169


Function2 1jm6A02.blast Status: 0 - Protein Number 60 out of 169


Function2 2ao6A00.blast Status: 0 - Protein Number 68 out of 169


Function2 1qyaA02.blast Status: 0 - Protein Number 78 out of 169


Function2 1g57B00.blast Status: 0 - Protein Number 76 out of 169


Function2 1qnyA00.blast Status: 0 - Protein Number 67 out of 169

Bibliografia

Altschul SF et al,. *Gapped BLAST and PSI-BLAST: a new generation of protein database search programs*. Nucleic Acids Res. 1997

Altschul S. F., Gish W., Miller W., Myers E. W., Lipman D. J.. *Basic Local Alignment Search Tool*. J. Mol. Biol., 1990

Altschul S. F., Madden T. L., Schäffer A. A., Zhang J., Zhang Z., Miller W., Lipman D. J. Gapped *BLAST and PSI-BLAST: a new generation of protein database search programs*. Nucl Acid Res,1997;

Andrade MA et al,. *Homology-based method for identification of protein repeats using statistical significance estimates*. J. Mol. Biol. 2000

Atchley WR et al, *Solving the protein sequence metric problem*. Proc. Natl Acad. Sci. USA, 2005

Bairoch A , *PROSITE: a dictionary of sites and patterns in proteins*. Nucleic Acids Res, 1991

Biegert A et al, *De novo identification of highly diverged protein repeats by probabilistic consistency. Bioinformatics*, 2008

Heger A, Holm L, *Rapid automatic detection and alignment of repeats in protein sequences.* Proteins 2000

Holm L., Ouzounis C., Sander C., Tuparev G., Vriend G. *A database of protein structure families with common folding motifs.* Protein Science 1, 1992

Holm L., Sander C. *Mapping the protein universe.* Science, 1996

Kajander T et al, *A new folding paradigm for repeat proteins*. J. Am. Chem. Soc. 2005

Kajava AV, *Review: proteins with repeated sequence–structural prediction and modeling.* J. Struct. Biol. 2001

Kajava AV et al, *New HEAT-like repeat motifs in proteins regulating proteasome structure and function*. J. Struct. Biol. 2004

Kajava AV et al, *Beta-structures in fibrous proteins.* Adv. Protein Chem. 2006

Kobe B, Kajava AV *When protein folding is simplified to protein coiling: the continuum of solenoid protein structures.* Trends Biochem. Sci. 2000

Li P, *Wavelets in bioinformatics and computational biology: state of art and perspectives*. Bioinformatics 2003

Main ER et al, *A recurring theme in protein engineering: the design, stability and folding of repeat protein.* Curr. Opin. Struct. Biol. 2005

Marcotte EM et al, *Census of protein repeats*. J. Mol. Biol. 1999

Marsella L et al, *REPETITA: detection and discrimination of the periodicity of protein solenoid repeats by Fourier transform*. Bioinformatics 2009

Murray KB et al, *Toward the detection and validation of repeats in protein structure*. Proteins, 2004

Pietrokovski S., *Searching databases of conserved sequence regions by aligning protein multiple-alignments*. Nucleic Acids Res., 1996

Sirocco F, Tosatto SCE, *TESE: generating specific protein structure test set ensembles*. Bioinformatics 2008

Smith T. F., Waterman M. S. *Identification of common molecular subsequences*. J. Mol. Biol., 1981

Sonnhammer EL, Eddy SR, Birney E, Bateman A, Durbin R (1998) *Pfam: multiple sequence alignments and HMM-profiles of protein domains*. Nucleic Acids Res, 1998

Tosatto SC, *The Victor/FRST Function for Model Quality Estimation*. J. Comput. Biol. 2005

Tosatto S.C.E., Albiero A., Mantovan A., Ferrari C., Bindewald E., Toppo S. *Align: a C++ Class Library and Web Server for Rapid Sequence Alignment Prototyping*. Bentham Science Publishers Ltd, 2006

Van Belkum, A., Scherer, S., Van Alphen, L. & Verbrugh, H. *Short-sequence DNA repeats in prokaryotic genomes*. *Microbiol Mol Biol*, 1998

Ringraziamenti

I miei ringraziamenti vanno al professor Carlo Ferrari e al professor Silvio Tosatto per la loro cordialità e per il supporto datomi durante tutto il periodo di lavoro nel gruppo di ricerca Biocomputing. Senza la loro pazienza e la loro guida non sarebbe stato possibile affrontare un così complesso argomento.

Ringrazio i ragazzi del laboratorio Biocomputing, in particolare il dott. Alberto J. Martin e il dott. Ian Walsh, per i loro consigli e per aver condiviso con me le loro conoscenze.