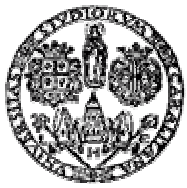


UNIVERSITÀ DEGLI STUDI DI CAGLIARI

DOTTORATO DI RICERCA IN INGEGNERIA

ELETTRONICA ED INFORMATICA

XVII CICLO

STUDIO E SPERIMENTAZIONE DI METODOLOGIE INNOVATIVE PER L'ANALISI DI SEQUENZE BIOLOGICHE

Relatore:

Giuliano Armano

Tesi di dottorato di:

Alessandro Orro

Coordinatore:

Massimo Vanzi

Novembre 2004

Premessa

L'argomento della presente tesi è lo studio di algoritmi innovativi per l'analisi di sequenze biologiche. In particolare l'attenzione è stata focalizzata su due importanti problemi di bioinformatica: la comparazione multipla di sequenze proteiche e la predizione di strutture secondarie proteiche.

Comparazione Multipla. Uno dei metodi di analisi proteica più diffusi in bioinformatica è quello comparativo. Proteine con sequenza simile hanno un'elevata probabilità di manifestare anche una struttura o funzione simile per cui una sequenza proteica può essere studiata tramite dei confronti con delle proteine a struttura/funzione nota.
L'allineamento multiplo è un metodo rigoroso che permette rappresentare le relazioni esistenti fra un'insieme di proteine mettendo in evidenza le operazioni di inserimento/cancellazione che hanno regolato l'evoluzione di una certa sequenza.
L'algoritmo proposto rientra in una particolare classe di algoritmi in cui la ricerca viene guidata da un confronto preliminare delle strutture secondarie. E' stata adoperata una tecnica di astrazione che ha come scopo quello di suddividere la ricerca in più livelli di dettaglio in modo che le soluzioni trovate ad alto livello siano una guida per quelle a livello più basso.
I risultati sperimentali hanno mostrato che l'algoritmo ha delle performance paragonabili allo stato dell'arte e in certi casi addirittura migliori (sequenze a bassa similarità).

Predizione di strutture secondarie. Un altro problema molto frequente in bioinformatica è quello della predizione delle strutture proteiche. E' noto infatti che la forma che una proteina assume nello spazio determina in gran parte quella che sarà la sua funzione all'interno della cellula. Il calcolo della struttura tridimensionale completa di una proteina è un problema intrattabile per cui la maggior parte dei metodi predittivi hanno come obiettivo la predizione della struttura secondaria che rappresenta il folding locale di proteina. Sebbene la struttura secondaria rappresenti una semplificazione del problema, può essere considerato come un punto di partenza per la predizione terziaria e in molti casi è sufficiente a determinare le caratteristiche funzionali della proteina.

I metodi attuali per la predizione di strutture secondarie proteiche si basano prevalentemente su tecniche di machine learning e in particolare su reti neurali.

Il lavoro proposto presenta un architettura innovativa chiamata *NXCS* in cui l'apprendimento viene affidato ad una popolazione di classificatori a visibilità locale anziché ad un singolo classificatore monolitico come avviene nei metodi in letteratura.

L'architettura presenta due livelli di apprendimento: il singolo classificatore formato da una rete neurale apprende e si specializza su un sottoinsieme degli input; il classificatore multiplo fa evolvere la popolazione dei classificatori con un algoritmo genetico in modo da premiare quelli che riescono a incrementare maggiormente le prestazioni della popolazione complessiva.

Anche in questo caso i risultati sperimentali hanno mostrato che l’architettura proposta presenta delle performance paragonabili con gli algoritmi attualmente disponibili.

Indice

Prima Parte

Seconda Parte

Capitolo 1
Introduzione

La Bioinformatica è una disciplina che si pone come scopo quello di affrontare i problemi della biologia e della medicina con metodologie tipiche dell'informatica. E' un ramo della scienza interdisciplinare che spazia dalla fisica alla matematica, dalla biologia all'informatica, dall'ingegneria alla medicina.

La Bioinformatica è nata innanzitutto dalla necessità di archiviare le grandi moli di dati che è in grado di produrre il mondo della ricerca in questi ultimi tempi (progetto Genoma) grazie anche alle recenti tecnologie come quella del DNA ricombinante. A questo scopo sono nate nel corso degli anni banche dati pubbliche gestite da organismi internazionali (NCBI, EMBL) che rendono disponibili ai ricercatori informazioni di vario tipo come sequenze (proteine/DNA), strutture, geni, pathway, etc. Il compito principale della Bioinformatica è dunque quello di trattare l'informazione biologica che si presenta spesso in forma di sequenza (DNA e proteine). Questo avviene in tre fasi principali:

- la rappresentazione dell'informazione, ovvero lo sviluppo di modelli matematici per descrivere le caratteristiche delle sequenze biologiche
- l'immagazzinamento dell'informazione, cioè la creazione e il mantenimento di database di informazioni biologiche che permettano la ricerca di sequenze e soprattutto la ricerca di relazioni fra i dati
- l'analisi delle informazioni; questo ramo, noto come Biologia Computazionale, è di fondamentale importanza per lo sviluppo della Bioinformatica e di tutte le tecnologie ad essa correlate (biomediche, agrarie, farmaceutiche)

La tesi proposta si occupa di quest'ultimo aspetto della bioinformatica, l'analisi delle sequenza biologiche, e farà riferimento in modo particolare alle sequenze proteiche. Il metodi sviluppati riguardano la predizione di strutture proteiche e i confronti fra la sequenze.

In Figura 1 sono mostrate schematicamente le relazioni che intercorrono tra la bioinformatica e le altre discipline. La bioinformatica riunisce in se diverse discipline fornendo degli strumenti utili in diverse aree di applicazione.

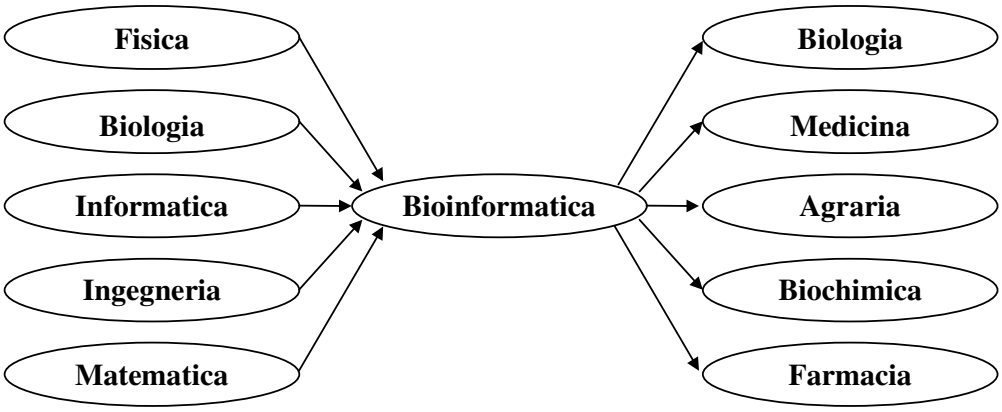


Figura 1: Relazioni fra la bioinformatica e le altre discipline

In questo primo capitolo verranno presentati alcuni richiami di biochimica, di bioinformatica funzionale e una breve panoramica sulle molteplici risorse bioinformatiche presenti sulla rete.

Nella prima parte della tesi saranno presentate le tecniche fondamentali di Biologia Computazionale con riferimento particolare agli allineamenti multipli. Verrà inoltre illustrato un nuovo metodo di allineamento basato sull'utilizzo di informazioni di struttura secondaria per migliorare la qualità dei risultati.

Nella seconda parte verrà presentato il problema della predizione di strutture proteiche e verrà proposta un'architettura generale per la predizione di strutture secondarie.

1.1 Richiami di Biochimica

La materia vivente pur essendo costituita da molecole che sono intrisecamente inanimate possiede molte caratteristiche peculiari non riscontrabili in altri tipi di materia, prima fra tutte la capacità di mantenere e perpetuare la vita.

La biochimica si pone come scopo fondamentale quello di studiare gli agglomerati di molecole inanimate e i processi chimici che sono alla base delle funzioni vitali degli organismi viventi.

Le biomolecole sono ordinate per dimensioni e complessità crescenti secondo la seguente gerarchia:

- precursori ambientali: molecole molto semplici derivate direttamente dall'ambiente esterno (anidride carbonica, acqua, ammoniaca, azoto);
- biomolecole di base: unità costitutive della materia vivente (nucleotidi, aminoacidi)
- macromolecole: biomolecole di base legate da legami covalenti in modo da formare molecole relativamente complesse a cui è già possibile attribuire una funzione biologica all'interno di una cellula (acidi nucleici, proteine, lipidi)
- organelli: strutture biologiche complesse che rappresentano gli elementi fondamentali della cellula (nucleo, mitocondri)

Nei paragrafi successivi faremo riferimento soprattutto ai livelli di organizzazione delle biomolecole di base e delle macromolecole, e in particolare le proteine e il DNA.

1.1.1 Aminoacidi e Proteine

Gli aminoacidi sono delle molecole molto importanti nella materia vivente in quanto sono le unità costitutive delle proteine. I venti aminoacidi standard (Figura 2) ovvero quelli che si trovano più comunemente nelle proteine presentano tutti una struttura simile (Figura 3) formata da:

- un gruppo HC, costituito da un atomo carbonio alfa legato con l'idrogeno,
- un gruppo α-aminico NH_3^+ legato all'atomo di carbonio alfa,
- un gruppo carbossilico COO^- legato all'atomo di carbonio alfa,
- un gruppo R caratteristico di ogni aminoacido.

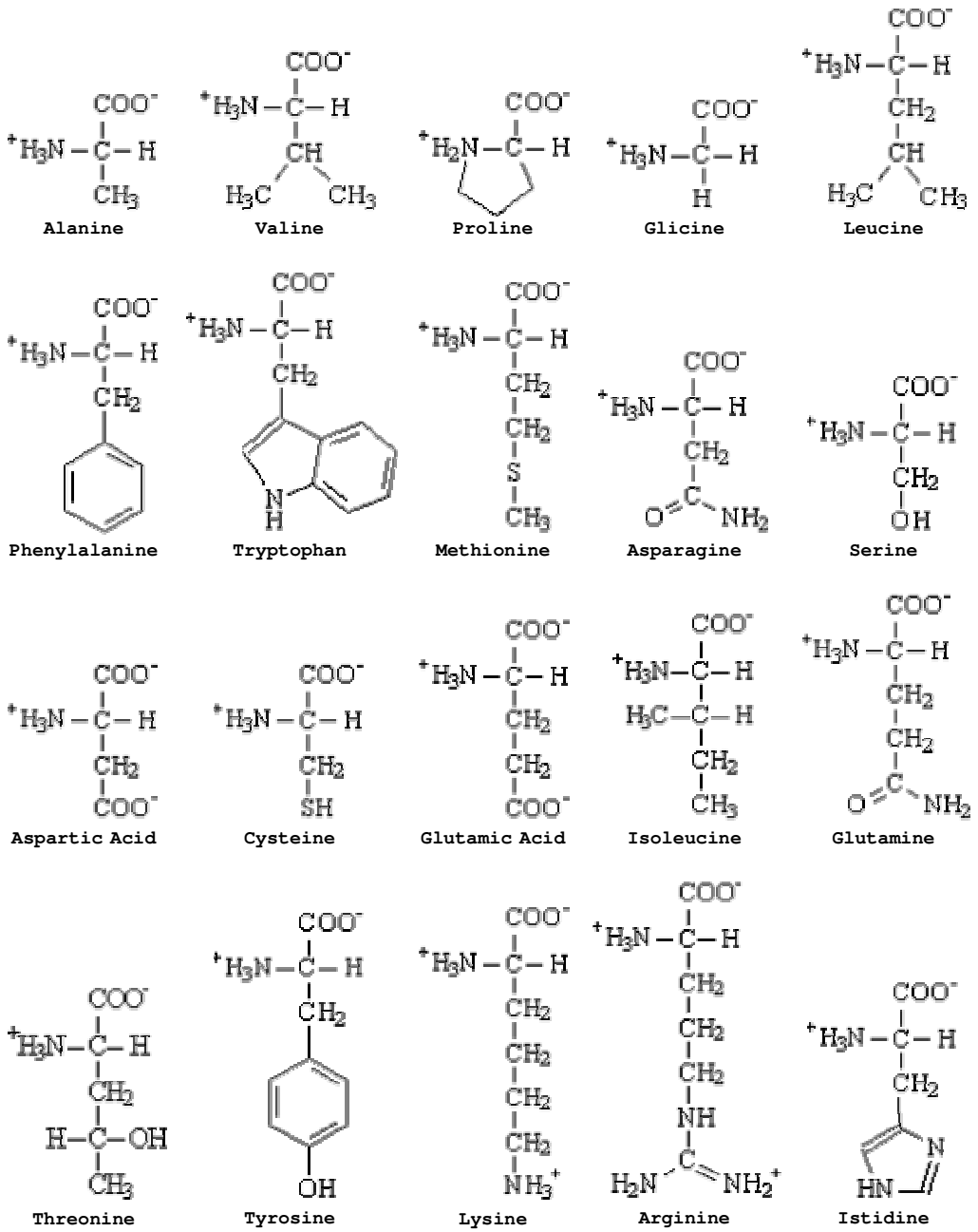


Figura 2: Struttura dei venti aminoacidi standard (forma ionizzata)

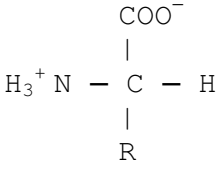


Figura 3: Struttura di un aminoacido

Fa eccezione la prolina in cui il gruppo R è legato sia all'atomo di carbonio che al gruppo aminico; quest'ultimo si presenta in forma differente dal caso standard: NH_2^+.

Le proteine, presenti in abbondanza in tutti gli organismi viventi, sono formate da una o più sequenze di aminoacidi dette catene polipeptidiche. Si tratta di macromolecole molto importanti in quanto sono responsabili di funzioni vitali delle cellule come l'attivazione e l'accelerazione di reazioni chimiche (attività enzimatica), il trasporto di particelle, il trasporto di impulsi nervosi, regolazione dei processi di sintesi. Inoltre le proteine sono le unità di base di muscoli e di strutture di supporto meccanico delle cellule.

Ogni proteina, nel suo stato nativo, presenta una forma tridimensionale caratteristica ben definita detta conformazione che è di fondamentale importanza per determinare la sua funzione biologica. A condizioni estreme di PH oppure ad alte temperature una proteina può subire un processo di denaturazione che la porta a perdere la sua struttura tridimensionale senza però rompere i legami covalenti della catena polipeptidica. Riportando successivamente la proteina al suo stato nativo vengono ripristinate la struttura tridimensionale e tutte le attività biologiche presenti nella proteina originale. Questo semplice esperimento dimostra che la sequenza degli amminoacidi contiene in sé tutta l'informazione necessaria per specificare la struttura tridimensionale nativa di una proteina e di conseguenza anche le sue funzionalità.

In realtà allo stato attuale la struttura tridimensionale può essere determinata con esattezza soltanto con esperimenti in laboratorio (diffrazione dei raggi X) il cui costo e complessità non consentirebbero ai ricercatori di seguire di pari passo la il numero di nuove sequenze che ogni giorno vengono scoperte.

La descrizione della struttura di una proteina può essere fatta, seppur in maniera semplicistica, utilizzando la gerarchia a quattro livelli di organizzazione proposta da Linderstrøm-Lang: struttura primaria, secondaria, terziaria e quaternaria. Ogni livello è caratterizzato da un particolare tipo di forze di legame fra aminoacidi e ogni livello di organizzazione è composto da elementi del livello precedente.

Per **struttura primaria** si intende la struttura chimica della catena polipeptidica e cioè il numero e la sequenza di aminoacidi legati fra loro dal legame peptidico.

Tale legame si forma per eliminazione di una molecola di acqua dal gruppo carbossilico di un amminoacido e dal gruppo α-aminico del successivo. Nella struttura primaria non sono presenti altri legami fra aminoacidi se non quello peptidico che, trattandosi di un legame covalente e perciò molto forte, garantisce una notevole stabilità alla struttura. In Figura 4 è mostrata una semplice catena polipeptidica in forma non ionizzata, formata da tre aminoacidi legati fra loro da legami peptidici. Notare che nella catena polipeptidica è univocamente definita una direzione che va dall'estremità N-terminale, corrispondente al gruppo aminico più esterno, all'estremità C-terminale, corrispondente al gruppo carbossilico più esterno.

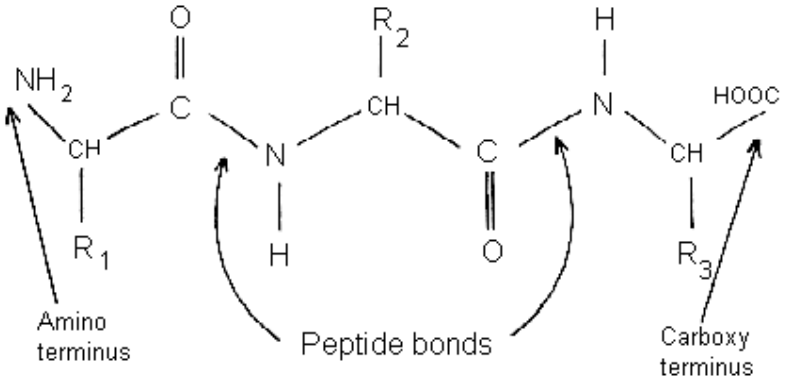


Figura 4: Esempio di catena polipeptidica

Sebbene la struttura primaria mette in evidenza soltanto la struttura unidimensionale della proteina, essa contiene tutta l'informazione necessaria per descrivere la sua conformazione tridimensionale che nasce dai ripiegamenti dovuti a legami che si formano fra i residui delle catene. Tali legami, generalmente non covalenti, conferiscono alla struttura tridimensionale una stabilità minore rispetto a quella primaria e di conseguenza anche una maggiore adattabilità strutturale in risposta ad eventuali interazioni con molecole esterne.

La struttura tridimensionale viene descritta nei livelli di struttura secondaria e terziaria. Per **struttura secondaria** si intende la disposizione regolare e ricorrente in una dimensione della catena polipeptidica. Tale struttura, evidente soprattutto nelle proteine fibrose, è dovuta ai legami a idrogeno che si formano fra il gruppo carbossilico (COO^-) di un residuo e il gruppo aminico (NH_3^+) di un altro residuo.

La classificazione delle strutture secondarie più nota è la seguente:

- Helix: Si tratta di una struttura ripetitiva in cui tutti i peptiti hanno la stessa relazione di legame con il successivo e con angoli tali da generare una configurazione a elica. Tali strutture sono caratterizzate dal numero (n) di residui per torsione e dalla salita (r) per residuo elicoidale. Per convenzione n>0 quando l'elica è orientata secondo la regola della mano destra. Fra le strutture secondarie è la più abbondante nelle proteine (32-38% di residui nelle proteine globulari); in paricolare la struttura a elica più comune è l'alfa-elica in cui il gruppo C=O del residuo i è legato tramite legame a idrogeno al gruppo HN del residuo i+4.
- Sheet: Si tratta di una struttura ripetitiva con conformazione estesa (foglio) formata da una sequenza parallela di beta strand uniti fra loro da legami a idrogeno. Il beta strand può essere visto come una struttura a zigzag (elica con n=2) che non presenta una configurazione stabile se presa singolarmente..
- Turns: Si tratta di una struttura (non ripetitiva) in cui un legame a idrogeno fra due residui causa un cambiamento di direzione della catena polipeptidica. A seconda degli angoli e dell'orientamento della volta si possono distinguere diversi tipi di strutture turns.
- Random Coil: Sono zone della proteina che non presentano nessuna particolare struttura secondaria.

Alcuni esempi di strutture secondarie sono mostrati in Figura 5.

Per **struttura terziaria** si intende l'organizzazione spaziale di una catena polipeptidica, ovvero il modo in cui la catena polipeptidica è ripiegata tridimensionalmente in modo da formare la struttura compatta e strettamente avvolta delle proteine globulari.

La presenza frequente di piccole successioni di elementi di struttura secondaria come per esempio alpha-beta-beta o beta-beta-alpha rende necessaria la definizione di un livello intermedio di organizzazione; si parla in tal caso di elementi di struttura-supersecondaria o motivi.

Un tipico legame che entra in gioco nella struttura terziaria è il ponte disolfuro (disulfide bonds) fra i gruppi R residui di cisteina. Notare che aminoacidi lontani nella struttura primaria possono essere vicini in 3D cioè nella struttura terziaria.

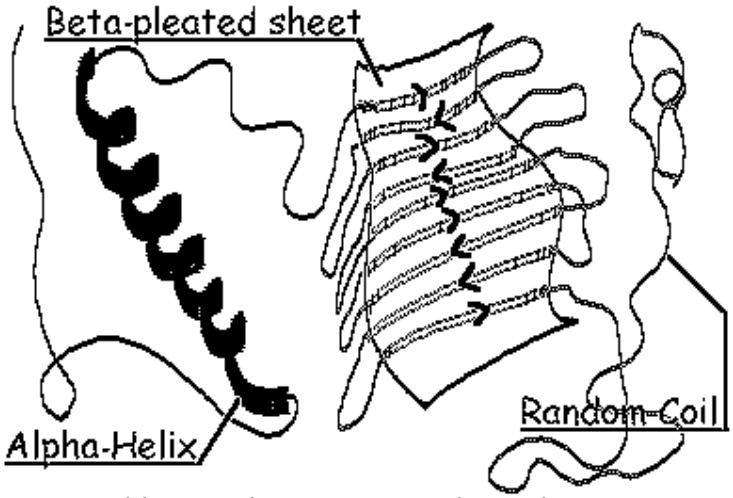


Figura 5: Strutture secondarie

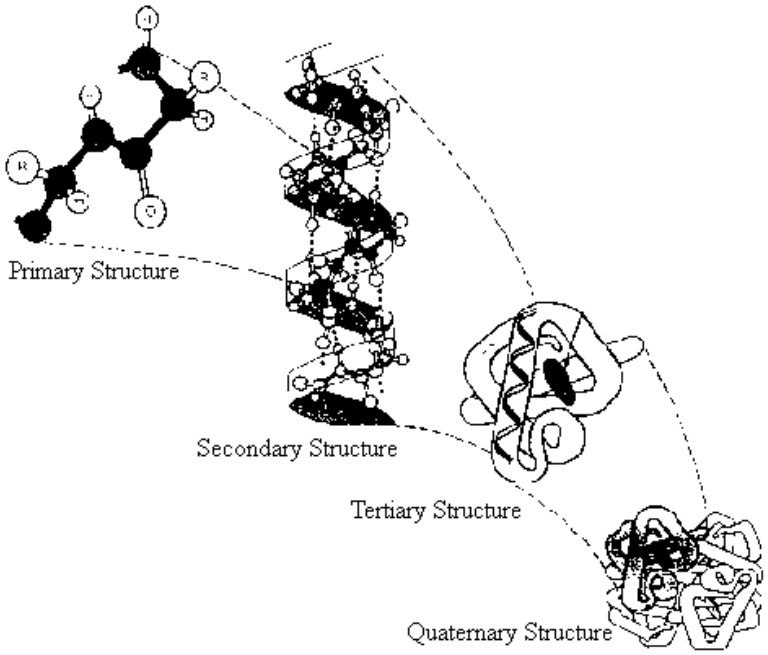


Figura 6: Livelli di organizzazione strutturale delle proteine

La **struttura quaternaria** infine è determinata dall'associazione di due o più unità polipeptidiche unite tra loro da legami deboli in modo da formare strutture complesse.

In Figura 6 sono mostrate le relazioni fra i differenti livelli di organizzazione strutturale della proteina. Notare come la struttura primaria rappresenta le sequenze dal punto di vista puntuale facendo riferimento cioè ai singoli aminoacidi, la struttura secondaria e la struttura terziaria rappresentano le relazioni esistenti fra aminoacidi rispettivamente vicini e lontani mentre la struttura quaternaria rappresenta l'intera proteina come insieme di catene polipeptidiche.

1.1.2 Nucleotidi e DNA

L'acido deossiribonucleico (DNA) e l'acido ribonucleico (RNA) sono le macromolecole responsabili del mantenimento e del trasferimento del patrimonio genetico nelle specie viventi. In particolare il DNA è formato da una catena di quattro tipi diversi di nucleotidi che sono l'adenina, la guanina, la citosina e la timina (Figura 7).

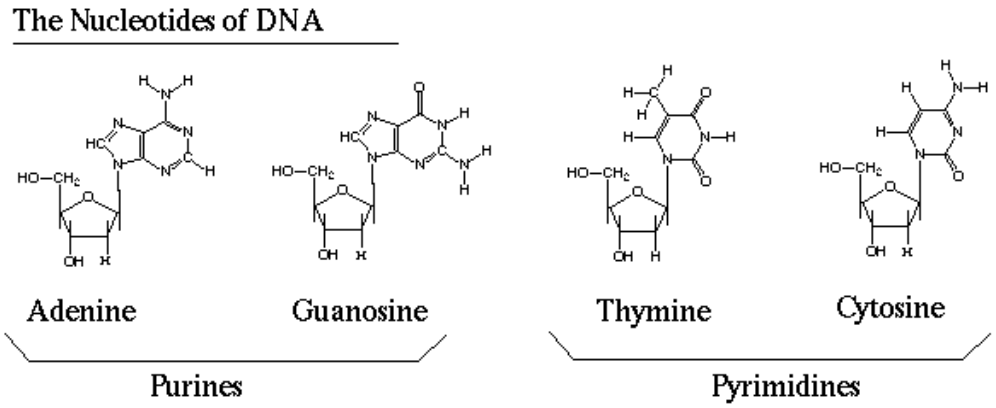


Figura 7: Nucleotidi

Secondo il modello proposto da Watson e Crick (1953) il DNA presenta una struttura a doppia elica in cui due catene antiparallele di nucleotidi sono avvolte a spirale attorno allo stesso asse. Le basi di un filamento formano legami a idrogeno con le corrispondenti basi dell'altro filamento in modo tale che soltanto due tipi di accoppiamento sono permessi: adenina-timina e guanina-citosina. Le due coppie

consentite sono tali da conferire alla struttura una stabilità maggiore in quanto formano un legame più forte e hanno circa la stessa dimensione.

Il DNA contiene tutte le informazioni necessarie per la sintesi delle proteine in quanto la sua sequenza è in grado di codificare con esattezza la sequenza polipeptidica. Un segmento di DNA che contiene un'intera catena polipeptidica è detto gene.

Il linguaggio di codifica si basa su una corrispondenza fra gli aminoacidi e le triplette di nucleotidi (Figura 8). Un aminoacido può essere rappresentato da più triplette che solitamente differiscono solo per l'ultimo nucleotide (il meno caratteristico), mentre una tripletta può rappresentare un solo aminoacido. Inoltre sono presenti dei simboli speciali che codificano la fine della sequenza.

A volte può accadere che un gene che specifica una certa proteina subisca una modifica chimica, causata per esempio da radioattività o particolari agenti chimici, che si ripercuote sulla proteina codificata. La proteina presenterà un'alterazione che si manifesta con la mutazione, l'inserimento o la cancellazione da una sequenza di uno o più aminoacidi e di conseguenza ci sarà anche una possibile alterazione della funzione.

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

Figura 8: Codifica degli aminoacidi per l'RNA

1.1.3 Omologia

Il problema della predizione della struttura tridimensionale di una proteina a partire dalla sequenza di aminoacidi è particolarmente difficile in quanto la catena polipeptidica ha un elevato numero di conformazioni potenziali e la struttura tridimensionale è scarsamente stabile, per cui la differenza in energia libera tra gli stati ripiegati e non ripiegati è piccola. Non è stato ancora formulata una teoria generale per correlare l'informazione unidimensionale con la complessa forma tridimensionale della proteina ripiegata.

In problemi di questo tipo sta assumendo un ruolo importante l'analisi comparativa fra sequenze di proteine omologhe. Per la definizione di proteine omologhe esistono diverse scuole di pensiero. Una definizione rigorosa si basa sul rapporto evolutivo: due proteine sono omologhe se derivano dallo stesso progenitore. Questa definizione però non è adeguata dal punto di vista pratico in quanto non è semplice stabilire questo tipo di relazione soprattutto se hanno una percentuale di identità bassa (<30%).

Una definizione più accettabile è quella basata sulla somiglianza strutturale: due proteine sono omologhe se hanno strutture e funzioni simili. Chiaramente va definita una metrica rigorosa che tenga conto dei vari gradi di similitudine per cui sono stati definiti i cosiddetti “modelli di punteggio” ovvero delle funzioni che restituiscono un punteggio di similarità per una data coppia (o insieme) di sequenze. Si tratta ovviamente di modelli approssimati che forniscono una stima dell’omologia.

1.2 Bioinformatica funzionale

La bioinformatica funzionale è un sottoramo della bioinformatica che si occupa dello sviluppo di algoritmi e modelli biologici orientati allo studio delle funzioni svolte dalle macromolecole più importanti che sono il DNA e le proteine. In generale l’analisi di queste macromolecole può essere fatto a diversi livelli di conoscenza: atomico e molecolare.

I modelli che lavorano a livello atomico solitamente sono molto complessi sia da un punto di vista teorico che computazionale e per questo possono essere applicati soltanto a delle molecole relativamente semplici. Purtroppo sia le proteine che le

catene di DNA sono molecole molto complesse per cui si può semplificare l'analisi se si considerano direttamente i blocchi elementari che le costituiscono ovvero i nucleotidi e gli aminoacidi rispettivamente. Per questo motivo le proteine e il DNA sono spesso rappresentate da un punto di vista informatico con delle stringhe su un alfabeto che rappresenta tutti i possibili aminoacidi o nucleotidi.

1.2.1 Analisi di sequenze biologiche

Nella sequenza di aminoacidi che compone una proteina è codificata la struttura tridimensionale e di conseguenza loro funzione che assumerà all'interno della cellula. Analogamente nella sequenza di nucleotidi che compongono il DNA di un gene è codificata la funzione del gene che si esprime attraverso la produzione di proteine controllata da complessi meccanismi di regolazione genica.

Nelle sequenze è perciò contenuta tutta l'informazione necessaria per capire il significato del messaggio codificato. Questa operazione di decodifica risulta però estremamente complessa a causa della rete di interazioni che avvengono fra le macromolecole. Per la maggior parte delle proteine e geni è impossibile effettuare uno studio approfondito dopo averli isolati in quanto non si terrebbero conto di queste interazioni. Nelle figure successive sono mostrati alcuni esempi di interazioni: in Figura 9 l'interazione fra due catene proteiche nella formazione di un legame cisterna, importante per la stabilizzazione delle strutture terziare; in Figura 10 un complesso pathway genetico.

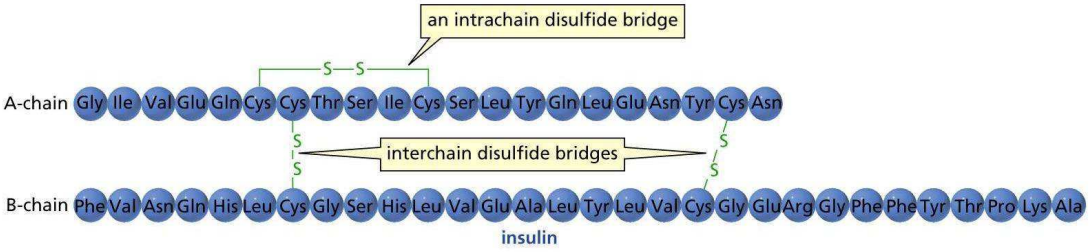


Figura 9: Interazione fra aminoacidi di catene proteiche

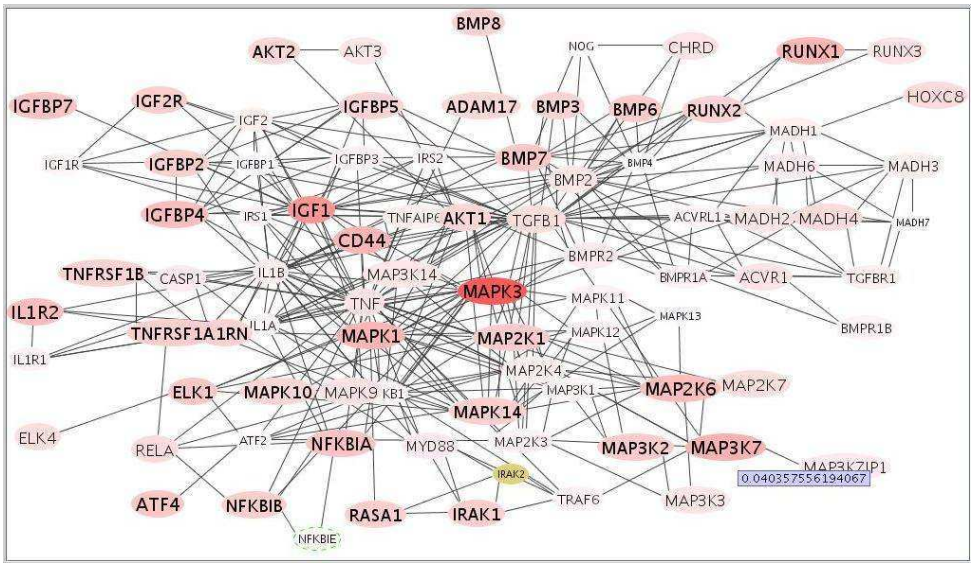


Figura 10: pathway di interazioni fra geni e proteine

Sebbene ogni singolo pathway possa essere studiato singolarmente sia da un punto di vista sperimentale che tramite modelli biologici, risulta molto difficile creare dei modelli di validità generale che tengano conto anche di agenti esterni. Questi tipi di studio sono indispensabili per capire il funzionamento dei metabolismi cellulari ma soprattutto per studiare l'effetto di farmaci e la loro relazioni con agenti esterni imprevisti.

Esistono vari tipo di algoritmi di analisi che hanno come target principale una sequenza. In particolare nei capitoli successivi verranno mostrati i metodi comparativi, basati sul fatto che una funzione proteica può essere inferita effettuando dei confronti con sequenze note, e metodi predittivi, che provano a costruire un modello struttura/funzione a partire dai dati campione con tecniche di machine learning.

1.2.2 Risorse Web

La possibilità di disporre liberamente di risorse distribuite in rete offre enormi vantaggi per tutte le attività di ricerca scientifica. Per la bioinformatica in particolare l'utilizzo di queste risorse è quasi obbligatorio soprattutto per quanto riguarda la necessità di accedere rapidamente alle informazioni contenute nelle banche dati. Infatti a causa dell'enorme quantità di dati il mantenimento di questi

database è molto costoso e può essere sostenuto soltanto da pochi centri specializzati. In Figura 11 è mostrato l'incremento in questi ultimi anni del numero di nucleotidi mantenuti da tre database fra i più noti:

- il database americano NCBI mantenuto dal National Center for Biotechnology Information
- il database europeo EMBL mantenuto dal European Bioinformatics Institute
- il database giapponese DDBJ mantenuto dal National Institute of Genetics

I tre database presentano un sistema di collegamento in modo da scambiarsi le informazioni e rimanere così aggiornati contemporaneamente. I grossi centri di ricerca come quelli sopra citati offrono inoltre innumerevoli servizi di calcolo biocomputazionale accessibili online come per esempio: tools di ricerca all'interno del database (BLAST), programmi di predizione di strutture (PHD), visualizzatori di strutture tridimensionale, calcolo e visualizzazione di allineamenti, ecc.

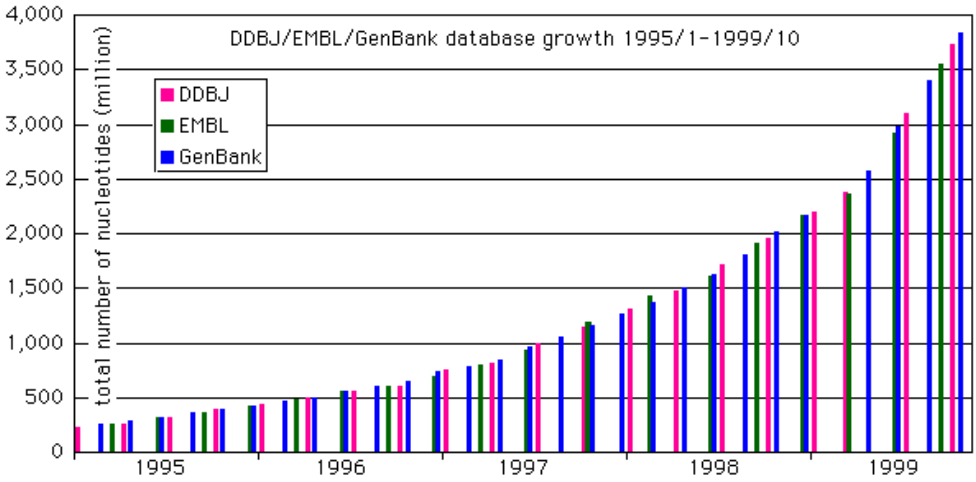


Figura 11: Incremento delle sequenze nucleotidiche

In particolare NCBI consente di effettuare ricerche di tipo bibliografico con il sistema PubMed che permette di accedere a 9 milioni di citazioni in MEDLINE.

In Figura 12 è mostrato il risultato di una ricerca (entry) nel database EMBL dove si può notare il sistema di riferimenti ad altri database come PROSITE (database di pattern di famiglie proteiche) o PDB (database di strutture); inoltre sono presenti i riferimenti ad altre sequenze correlate con quella in esame.

```
ID   NIFD_AZOVI     STANDARD;      PRT;   491 AA.
AC   P07328;
DT   01-APR-1988 (Rel. 07, Created)
DT   01-APR-1990 (Rel. 14, Last sequence update)
DT   30-MAY-2000 (Rel. 39, Last annotation update)
DE   NITROGENASE MOLYBDENUM-IRON PROTEIN ALPHA CHAIN (EC 1.18.6.1)
DE   (NITROGENASE COMPONENT I) (DINITROGENASE).
GN   NIFD.
OS   Azotobacter vinelandii.
OC   Bacteria; Proteobacteria; gamma subdivision; Pseudomonadaceae;
OC   Azotobacter.
OX   NCBI_TaxID=354;
RN   [1]
RP   SEQUENCE FROM N.A.
RX   MEDLINE=89123097; PubMed=2644218;
RA   Jacobson M.R., Brigle K.E., Bennett L.T., Setterquist R.A.,
RA   Wilson M.S., Cash V.L., Beynon J., Newton W.E., Dean D.R.;
RT   "Physical and genetic map of the major nif gene cluster from
RT   Azotobacter vinelandii.";
RL   J. Bacteriol. 171:1017-1027(1989).
...
DR   EMBL; M20568; AAA64710.1; -.
DR   EMBL; M11579; AAA22143.1; -.
DR   EMBL; X06886; CAA30004.1; -.
DR   PIR; B23969; NIAVMA.
DR   PIR; B13724; B13724.
DR   PIR; S02324; S02324.
DR   PDB; 2MIN; 01-APR-97.
DR   PDB; 3MIN; 01-APR-97.
DR   PDB; 1N2C; 12-NOV-97.
DR   InterPro; IPR000318; Nitrognse_comp1.
DR   InterPro; IPR000510; Oxidored_nitrognse_1.
DR   Pfam; PF00148; oxidored_nitro; 1.
DR   PROSITE; PS00090; NITROGENASE_1_2; 1.
DR   PROSITE; PS00699; NITROGENASE_1_1; 1.
KW   Oxidoreductase; Nitrogen fixation; Molybdenum; Iron-sulfur;
...
SQ   SEQUENCE   491 AA;  55158 MW;  6F02087DD889E52F CRC64;
     TGMSREEVES LIQEVLEVYP EKARKDRNKH LAVNDPAVTQ SKKCIISNKK SQPGLMTIRG
     CAYAGSKGVV WGPIKDMIHI SHGPVGCGQY SRAGRRNYYI GTTGVNAFVT MNFTSDFQEK
     DIVFGGDKKL AKLIDEVETL FPLNKGISVQ SECPIGLIGD DIESVSKVKG AELSKTIVPV
     RCEGFRGVSQ SLGHHIANDA VRDWVLGKRD EDTTFASTPY DVAIIGDYNI GGDAWSSRIL
     LEEMGLRCVA QWSGDGSISE IELTPKVKLN LVHCYRSMNY ISRHMEEKYG IPWMEYNFFG
     PTKTIESLRA IAAKFDESIQ KKCEEVIAKY KPEWEAVVAK YRPRLEGKRV MLYIGGLRPR
     HVIGAYEDLG MEVVGTGYEF AHNDDYDRTM KEMGDSTLLY DDVTGYEFEE FVKRIKPDLI
     GSGIKEKFIF QKMGIPFREM HSWDYSGPYH GFDGFAIFAR DMDMTLNNPC WKKLQAPWEA
     SEGAEKVAAS A
//
```

Figura 12: Entry di un database EMBL

Prima Parte

Allineamenti Multipli Proteici

Capitolo 2
Metodi Comparativi

Uno degli strumenti più usati in bioinformatica per l'analisi delle sequenze biologiche è il confronto con sequenze note. Come è stato già detto infatti, due sequenze che presentano una certa similarità probabilmente presentano anche caratteristiche simili dal punto di vista funzionale; inoltre una somiglianza di sequenze appartenenti a organismi differenti può essere interpretata in termini di relazione evolutiva fra le specie.

In questo capitolo, si analizzeranno alcuni dei metodi di confronto di sequenze proteiche facendo riferimento in particolare agli allineamenti a coppie e multipli. Nell'ultimo paragrafo verranno presi in esame alcuni modelli di rappresentazione e di confronto alternativi come i pattern e le catene markoviane.

2.1 Rappresentazione di sequenze biologiche

Il resto del capitolo si concentra sul problema della predizione di strutture secondarie proteiche.

Lavorando esclusivamente a livello di sequenza è possibile rappresentare la struttura primaria e quella secondaria con delle stringhe. Secondo la convenzione del codice IUPAC, gli aminoacidi e i nucleotidi possono essere codificati utilizzando delle lettere dell'alfabeto come mostrato in Figura 1 e Figura 2.

Una sequenza biologica $\boldsymbol{s}$ di lunghezza n sarà pertanto una stringa di n caratteri dove il generico elemento s_i appartiene ad un alfabeto Σ:

$$\mathbf{s} = s_1 s_2 s_3 ... s_n \qquad s_i \in \Sigma$$

dove nel caso di sequenze proteiche l'alfabeto è

$$\Sigma_P = \{A,V,L,I,P,F,W,M,G,S,T,C,Y,N,Q,D,E,K,R,H\}$$

mentre nel caso di sequenze di DNA:

$$\Sigma_{DNA}=\{G,A,T,C\}$$

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

Figura 1: Simboli degli aminoacidi

Nucleotide	Simbolo
Guanine	G
Adenine	A
Thymine	T
Cytosine	C

Figura 2: Simboli dei nucleotidi

2.2 Allineamenti a coppie

Come abbiamo visto la sequenza di aminoacidi di una proteina determina la sua forma tridimensionale e perciò anche la sua funzione biochimica. Nell'analisi delle sequenze biologiche risultano di fondamentale importanza le tecniche di confronto: infatti se due proteine presentano una sequenza di aminoacidi simile, allora, molto probabilmente presenteranno anche caratteristiche biochimiche simili.

Un confronto rigoroso fra due o più sequenze biologiche può essere fatto per mezzo di un allineamento. Un allineamento infatti permette di mettere in evidenza come una sequenza può essere ottenuta da un'altra a tramite una serie di operazioni di sostituzione, cancellazione e inserimento di aminoacidi. Se questa serie di operazioni è fisicamente probabile significa che le due sequenze sono vicine dal punto di vista biologico o evolutivo; in tal caso le sequenze possono rappresentare proteine con struttura e funzioni simili.

In termini più rigorosi, un allineamento di due sequenze **a** e **b** si ottiene inserendo in esse alcuni gap symbol in modo da ottenere due nuove sequenze **a'** e **b'** aventi stessa lunghezza L:

	1	2	3	…	L
a'	a'_1	a'_2	a'_3	…	a'_L
b'	b'_1	b'_2	b'_3	…	b'_L

Per esempio date le sequenze **a** = ADCGIHL e **b** = NADCCL, un possibile allineamento è il seguente:

	1	2	3	4	5	6	7	8
a'	–	A	D	C	G	I	H	L
b'	N	A	D	C	C	–	–	L

Questo tipo di allineamento fra due sequenze è detto "a coppie" per distinguerlo da quello multiplo che confronta più di due sequenze.

Una volta che due sequenze sono state allineate possono essere confrontate carattere per carattere tenendo conto che per la generica coppia di caratteri (a'_i,b'_i) vale la seguente interpretazione in termini di sostituzioni, inserimenti e cancellazioni:

$(a'_i,b'_i) = (a,a) \Rightarrow$ adattamento (matching)

$(a'_i,b'_i) = (a,b) \Rightarrow$ sostituzione di a con b in **a** (o sostituzione di b con a in **b**)

$(a'_i,b'_i) = (–,b) \Rightarrow$ inserimento di b in **a** (o cancellazione di b da **b**)

$(a'_i,b'_i) = (a,–) \Rightarrow$ cancellazione di a da **a** (o inserimento di a in **b**)

Notare le dualità fra le operazioni di sostituzione e inserimento: un inserimento in una sequenza può essere visto come una cancellazione sull'altra sequenza.

Le operazioni di sostituzione, inserimento e cancellazione per le coppie (a'_i,b'_i) i=1,2,…,L sono tutte le operazioni che permettono di trasformare la sequenza **a** nella sequenza **b** o viceversa.

Riprendendo l'esempio precedente, possiamo dire che la sequenza **a** può essere trasformata nella sequenza **b** tramite un inserimento di N nella posizione 1, una

sostituzione di G con C nella posizione 5 e una cancellazione di I e H nelle posizioni 6 e 7. Un discorso simile si può fare nella trasformazione da **b** ad **a**.
In generale un allineamento è detto ottimo quando rappresenta correttamente la relazione evolutiva e/o funzionale fra due o più sequenze.

2.3 Similarità

La similarità è un concetto abbastanza semplice dal punto di vista intuitivo: due proteine sono simili se presentano caratteristiche simili o se hanno funzioni biologiche simili. Siccome le proprietà di una proteina sono determinate dalla sua sequenza di aminoacidi è lecito supporre che la similarità fra sequenze proteiche abbia una stretta relazione con la similarità funzionale delle proteine stesse.
Il modello classico usato per la misura della similarità fra aminoacidi è la funzione di punteggio che associa ad ogni coppia di aminoacidi un valore numerico intero che indica la facilità con cui gli aminoacidi a e b possono sostituirsi durante il processo evolutivo senza compromettere la funzione:

$$P = P(a,b) \text{ con } a,b \in \Sigma .$$

Di conseguenza c'è da aspettarsi che P(a,b) i valori maggiori per a=b in quanto è più facile che un aminoacido non subisca nessuna sostituzione piuttosto che la subisca. Il modello inoltre prevede che P(a,b) = P(b,a), cioè che a possa sostituire b con la stessa facilità con cui b può sostituire a.
La funzione di punteggio viene normalmente rappresentata tramite una matrice triangolare P_{ab} = P(a,b) detta matrice di sostituzione. La matrice più semplice che si possa immaginare è la matrice identità la quale pesa in maniera uniforme tutte le diversità (0) e le similarità (1) presenti fra gli aminoacidi.
In Figura 3 è mostrata una matrice di sostituzione molto usata nelle applicazioni: la matrice PAM250 [Dayhoff78].
Osservando per esempio che la riga W della matrice PAM250 si notano valori molto bassi, possiamo intuire che l'aminoacido W non si accoppia facilmente con nessun altro aminoacido eccetto che con R, F e W stesso.
La funzione di punteggio può essere estesa in modo da includere nel proprio dominio anche i gap. In tal caso P(a,–) rappresenta la probabilità con cui

l'aminoacido *a* viene cancellato da una sequenza, mentre P(–,b) rappresenta la probabilità con cui l'aminoacido *b* viene inserito in un sequenza.

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	2																			
R	-2	6																		
N	0	0	2																	
D	0	-1	2	4																
C	-2	-4	-4	-5	12															
Q	0	1	1	2	-5	4														
E	0	-1	1	3	-5	2	4													
G	1	-3	0	1	-3	-1	0	5												
H	-1	2	2	1	-3	3	1	-2	6											
I	-1	-2	-2	-2	-2	-2	-2	-3	-2	5										
L	-2	-3	-3	-4	-6	-2	-3	-4	-2	2	6									
k	-1	3	1	0	-5	1	0	-2	0	-2	-3	5								
M	-1	0	-2	-3	-5	-1	-2	-3	-2	2	4	0	6							
F	-3	-4	-3	-6	-4	-5	-5	-5	-2	1	2	-5	0	9						
P	1	0	0	-1	-3	0	-1	0	0	-2	-3	-1	-2	-5	6					
S	1	0	1	0	0	-1	0	1	-1	-1	-3	0	-2	-3	1	2				
T	1	-1	0	0	-2	-1	0	0	-1	0	-2	0	-1	-3	0	1	3			
W	-6	2	-4	-7	-8	-5	-7	-7	-3	-5	-2	-3	-4	0	-6	-2	-5	17		
Y	-3	-4	-2	-4	0	-4	-4	-5	0	-1	-1	-4	-2	7	-5	-3	-3	0	10	
V	0	-2	-2	-2	-2	-2	-2	-1	-2	4	2	-2	2	-1	-1	-1	0	-6	-2	4

Figura 3: Matrice di sostituzione PAM250

A questo punto la similarità fra due sequenze allineate **a'** e **b'** può essere misurata in termini di punteggio dell'allineamento dato dalla somma dei punteggi relativi alle coppie di aminoacidi delle due sequenze allineate in esame:

$$P(\mathbf{a'},\mathbf{b'}) = \sum_{i=1}^{L} P(a'_i, b'_i)$$

In generale il meccanismo di attribuzione del punteggio associato ai gap, detto "gap model", può essere più complesso di quello appena visto. E' molto utilizzato un modello di punteggio che associa ad ogni gap un costo del tipo:

$$Cost = g_o + g_e \cdot L$$

dove ad un contributo legato alla presenza di un gap g_o si aggiunge un contributo proporzionale alla sua lunghezza $g_e \cdot L$.

2.4 Allineamento ottimo

Veniamo ora al problema dell'allineamento ottimo definito come segue:

*"Date due sequenze **a** e **b**, una matrice di sostituzione P e un gap model, trovare l'allineamento **a'** e **b'** che massimizza il punteggio P(**a'**,**b'**) "*

La soluzione del problema dell'allineamento ottimo permette di trovare l'allineamento che meglio degli altri rappresenta la relazione evolutiva o la relazione funzionale fra le due sequenze di partenza.

Si parlerà più diffusamente degli algoritmi di allineamento nei paragrafi successivi; per adesso ci limitiamo ad osservare come la bontà di un allineamento ottimo così come la chiave di lettura del risultato ottenuto dipenda fortemente dalla scelta della matrice di sostituzione e dal gap model.

Le prime matrici di sostituzione sono quelle della serie PAM proposte da Dayhoff [Dayhoff78]. Per definizione un'unità PAM (point-accepted-mutation) è il periodo evolutivo lungo il quale cambiano 1% degli aminoacidi di una proteina.

Partendo da allineamenti di un set di 71 gruppi di sequenze che presentano almeno l'85% di aminoacidi identici si calcola, per ogni coppia di aminoacidi *i* e *j*, il numero di sostituzioni accettate durante il processo evolutivo, cioè il numero di volte che avviene lo scambio fra i due aminoacidi senza compromettere la funzionalità della proteina. Si ottiene la matrice A_{ij} delle cosiddette "target frequencies". A questo punto si ricava la propensione m_i dell'aminoacido *i* ad essere sostituito e la probabilità M_{ij} di avere una mutazione che coinvolge gli aminoacidi i e j in un certo periodo di evoluzione:

$$m_i = \frac{A_{ij}}{f_i} \qquad M_{ij} = \frac{m_j A_{ij}}{\sum_i A_{ij}}$$

dove f_i è la frequenza dell'aminoacido *i* nell'allineamento (background frequencies) e $M_{jj} = 1 - m_j$. Normalizzando rispetto alla frequenza e applicando il logaritmo si ottiene la matrice PAM1:

$$PAM1_{ij} = 10 \cdot \log \frac{M_{ij}}{f_i}$$

La matrice M_{ij} può essere usata per simulare l'evoluzione di una sequenza: infatti applicando ad una sequenza **s** delle trasformazioni casuali con distribuzione di

probabilità definite da M_{ij} si ottiene una nuova sequenza **s'** che può essere considerata come l'evoluzione di **s** in un dato periodo. Per avere una trasformazione relativa a N periodi evolutivi è sufficiente applicare la trasformazioni N volte, e questo equivale a moltiplicare la matrice M_{ij} per se stessa N volte. In questo modo si ottiene la serie PAMN delle matrici PAM.

Notare che in questo modello si assume che il processo evolutivo di una proteina possa essere scomposto in una successione di singoli processi mutazionali fra loro indipendenti ognuno dei quali corrisponde ad un periodo evolutivo. Le matrici PAM, per come sono costruite, sono molto adatte nel caso si voglia scoprire una relazione di omologia fra due o più sequenze.

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

Figura 4: Matrice di sostituzione BLOSUM62

Un'altra serie di matrici di sostituzione molto utilizzate sono le BLOSUM [Henikoff92] ottenute in maniera simile alle PAM ma con una strategia differente per il calcolo delle frequenze, che viene fatto considerando blocchi senza gap di allineamenti multipli fra sequenze appartenenti alla stessa famiglia. Per la generica matrice BlosumN si calcolano i punteggi considerano blocchi con percentuale di identità minima pari a N. In Figura 4 è mostrata la matrice Blosum62.

A differenza delle matrici PAM non si fa nessuna ipotesi esplicita sul processo evolutivo del quale si tiene conto nel fatto che le sequenze appartengono ad una stessa famiglia e perciò omologhe. Inoltre le frequenze sono calcolate su blocchi

conservati per cui queste matrici sono più adatte ad allineamenti in cui si vuole mettere in evidenza i domini funzionali.
In generale bisogna dire che le funzioni di punteggio per quanto accurate sono soltanto una stima delle similarità o dell'omologia. Infatti molti esempi dimostrano che esistono sequenze con similarità stimata elevata che però non sono simili dal punto di vista della struttura tridimensionale e della funzione svolta.
La complessità nei confronti di biosequenze è dovuta soprattutto a questa difficoltà intrinseca di trovare una funzione di punteggio generale che tenga conto di tutte le caratteristiche importanti di una sequenza.

2.5 Tecniche di allineamento a coppie

In questo paragrafo vedremo alcune delle tecniche più usate nei problemi di allineamento. Nel seguito si farà una distinzione fra allineamenti globali e locali: nei primi le sequenze vengono confrontate in tutta la loro lunghezze mentre nei secondi il punteggio complessivo è calcolato soltanto su sottosequenze. Gli allineamenti locali sono utilizzati per analizzare le funzionalità di determinate sottosequenze e per determinare gli eventuali siti attivi.
Un altro concetto importante è quello di allineamento subottimo, in cui non è richiesto di raggiungere il massimo della funzione di punteggio ma ci si accontenta di un valore vicino.

2.5.1 Algoritmo di Seller

Vediamo innanzitutto un algoritmo di allineamento molto generale e su cui si basano molti altri metodi più evoluti: l'algoritmo di Seller detto anche "Algoritmo di Programmazione Dinamica".
Si basa sull'osservazione che da un allineamento ottimo di lunghezza L posso eliminare l'ultima colonna, ottenendo un allineamento di lunghezza L−1 che è ancora ottimo.
Date due sequenze $\mathbf{a}=a_1a_2\ldots a_n$ e $\mathbf{b}=b_1b_2\ldots b_m$, indichiamo con H(i,j) il punteggio dell'allineamento ottimo delle sottosequenze $a_1a_2\ldots a_i$ e $b_1b_2\ldots b_j$.

Tale punteggio può essere espresso con la seguente relazione ricorrente:

$$H(i,j) = \max \begin{cases} H(i-1, j-1) + P(a_i, b_j) \\ H(i, j-1) + P(-, b_j) \\ H(i-1, j) + P(a_i, -) \end{cases} \qquad i,j>0$$

con le condizioni iniziali:

$$H(0,0) = 0\,,\ H(0,j) = H(0,j-1) + P(-,b_j)\ \text{ e }\ H(i,0) = H(i-1,0) + P(a_i,-)$$

Si tratta di una relazione abbastanza intuitiva: dato un allineamento ottimo di lunghezza k possiamo aggiungere a tale allineamento una colonna k+1; per ottenere un allineamento che sia ancora ottimo la colonna aggiunta deve essere tale da massimizzare il punteggio complessivo.

Come mostrato in Figura 5 il generico elemento H(i,j) della matrice può essere visto come il vertice di un quadrato 2×2 in cui gli altri elementi influiscono nel calcolo di H(i,j) ma è solo uno degli elementi (il massimo) a determinarlo.

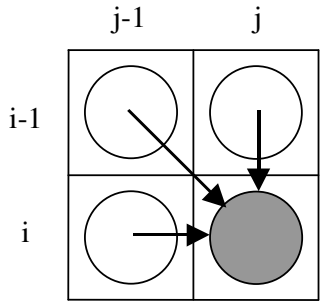


Figura 5: Calcolo dell'elemento H(i,j)

		G	T	T	A	T
	0	−1	−3	−4	−5	−6
A	−1	2	1	−1	−2	−3
G	−3	3	2	1	0	−1
G	−4	−1	4	4	−1	−2
T	−5	1	3	5	2	4
T	−6	−4	3	7	6	10

Sequenze

A	**G**	**G**	**T**	**T**
G	**T**	**T**	**A**	**T**

Allineamento (Punteggio=10)

A	**G**	**G**	**T**	**–**	**T**
–	**G**	**T**	**T**	**A**	**T**

Figura 6: Algoritmo di programmazione dinamica

Posso partire perciò da un allineamento banale di lunghezza k=0 (punteggio 0) e costruire l'allineamento aumentando passo dopo passo la sua lunghezza fino a k=L.

L'applicazione di questo metodo porta alla costruzione di una matrice H di dimensione (n+1)×(m+1) dove le righe corrispondono agli elementi di **a** e le colonne agli elementi di **b**. Su questa matrice risulta definito un percorso dall'elemento (n,m) all'elemento (0,0) che rappresenta l'allineamento ottimo. Tale percorso è basato sulla relazione ricorsiva precedente; in particolare a seconda di quale dei tre valori sia il massimo si ha il seguente schema:

$H(i,j) = H(i-1,j-1) + P(a_i, b_j)$	spostamento diagonale: sostituzione a_i con b_j
$H(i,j) = H(i,j-1) + P(-, b_j)$	spostamento a destra: inserimento di b_j in **a**
$H(i,j) = H(i-1,j) + P(a_i, -)$	spostamento in alto: cancellazione di a_i da **a**

In Figura 6 è mostrato un esempio di allineamento ottimo con matrice H e relativo percorso.

Si tratta di un algoritmo di allineamento globale e ottimo in quanto la ricerca viene fatta su tutti i possibili allineamenti fra le due sequenze. Per come è stato esposto l'algoritmo presenta una complessità computazionale pari al numero di elementi della matrice ovvero $O\big((n+1)\cdot(m+1)\big)$ più un certo numero di operazioni necessarie per ricostruire a ritroso l'allineamento. In realtà esistono diversi miglioramenti che possono migliorare l'efficienza (O(m) per m<n) e l'utilizzo di memoria.

2.5.2 Algoritmo di Wunsch-Needleman

L'algoritmo proposto da Needleman e Wunsch nel 1970 [Needle70] è stato uno dei primi algoritmi di allineamento. Anche se formalmente differente si basa sull'algoritmo di programmazione dinamica precedentemente descritto per cui è anche questo un metodo di allineamento ottimo globale.

Si parte da una matrice H (n+1)×(m+1) simile a quella vista in precedenza in cui le righe e le colonne sono associate alle sequenze da allineare che inizialmente viene riempita con i valori della matrice di sostituzione e poi viene aggiornata con operazioni di somma sulle righe. La matrice H finale presenta un percorso dall'elemento (0,0) all'elemento (n,m) che rappresenta l'allineamento.

La particolarità di questo algoritmo è che permette l'utilizzo di modelli di gap più complessi e in particolare quello basato sul costo di apertura e lunghezza del gap che abbiamo visto precedentemente: $Cost = g_o + g_e \cdot L$.

2.5.3 Algoritmo di Smith-Waterman

Tramite una semplice modifica la formula della programmazione dinamica può essere estesa al caso di allineamenti locali ottimi. La modifica consiste nel porre a zero tutti i valori di H(i,j) minori di 0 sia nella formula ricorsiva che nelle condizioni iniziali.

Costruita la matrice H si può trovare l'allineamento ottimo seguendo il percorso che va, anziché dall'elemento (n,m), dall'elemento di massimo punteggio al primo elemento nullo che si trova nel percorso.

In formule:

$$H(i,j) = \max \begin{cases} 0 \\ H(i-1, j-1) + P(a_i, b_j) \\ H(i, j-1) + P(-, b_j) \\ H(i-1, j) + P(a_i, -) \end{cases} \qquad i,j>0$$

con le condizioni iniziali: $H(0,0) = H(0,j) = H(i,0) = 0$.

In Figura 7 è mostrato l'allineamento locale delle sequenze dell'esempio precedente:

		G	**T**	**T**	**A**	**T**
	0	0	0	0	0	0
A	0	2	1	0	0	0
G	0	3	2	1	0	0
G	0	1	4	4	2	0
T	0	0	3	7	2	4
T	0	0	0	3	4	5

Sequenze

A	**G**	**G**	**T**	**T**
G	**T**	**T**	**A**	**T**

Allineamento (Punteggio =7)

G	**G**	**T**
G	**T**	**T**

Figura 7: Algoritmo Smith-Waterman

Notare che l'allineamento locale in generale è più corto di quello globale in quanto mette in evidenza le due sottosequenze che più simili.

2.5.4 Algoritmi BLAST e FASTA

In questo paragrafo introduciamo brevemente una classe di algoritmi sub-ottimi molto importante che costruisce l'allineamento a partire da match di sottosequenze di lunghezza k. In realtà BLAST e FASTA rientrano nella categoria dei programmi di ricerca di similarità di una sequenza query all'interno di un database ma producono un output in forma di allineamento della sequenza query con le sequenze del database con cui hanno mostrato una similarità significativa.

L'algoritmo FASTA [Pearson88] è un metodo euristico per la comparazione di stringhe che può essere rappresentato con i seguenti passi:

1. Per ogni sequenza del database si cercano le sottostringhe di lunghezza ktup dette "hot spot" che si adattano ad analoghe sottostringhe della sequenza query. Tali sottostringhe vengono posizionate lungo la diagonale nella matrice della programmazione dinamica.
2. Si cercano nella matrice le "diagonal runs" ovvero le sequenze formate da hot spot vicini anche se non necessariamente adiacenti. Ad ognuna di esse viene assegnato un punteggio dato da un valore positivo per ogni hot spot e un valore negativo proporzionale alle distanze fra gli hot spot. In questo modo vengono selezionate le 10 diagonal runs con punteggio maggiore.
3. Si calcolano i punteggi dei suballineamenti considerando la diagonal runs che presentano un miglior punteggio utilizzando una matrice di sostituzione. Il migliore suballineamento trovato in questo stadio è detto $init_1$.
4. Si cerca di allargare gli allineamenti combinando fra loro i suballineamenti che hanno presentato punteggio migliore. Il migliore suballineamento trovato in questo stadio è detto $init_n$.
5. Si trova un ulteriore allineamento detto opt utilizzando l'algoritmo di programmazione dinamica su una banda centrata nella diagonale definita da $init_1$.
6. Infine si utilizza l'algoritmo di programmazione dinamica completo (su tutta la matrice) per allineare la sequenza query con tutte le sequenze del database che hanno dato punteggi otp e $init_n$.

L'algoritmo FASTA non garantisce l'ottimalità dell'allineamento ma produce comunque risultati soddisfacenti e ha il vantaggio di una maggiore velocità rispetto

all'algoritmo di programmazione dinamica applicato a tutte le sequenze del database.

Il programma BLAST [Altschul90] è stato sviluppato con lo scopo di migliorare sia la velocità che la qualità dell'allineamento rispetto a FASTA. L'idea consiste nel selezionare un numero minore di hot spots durante l'algoritmo ma di utilizzare l'analisi con matrice di sostituzione a partire da primo passo in cui si seleziona la lista degli hot spots.

Vediamo i passi dell'algoritmo:

1. Fissata una lunghezza di riferimento w e un valore di soglia t, BLAST cerca e memorizza tutte le sottostringhe del database di lunghezza w che allineate senza gap con analoghe sottostringhe della query danno un punteggio maggiore di t. Queste sottostringhe sono dette hit. Normalmente il parametro w assume valori da 3 a 5 per gli aminoacidi e circa 12 per i nucleotidi.
2. Gli hit vengono estesi per vedere se sono contenuti in coppie di segmenti (detti HSP) con punteggio maggiore di un valore di soglia S. Il valore S è scelto a partire dalla sequenza query in modo che sia poco probabile trovare per puro caso nel database sottosequenze con punteggio maggiore di S.
3. Vengono fornite in uscita gli allineamenti con le sequenze del database che hanno fornito un punteggio maggiore

La significatività degli hits viene valutata considerando la probabilità che un HSP abbia punteggio superiore a t: $P(H(a,b) > t)$; $1 - e^{-\gamma \cdot n \cdot m \cdot p^t}$ dove γ è un parametro calcolabile numericamente, n e m sono le lunghezze delle due sequenze e p è la probabilità di avere un match.

Al crescere di *t* aumenta la probabilità di trovare degli hits e di conseguenza diminuisce la significatività della ricerca. Il parametro t, rappresenta perciò la sensibilità della ricerca e va scelto con cura affinché l'algoritmo riesca a trovare tutte le sequenze significative del database senza introdurre sequenze la cui similarità con la query sia soltanto casuale.

I programmi PSI-BLAST e PHI-BLAST [Altschul97] sono distribuiti insieme a BLAST e utilizzano l'algoritmo precedente in maniera iterativa per migliorare passo dopo passo la qualità dell'allineamento.

In PSI-BLAST, al passo *k* dell'algoritmo si utilizza il programma BLAST per trovare le sequenze del database allineate con la query. A partire da questo allineamento viene calcolata una matrice di punteggio position-specific che sarà usata come schema di punteggio per il passo successivo.

In PHI-BLAST invece data in ingresso una sequenza query S insieme a un pattern P (espressione regolare) in essa contenuto si cercano in tutto il database le sequenze che si adattano a P e che contemporaneamente presentano una relazione di omologia con S. La costruzione dell'allineamento locale avviene in maniera simile a BLAST.

2.6 Allineamenti multipli

I confronti di più di due sequenze possono essere fatti per mezzo di allineamenti multipli che possono essere definiti come semplice estensione di quelli a coppie. Un allineamento multiplo fra N sequenze $\mathbf{s_1}$, $\mathbf{s_2}$, …, $\mathbf{s_N}$ è semplicemente una matrice $N \times L$ ottenuta aggiungendo dei gap alle sequenze in modo che assumano la stessa lunghezza L.

In Figura 8 è mostrato un allineamento di 5 sequenze. I concetti di gap, sostituzione, cancellazione e inserimento hanno lo stesso significato in quanto l'allineamento multiplo può essere visto come un allineamento semplice fatto su tutte le coppie $\mathbf{s_i'}$ e $\mathbf{s_j'}$ per cui un gap sulla sequenza $\mathbf{s_1'}$ può essere pensato come un inserimento in $\mathbf{s_1'}$ di un aminoacido da parte di tutte le sequenze dell'allineamento (a parte quelle che presentano un gap nella stessa posizione).

	1	2	3	4	5	6	7	8
$\mathbf{s_1'}$	–	A	D	C	G	I	H	L
$\mathbf{s_2'}$	N	A	D	C	C	–	–	L
$\mathbf{s_3'}$	N	A	D	E	K	–	H	S
$\mathbf{s_4'}$	N	D	D	–	K	I	S	S
$\mathbf{s_5'}$	–	D	D	E	–	I	H	S

Figura 8: Allineamento multiplo

Un allineamento multiplo è ottimo quando è tale da massimizzare una funzione di punteggio $P(\mathbf{s_1'},\mathbf{s_2'},...,\mathbf{s_N'})$ dipendente dalle relazioni fra tutte le sequenze in esame.

E' possibile definire molti tipi di funzione di punteggio, il più semplice è quello di calcolare la somma di tutti i punteggi di tutte le coppie di sequenze $\mathbf{s_i'}$ e $\mathbf{s_j'}$:

$$P(\mathbf{s_1'}, \mathbf{s_2'}, ..., \mathbf{s_N'}) = \sum_{\mathbf{i>j}} P(\mathbf{s_i'}, \mathbf{s_j'}) = \sum_{\mathbf{i>j}} \sum_{p=1}^{L} P(s_{ip}', s_{jp}')$$

dove P è una matrice di sostituzione.

Come gli allineamenti a coppie, anche quelli multipli possono rivelare informazioni utili sulle relazioni evolutive e funzionali fra sequenze in esame. Infatti se alcune di queste sequenze hanno una struttura nota e l'allineamento ha dato un esito positivo, è possibile predire le funzionalità delle proteine sconosciute (struttura secondaria, domini) e scoprire eventuali siti attivi.

Gli allineamenti multipli sono molto importanti anche nei problemi di classificazione di proteine in famiglie: da un allineamento multiplo di un campione di sequenze appartenenti ad una famiglia è possibile ricavare dei pattern caratteristici che vengono usati successivamente per verificare l'appartenenza di una sequenza in esame nella famiglia stessa.

Un'altra importante applicazione degli allineamenti multipli è la predizione di alberi evolutivi. Un albero evolutivo è un albero le cui foglie sono le specie esistenti esaminate, i nodi interni sono le specie incognite che possono essere considerate come i progenitori delle specie in esame. Il nodo radice rappresenta il più antico dei progenitori e i rami possono essere etichettati con il periodo temporale che separa le due specie.

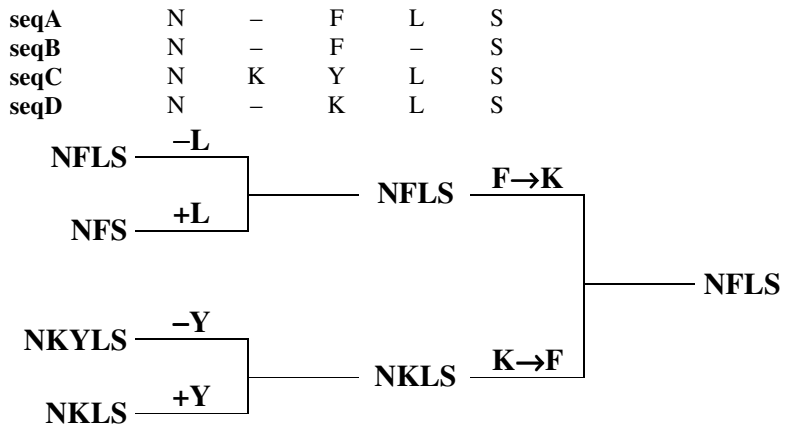


Figura 9: Albero evolutivo

Un albero evolutivo può essere riferito sia a specie di esseri viventi sia, come nel nostro caso, a singole sequenze proteiche. In tal caso, assumendo che in un allineamento multiplo sequenze che presentano una similarità maggiore sono da considerare diretti discendenti di un'unica sequenza progenitrice, si costruisce l'albero a partire dalle sequenze in esame accoppiando a due a due le sequenze che presentano nel allineamento una similarità maggiore di una certa soglia. Nei passi successivi si procede a raggruppare i nodi in base alle similarità delle sequenze figlie fino ad arrivare al nodo radice.
In Figura 9 è mostrato un esempio di albero evolutivo costruito a partire da un allineamento di 4 sequenze.

Nei paragrafi successivi vengono presentati alcuni algoritmi di allineamento multiplo. Notare che dal punto di vista concettuale è possibile calcolare l'allineamento ottimo con la tecnica della programmazione dinamica rappresentando l'allineamento come percorso in una matrice N-dimensionali.

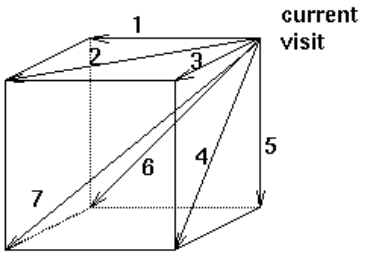


Figura 10: Calcolo dell'elemento H(i,j,k)

Per esempio caso N=3 ho sette spostamenti possibili all'interno di un cubo a tre dimensioni (Figura 10) per cui la formula ricorsiva sarà:

$$H(i,j,k) = \max \begin{cases} H(i-1,j-1,k-1) + P(a_i, b_j, c_k) \\ H(i,j-1,k-1) + P(-, b_j, c_k) \\ H(i-1,j,k-1) + P(a_i, -, c_k) \\ H(i-1,j-1,k) + P(a_i, b_j, -) \\ H(i,j,k-1) + P(-, -, c_k) \\ H(i,j-1,k) + P(-, b_j, -) \\ H(i-1,j,k) + P(a_i, -, -) \end{cases} \qquad i,j,k>0$$

con le otto condizioni iniziali:

$$H(0,0) = 0$$

$$H(i,0,0) = H(i-1,0,0) + P(a_i,-,-),\ H(0,j,0) = H(0,j-1,0) + P(-,b_j,-)$$

$$H(0,0,k) = H(0,0,k-1) + P(-,-,c_k),\ H(0,j,k) = H(0,j-1,k-1) + P(-,b_j,c_k)$$

$$H(i,0,k) = H(i-1,0,k-1) + P(a_i,-,c_k),\ H(i,j,0) = H(i-1,j-1,0) + P(a_i,b_j,-)$$

Per N sequenze di lunghezza n la complessità computazionale dell'algoritmo lo rende inutilizzabile nelle applicazioni pratiche. Per questo motivo sono state studiate sia delle tecniche per ridurre tale complessità (limitando lo spazio di ricerca all'interno della matrice N-dimensionale), sia dei metodi di allineamento differenti.

2.7 Tecniche di allineamento multiplo

In questo paragrafo vedremo brevemente alcuni modelli di allineamento multiplo che si basano sostanzialmente su euristiche per la riduzione dello spazio di ricerca, indispensabili per rendere il problema trattabile sebbene portino a soluzioni subottime.

In particolare faremo riferimento alle tecniche di tipo progressivo che consistono nel costruire l'allineamento aggiungendo progressivamente le sequenze ad una sequenza iniziale, e alle tecniche di tipo iterativo in cui l'allineamento è costruito con una successione passi di modifica dell'allineamento precedente con lo scopo di migliorare una funzione di punteggio complessiva.

2.7.1 MSA

L'algoritmo proposto da Carrillo-Lipman nel 1988 [Carrillo88], implementato nel programma MSA, si basa sull'osservazione che spesso il percorso sulla matrice H che rappresenta l'allineamento ottimo, non si discosta di molto da un intorno V in prossimità della diagonale della matrice stessa. E' perciò possibile vincolare l'allineamento a stare in tale intorno limitando notevolmente lo spazio di ricerca.

Dal punto di vista matematico abbiamo un problema di ottimizzazione vincolata o di ricerca vincolata di percorso ottimo:

$$\max\{P(\mathbf{s_1}', \mathbf{s_2}', ..., \mathbf{s_N}')\} \qquad (\mathbf{s_1}', \mathbf{s_2}', ..., \mathbf{s_N}') \in V$$

Le regole per il calcolo di H sono le stesse della programmazione dinamica con l'unica differenza che nel calcolo del generico elemento entrano in gioco gli elementi del cubo N-dimensionale escludendo quelli che non appartengono a V.

L'aspetto fondamentale dell'algoritmo è la scelta della zona consentita in modo da arrivare rapidamente ad un buon risultato. Una scelta non accurata può portare a risultati completamente sbagliati senza la possibilità di feedback.

L'intorno V può essere valutato in diversi modi: avendo a disposizione una stima di allineamento ottimo trovato con altri metodi è possibile utilizzarlo come punto di partenza "espandendolo" in modo da riempire una certa zona V della matrice H; oppure è possibile calcolare gli allineamenti di tutte le coppie ($\mathbf{s_i}$,$\mathbf{s_j}$), che nella matrice H sono delle proiezioni sugli assi i e j, espanderli e intersecarli per generare un insieme V da usare come vincolo. Quest'ultimo caso è un tipico esempio di allineamento multiplo pilotato da allineamenti semplici.

Lo svantaggio principale di questo metodo consiste nella difficoltà a trovare un vincolo V che sia abbastanza piccolo da ridurre la complessità ma abbastanza preciso da includere la soluzione ottima. Infatti più V è piccolo, maggiore è la possibilità di escludere la soluzione cercata, mentre un allargamento eccessivo di V porterebbe una perdita notevole di efficienza.

2.7.2 ClustalW

L'algoritmo proposto da Thompson e Higgins, implementato nel programma ClustalW [Thompson94], rientra nella classe dei metodi progressivi ed è uno dei programmi più utilizzati.

L'algoritmo è composto da tre stadi fondamentali:

1) Allineamento a coppie di tutte le coppie $\mathbf{s_i}$,$\mathbf{s_j}$ in modo da calcolare la matrice delle divergenze δ_{ij} che indica il grado di diversità di ogni coppia.
2) Costruzione dell'"albero guida" in cui le foglie rappresentano le sequenze e i nodi rappresentano gruppi di sequenze allineate. L'ordine del raggruppamento

è quello suggerito dalla matrice δ_{ij}. La tecnica di costruzione è di tipo Neighbour-Joining, cioè vengono uniti due nodi che risultano essere più vicini in modo che alla fine il nodo radice è posizionato nel punto in cui le lunghezze dei rami a sinistra e destra sono uguali.

3) Allineamento progressivo delle sequenze seguendo l'albero guida. L'idea di base consiste nel utilizzare una serie di allineamenti a coppie in modo da allineare gruppi sempre più grandi di sequenze seguendo l'ordine suggerito dall'albero guida, procedendo dalle foglie verso la radice.

Ogni passo dell'algoritmo consiste dunque nel raggruppare due nodi in modo da creare un nuovo nodo seguendo la guida dell'albero e per questo è stato introdotto il concetto di allineamento di due allineamenti; viene utilizzato l'algoritmo di programmazione dinamica mantenendo fissi i gap dei due allineamenti iniziali, che vengono considerati semplicemente come nuovi simboli dell'alfabeto, e aggiungendo nuovi gap indipendenti dai precedenti. Il calcolo dei gap al passo k è indipendente da quello effettuato al passo k+1 nel senso che l'inserimento di un nuovo gap in un gap gia esistente è considerato di nuovo come un gap opening.
In Figura 11 è mostrato un esempio di creazione di un allineamento di quattro sequenze a partire dai due allineamenti a coppie; i nuovi gap aggiunti, evidenziati in grigio, hanno un costo che non dipende dalla presenza degli altri gap.

Figura 11: Allineamento di due allineamenti

Nel calcolare il punteggio del confronto fra due posizioni dell'allineamento si considera il valore medio del punteggio associato a tutte le coppie di aminoacidi che entrano in gioco nel confronto, associando punteggio nullo ai confronti con i gap già esistenti.
Durante il procedimento non tutte le sequenze sono pesate allo stesso modo nell'allineamento multiplo: si utilizzano dei pesi w_{ij}, calcolati a partire dall'albero guida, che danno maggiore importanza alle coppie di sequenze più divergenti con

lo scopo di evitare che l'allineamento multiplo sia pilotato da sequenze troppo simili fra loro. In Figura 12 è mostrato lo schema della costruzione dell'allineamento.

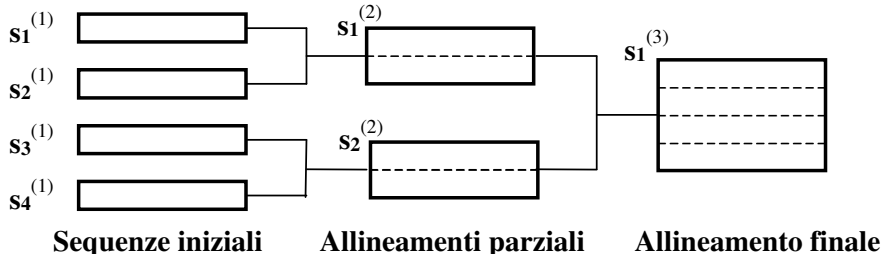


Figura 12: Costruzione dell'allineamento multiplo

La funzione di punteggio assume una forma del tipo:

$$P(\mathbf{s_1'}, \mathbf{s_2'}, ..., \mathbf{s_N'}) = \sum_{\mathbf{i>j}} w_{\mathbf{ij}} \cdot \sum_{p=1}^{L} P(s_{ip}', s_{jp}')$$

Il meccanismo di punteggio in realtà è più complicato di quello previsto nella programmazione dinamica in particolare per quanto riguarda il costo dei gap. I valori di gap opening e gap extension vengono assegnati ad ogni passo, tranne all'inizio in cui sono definiti dall'utente, con un meccanismo position-specific che tiene conto di diversi fattori soprattutto della presenza di gap nelle altre posizioni. Infatti viene aggiornata ad ogni passo una tabella che tiene conto dei gap presenti ad ogni posizione e viene così calcolata la propensione di ogni zona dell'allineamento a presentare un gap.

La matrice di sostituzione viene variata ad ogni passo in funzione della distanza fra le sequenze (o gruppi di sequenze) che si stanno allineando. Questo rende il metodo capace di adattarsi maggiormente al caso in esame.

I pregi del metodo ClustalW riguardano soprattutto l'adattabilità dei parametri dell'allineamento alla particolare situazione in cui si trova. Questo rende l'algoritmo di applicabilità molto più generale rispetto ad altri metodi che utilizzano informazioni aggiuntive note a priori spesso incomplete. ClustalW infatti ricava queste informazioni direttamente dalle sequenze in esame.

I limiti sono legati sostanzialmente alla natura “greedy” dell’algoritmo: il rischio di fermarsi su massimi locali molto distanti dal massimo assoluto. Il verificarsi di un errore o un imprecisione in un qualsiasi punto dell’algoritmo infatti si ripercuote sino alla fine dell’algoritmo senza la possibilità di tornare indietro. Questo problema si accentua enormemente se le sequenze sono fra loro molto distanti in quanto è molto facile avere degli errori sui primi allineamenti (foglie dell’albero).

2.7.3 DIALIGN

Il metodo DIALIGN [Morgenstern96] è un metodo iterativo basato sul confronto di segmenti piuttosto che di residui. Segmenti di uguale lunghezza e privi di gap sono detti diagonali in quanto sono rappresentati da diagonali nella matrice dot. Un set di diagonali disgiunte e ordinate secondo la direzione delle sequenze è detto consistente.

Il primo passo consiste nel creare gli allineamenti di tutte le coppie dai quali vengono estratte le diagonali e ordinate in base ad un punteggio. Queste diagonali sono usate per costruire un allineamento multiplo in maniera greedy aggiungendo le diagonali all’allineamento a partire da quella con punteggio maggiore. A questo punto si ricalcolano iterativamente gli allineamenti a coppie e si crea un’altra lista di diagonali trascurando quelle già inserite nell’allineamento multiplo. Il processo continua fino a quando non vengono più trovate altre diagonali.

Come passo finale vengono aggiunti dei gap allo scopo di connettere le diagonali e riarrangare correttamente i residui. Nell’uscita finale vengono evidenziati i residui che appartengono a una diagonale mentre gli altri residui sono considerati non allineati. Di conseguenza se le sequenze presentano solo somiglianze locali, l’algoritmo non tenta di fare un allineamento globale e il risultato può essere interpretato come allineamento locale.

A differenza degli altri metodi di tipo puntuale (confronto di singoli residui) DIALIGN si presenta come un metodo in cui l’unità del confronto sono le sottosequenze. Questo fatto lo rende molto adatto a localizzare piccole regioni conservative che non possono essere trovate con i metodi standard.

2.7.4 SAGA

Vediamo ora il metodo SAGA [Notredame96] proposto da Notredame e Higgins nel 1996 e implementato nell'omonimo programma.

Si tratta di un algoritmo iterativo di tipo genetico in cui una "popolazione" di allineamenti evolve con dei processi di nascita/morte in maniera che la selezione conservi gli individui che presentano un punteggio maggiore. Il punteggio è dato da una certa funzione obiettivo che indica la qualità dell'allineamento.

Si parte da una generazione iniziale G_0 di N individui creata casualmente. Ad ogni passo n si calcola il punteggio di ogni individuo della generazione G_n; la metà della popolazione che presenta un punteggio maggiore sopravvive nella generazione successiva G_{n+1} e genera N/2 figli mentre l'altra metà muore. In tal modo ogni generazione presenta sempre un numero costante N di individui.

Il processo di generazione dei figli è molto articolato: i potenziali genitori vengono selezionati in base ad un valore probabilistico di expected offspring (EO) calcolato a partire dal punteggio degli allineamenti; i valori di EO dei genitori scelti vengono poi decrementati per favorire l'utilizzo di altri genitori ai passi successivi.

A questo punto vengono applicati degli operatori di generazione che hanno lo scopo di creare un figlio fondendo insieme due genitori (crossovers operators) oppure modificando un solo genitore (mutations operators).

Sono stati definiti diversi tipi di operatori; i più semplici effettuano operazioni di inserimento o spostamento di gap mentre altri più complessi effettuano spostamenti di blocchi in un allineamento o da un allineamento all'altro.

La scelta dell'operatore da usare ad ogni passo è basata su una statistica degli operatori che fino a quel momento ha portato a risultati migliori.

Passo dopo passo vengono automaticamente scartati gli individui identici per garantire una certa diversità all'interno della popolazione; inoltre viene memorizzato l'individuo che fino a quel momento ha ottenuto un punteggio migliore. Se in un certo intervallo di generazioni non si registrano miglioramenti l'algoritmo viene fermato e si prende l'individuo migliore trovato fino a quel momento come allineamento finale.

2.7.5 T-Coffee

L'algoritmo T-Coffee [Notredame98] [Notredame00] può essere considerato una via di mezzo fra l'approccio progressivo e iterativo. Viene usato infatti un metodo progressivo per la costruzione del allineamento multiplo ma vengono evitati i problemi derivanti dalla natura greedy.

Partendo dalle sequenze da allineare $\mathbf{s_1}$,$\mathbf{s_2}$,…,$\mathbf{s_N}$ si costruisce una "libreria primaria" di allineamenti a coppie sia globali (ClustalW) che locali (LAlign) non necessariamente consistente nel senso che l'allineamento di una certa coppia può essere presente anche più di una volta. La libreria si presenta come una lista di coppie di residui di sequenze diverse che corrispondono alle posizioni degli allineamenti a coppie.

Il secondo passo consiste nell'assegnare un peso ad ogni coppia di residui pari alla percentuale di identità che si ha nell'allineamento a coppie a cui appartengono i due residui. Se una coppia di residui è presente sia un allineamento globale che in uno locale, vengono unite in una sola coppia avente come peso la somma dei due pesi. Questi pesi rappresentano l'accuratezza nell'allineamento multiplo che si ottiene allineando i due residui: solitamente si assume che una percentuale di identità superiore a 30% sia un buon indice di accuratezza per l'allineamento.

E' possibile migliorare notevolmente la qualità delle informazioni contenute nella libreria esaminando la consistenza di ogni coppia di residui con le analoghe coppie presenti in tutti gli allineamenti. I pesi associati ad una certa coppia sono aumentati in funzione della presenza della stessa coppia in altri allineamenti. Si ottiene in questo modo la libreria estesa.

A questo punto viene calcolato l'allineamento multiplo che meglio si adatta alla libreria. Viene usata la tecnica progressiva simile a quella di ClustalW: si calcola un albero guida a partire dalle somiglianze delle coppie di sequenze che viene usato per raggruppare le sequenze in modo da formare l'allineamento multiplo. I punteggi da associare nel confronto sono ricavati direttamente dalla libreria. Notare che non è necessario associare dei costi ai gap in quanto se ne tiene conto nella libreria stessa che è basata su allineamenti.

2.8 Significatività

Una volta ottenuto un allineamento occorre valutare la sua significatività ovvero la bontà del risultato ottenuto. Il punteggio di un allineamento non permette di per sé di valutare l'attendibilità dell'allineamento e nemmeno di fare ipotesi sulle relazioni funzionali o di omologia esistenti fra le sequenze in esame, a meno che non si conosca un valore di punteggio da usare come riferimento. La bontà di un metodo di allineamento dipende sostanzialmente da un equilibrio fra la sua sensibilità, ovvero la capacità di evidenziare relazioni di omologia, e la sua selettività, ovvero la capacità di evitare punteggi alti quando le sequenze sono scorrelate. Nei paragrafi successivi vedremo alcuni metodi per calcolare la significatività di un allineamento.

2.8.1 Metodi statistici

I metodi statistici si basano sull'idea che il punteggio ottimo di un allineamento ottenuto con un qualsiasi metodo sarà formato da due contributi: un contributo che indica l'effettiva somiglianza fra le sequenze e un contributo di natura statistica, detto "rumore statistico" o "rumore di fondo", derivante dal fatto che anche sequenze completamente scorrelate presentano un certo punteggio: $\text{Score} = \text{Score}' + \text{Noise}$ dove Score è il punteggio ottimo fornito dall'algoritmo.

Il calcolo del rumore che può essere fatto in maniera statistica per esempio allineando delle sequenze $\mathbf{r}_1,\mathbf{r}_2,\ldots,\mathbf{r}_N$ ottenute dalle sequenze iniziali $\mathbf{s}_1,\mathbf{s}_2,\ldots,\mathbf{s}_N$ scambiando casualmente un certo numero di residui. Il punteggio ottimo ottenuto può essere preso come rumore di fondo in quanto rappresenta il punteggio casuale che si ottiene allineando sequenze fra loro scorrelate ma composte dagli stessi aminoacidi delle sequenze iniziali.

Per rendere il risultato più attendibile si può ripetere l'esperimento tante volte e valutare il valore medio dei punteggi ottenuti.

Il problema di questo metodo è che considera le sequenze semplicemente come successioni di aminoacidi fra loro indipendenti trascurando le interazioni presenti fra residui vicini che sono fondamentali per determinare la struttura e la funzionalità delle proteine.

Una tecnica più efficace è quella di generare non delle nuove sequenze bensì degli allineamenti aggiungendo dei gap alle sequenze in esame $\mathbf{s_1}$,$\mathbf{s_2}$,…,$\mathbf{s_N}$ in maniera casuale. Il punteggio medio di tali allineamenti è utilizzato come rumore.
A questo punto il valore di Noise può essere usato come punto di riferimento definendo la significatività statistica percentuale:

$$\text{Statistical Significance} = \frac{\text{Score}'}{\text{Score}} = 1 - \frac{\text{Noise}}{\text{Score}}$$

Il rumore di fondo può essere considerato come un limite inferiore per il punteggio al di sotto del quale un allineamento deve essere considerato poco significativo.
L'importanza dei metodi statistici è dovuta soprattutto alla generalità e alla semplicità di applicazione. Si tratta infatti di una verifica che può sempre essere fatta. Il problema principale consiste nella scarsa accuratezza dei risultati in quanto una significatività di tipo statistico non implica necessariamente una somiglianza strutturale o una relazione di omologia. E' stato mostrata infatti l'esistenza di sequenze omologhe il cui allineamento corretto non presenta una significatività statistica rilevante.

2.9 Altri modelli di analisi

In questo paragrafo vengono presentati alcuni modelli per l'analisi biosequenze alternativi rispetto agli allineamenti. In realtà esiste un profondo legame tra i vari modelli in quanto metodi di tipo differente sono usati sequenzialmente o simultaneamente per perfezionare i risultati. Per esempio i profili possono considerarsi come la sintesi di un allineamento ma possono anche essere usati come punto di partenza per guidare il calcolo di un allineamento.
Lo scopo principale di questi modelli è quello di cercare, rappresentare e analizzare porzioni di sequenze che altamente conservate che hanno particolari funzioni biologiche.

2.9.1 Profili e Blocchi

Un profilo è una porzione di allineamento altamente conservata che viene rappresentato con una matrice avente una colonna per ogni aminoacido (gap

inclusi) e una riga per ogni posizione dell'allineamento. Il generico elemento della matrice contiene un punteggio proporzionale alla probabilità di trovare un certo aminoacido in quella posizione. Tale valore può essere pesato con i valori di una matrice di sostituzione (esempio BLOSUM62).

Le righe sono etichettate con la lettera dell'aminoacido che ha punteggio maggiore in quella riga. Le zone conservate sono date dalle posizioni in cui si ha una probabilità molto elevata di trovare un certo aminoacido; tali zone si presentano nel profilo come righe con punteggio massimo elevato.

In Figura 13 è mostrato un esempio di allineamento, e il relativo profilo, di un set proteine presenti in differenti specie ma simili dal punto di vista funzionale.

```
rhle_ecoli  GVDVLVATPG RLLDLEHQNA ....VKLDQV EILVLDEADR MLDMGFIHDI
dbp2_schpo  GVEICIATPG RLLDMLDSNK ....TNLRRV TYLVLDEADR MLDMGFEPQI
dbp2_yeast  GSEIVIATPG RLIDMLEIGK ....TNLKRV TYLVLDEADR MLDMGFEPQI
dbpa_ecoli  APHIIVATPG RLLDHLQKGT ....VSLDAL NTLVMDEADR MLDMGFSDAI
rm62_drome  GCEIVIATPG RLIDFLSAGS ....TNLKRC TYLVLDEADR MLDMGFEPQI
 p68_human  GVEICIATPG RLIDFLECGK ....TNLRRT TYLVLDEADR MLDMGFEPQI
rhlb_ecoli  GVDILIGTTG RLIDYAKQNH ....INLGAI QVVVLDEADR MYDLGFIKDI
yn21_caeel  RPHIIVATPG RLVDHLENTK ...GFNLKAL KFLIMDEADR ILNMDFEVEL
yhm5_yeast  KPHIIIATPG RLMDHLENTK ...GFSLRKL KFLVMDEADR LLDMEFGPVL
me31_drome  KVQLIIATPG RILDLMDKKV ....ADMSHC RILVLDEADK LLSLDFQGML
drs1_yeast  RPDIVIATPG RFIDHIRNSA ...SFNVDSV EILVMDEADR MLEEGFQDEL
if4a_rabit  APHIIVGTPG RVFDMLNRRY ....LSPKYI KMFVLDEADE MLSRGFKDQI
if41_human  APHIIVGTPG RVFDMLNRRY ....LSPKYI KMFVLDEADE MLSRGFKDQI
...         ...
```

Cons	A	B	C	D	E	F	G	H	I	K	L	M	N	P	Q	R	S	T	V
G	17	18	0	19	14	-22	31	0	-9	12	-15	-5	15	10	9	6	18	14	1
P	18	0	13	0	0	-12	13	0	8	-3	-3	-1	-2	23	2	-2	12	11	17
H	5	24	-12	29	25	-20	8	32	-9	9	-10	-9	22	7	30	10	0	4	-8
I	-1	-12	6	-13	-11	33	-12	-13	63	-11	40	29	-15	-9	-14	-15	-6	7	50
V	3	-11	1	-11	-9	22	-3	-11	46	-9	37	30	-13	-3	-9	-13	-6	6	50
V	5	-9	9	-9	-9	19	-1	-13	57	-9	35	26	-13	-2	-11	-13	-4	9	58
A	54	15	12	20	17	-24	44	-6	-4	-1	-11	-5	12	19	9	-13	21	19	9
T	40	20	20	20	20	-30	40	-10	20	20	-10	0	20	30	-10	-10	30	150	20
P	31	6	7	6	6	-41	19	11	-9	6	-16	-11	0	89	17	17	24	22	9
G	70	60	20	70	50	-60	150	-20	-30	-10	-50	-30	40	30	20	-30	60	40	20
...			...																

Figura 13: Allineamento multiplo e relativo profilo

Dal punto di vista del contenuto informativo un profilo può essere visto come una rappresentazione sintetica di tipo probabilistico di un allineamento. Il vantaggio che ne deriva da questa semplificazione spesso giustifica il rischio di avere perdita di informazioni importanti per caratterizzare le sequenze.

I profili sono degli importanti strumenti di ricerca e confronto: vengono usati soprattutto per cercare se certa proteina in esame ha delle sottosequenze che si

adattano al profilo e in caso positivo si possono dedurre le funzionalità di tali sottosequenze.

Nelle righe di un profilo possono essere presenti anche dei gap qualora in una certa posizione dell'allineamento il simbolo del gap sia il più presente. Profili estratti da pezzi di allineamenti senza gap sono detti più propriamente blocchi.

L'analisi per blocchi è concettualmente molto simile a quella con profili; anche essi servono per rappresentare le regioni conservate di un allineamento ma non ammettono la presenza di gap.

I blocchi possono essere calcolati a partire da un allineamento ma anche utilizzando dei match con pattern di lunghezza costante privi di gap. Esistono algoritmi per calcolare i blocchi conservati di un insieme di sequenze a partire dagli allineamenti a coppie senza calcolare l'allineamento multiplo. Tali blocchi possono essere usati come punto di partenza per costruire un allineamento multiplo attraverso l'aggiunta e l'arrangiamento dei gap nelle posizioni di unione fra i blocchi.

2.9.2 Pattern

Una tecnica di analisi molto utilizzata consiste nel raggruppare le proteine in famiglie in base di somiglianze strutturali, funzionali o evolutive. Questo consente di rappresentare le informazioni biologiche in maniera organica e ordinata semplificando notevolmente tutte le attività di ricerca dato che il numero stimato di famiglie proteiche, dell'ordine di 10^3, è enormemente inferiore rispetto al numero delle proteine (10^6) di cui si conosce la sequenza di aminoacidi. E' di fondamentale importanza perciò creare modelli adatti a rappresentare correttamente tutte le caratteristiche di tali famiglie.

Un pattern di sequenze biologiche non è altro che una stringa definita su un certo linguaggio che rappresenta una famiglia proteica descrivendo il tipo di aminoacidi presenti nelle posizione dei siti attivi. Si tratta di una rappresentazione simile ai profili ma molto più potente.

In questo paragrafo faremo riferimento ad una versione semplificata del linguaggio utilizzato nel database PROSITE [Bairoch98] che assomiglia molto al linguaggio delle "Regular Expression". Per esempio il pattern `P=AK[ILG]RR` rappresenta le

sequenze di una famiglia caratterizzate dal fatto che iniziano tutte con AK e terminano con RR e in mezzo presentano un aminoacido che può essere I, L o G. Si tratta perciò di una famiglia formata da tre sequenze: `AKIRR`, `AKLRR`, `AKERR`.

Alcuni simboli particolari servono per rappresentare sinteticamente pattern complessi come per esempio il simbolo $x(i_1,i_2)$ che rappresenta una stringa qualsiasi di lunghezza compresa fra i_1 e i_2; il simbolo $x(i) = x(i,i)$ che rappresenta una stringa qualsiasi di lunghezza i.

Vediamo per esempio un pattern ricavato dalla famiglia zinc finger c2h2 (241 proteine), e due sequenza che si adattano a tale pattern:

Pattern	**C**	**x(2,4)**	**C**	**x(3)**	**[ILVFYC]**	**x(8)**	**H**	**x(3,5)**	**H**
SeqA	C	AVFC	C	AKL	Y	QDSREWQQ	H	AS	H
SeqB	C	AVF	C	AKK	I	QDSREWQQ	H	ASF	H

Come per gli allineamenti è possibile definire il concetto di significatività. Un pattern è significativo quando la probabilità di trovarlo nelle sequenze è bassa; in questo caso infatti le zone messe in evidenza dal pattern non si sono conservate per puro caso ma per qualche importante motivo biologico.

I problemi tipici che vengono risolti in termini di pattern sono i problemi di classificazione e i problemi di conservazione. In un problema di classificazione è dato un set di sequenze S_+ che di sicuro appartiene ad una certa famiglia F e un set S_- che di sicuro non appartiene a F. Si vuole trovare una funzione di classificazione f che classifichi correttamente le sequenze di S_+ e S_- e classifichi correttamente con buona probabilità anche le sequenze di F. In formule:

$$f(s) = \text{true per } s \in S_+ \text{ e } f(s) = \text{false per } s \in S_-$$
$$\text{con buona probabilità: } f(s) = \text{true per } s \in F \text{ e } f(s) = \text{false per } s \notin F$$

dove chiaramente va specificato rigorosamente il significato di “buona probabilità”; per esempio si può imporre che f classifichi correttamente le sequenze nell’80% dei casi.

I problemi di conservazione assumono la stessa forma con la differenza che non si ha a disposizione di un set di esempi negativi S_-: può essere considerato come caso particolare di classificazione con S_- vuoto.

La funzione f di classificazione o di conservazione è rappresentata da un pattern P modo che *f(s)* sia true quando *s* combacia con *P*. Una definizione del problema in questi termini può essere molto rigida sia per il fatto che i dati possono presentare degli errori sia perché la descrizione in termini di pattern è di per se rigida (matching esatto). Per questo motivo sono stati definiti anche dei pattern con matching flessibile.

Una tipologia di pattern molto efficace per descrivere relazioni nascoste fra sequenze di famiglie proteiche sono i "pattern relazionali", che rispetto ai pattern appena visti presentano in aggiunta uno o più vincoli sugli elementi di tipo indefinito. La descrizione rigorosa di un pattern relazionale è data da una coppia $P = (\pi, R)$ dove π è un pattern semplice mentre R è un insieme di relazioni che vincolano gli elementi di tipo indefinito. Per esempio:

$$P = \left(G[FY]_1 \, G\, H\, x(1,3)[AEK]_2 \, , \{1,2 : \{FK, YA, YE\}\}\right)$$

questa notazione indica un vincolo sugli elementi indefiniti indicati con il pedice 1 e 2 che impone alla coppia `[FY][AEK]` di assumere soltanto alcuni determinati valori che sono quelli indicati nell'insieme `{FK,YA,YE}`. Per cui nel caso in cui `[FY]=F` avremo `[AEK]=K`, mentre nel caso `[FY]=Y` avremo `[AEK]=A` oppure `[AEK]=E`.

I linguaggi descrittivi tradizionali non si prestano alla descrizione di relazioni di questo tipo nel senso che, pur permettendo una descrizione esplicita, non prevedono un simbolismo specifico che ne semplifichi la notazione.

La presenza di queste relazioni fra aminoacidi, documentata in diverse famiglie proteiche permette di fare delle ipotesi sulla struttura terziaria delle sequenze. Infatti se due siti anche distanti sono correlati, è lecito supporre che possano essere vicini anche spazialmente.

2.9.3 Modelli di Markov

I modelli di Markov, utilizzati a partire dal 1970 nel riconoscimento della lingua parlata, sono dei modelli statistici molto adatti alla soluzione di problemi di biologia molecolare ([Krogh98]), soprattutto nella descrizione di famiglie proteiche.

Si tratta di modelli che possono essere considerati una via di mezzo fra i profili e i pattern visti in precedenza. La loro importanza risiede nel fatto che trattano i gap in maniera sistematica. Un modello di Markov (hidden Markov model) può essere ottenuto aggiungendo alla normale descrizione di un pattern delle informazioni probabilistiche. Questa modifica rende il pattern molto più potente come strumento di analisi e lo fa avvicina di più al concetto di profilo.

Per definizione un hidden Markov model è un grafo (Figura 14) in cui la generica posizione di un allineamento multiplo può essere rappresentata da tre stati:

- "match" (quadrato): rappresenta la distribuzione di caratteri in una colonna
- "stato di inserimento" (rombo): rappresenta l'inserimento di caratteri fra due posizioni dell'allineamento
- "stato di cancellazione" (cerchio): rappresenta una cancellazione (gap) in una posizione dell'allineamento.

Ad ogni ramo è associata la probabilità di transizione tra uno stato e l'altro ricavata dall'allineamento. Attraversare il grafo dal punto start al punto end passando per gli stati equivale a scorrere l'allineamento dalla prima colonna all'ultima. A seconda dello stato in cui mi trovo durante il percorso posso avere nella posizione corrente dell'allineamento un match, un inserimento o una cancellazione.

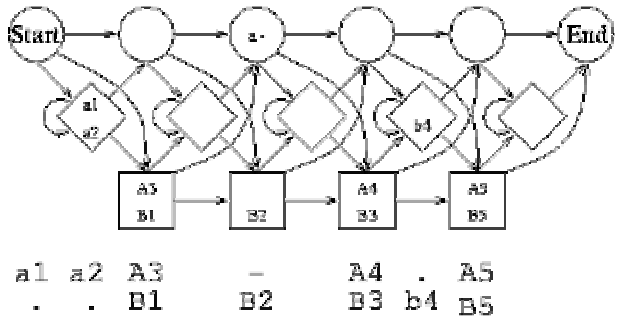


Figura 14: Hidden Markov Model

Un grafo di questo tipo non è altro che un "generatore casuale di sequenze allineate" con la statistica data dall'allineamento da cui è stata generato. Partendo dal punto Start si segue un percorso guidato dalla statistica fino ad arrivare a End e l'ordine degli stati per cui si è passati determina una sequenza con gap. La probabilità di avere in uscita una sequenza *s* è data dal prodotto delle probabilità dei rami del percorso che conduce ad avere s in uscita.

Consideriamo per esempio un allineamento di 5 sequenze di nucleotidi:

```
A C A - - - A T G
T C A A C T A T C
A C A C - - A G C
A G A - - - A T C
A C C G - - A T C
```

ad ogni colonna associamo le quattro probabilità di avere un certo simbolo. Per esempio nella prima posizione ho 4/5 di probabilità di avere A e 1/5 di avere T; nella quarta posizione ho 2/5 di probabilità di avere un gap di varia lunghezza. La traccia di tutte queste probabilità è mostrata nel diagramma in Figura 15.

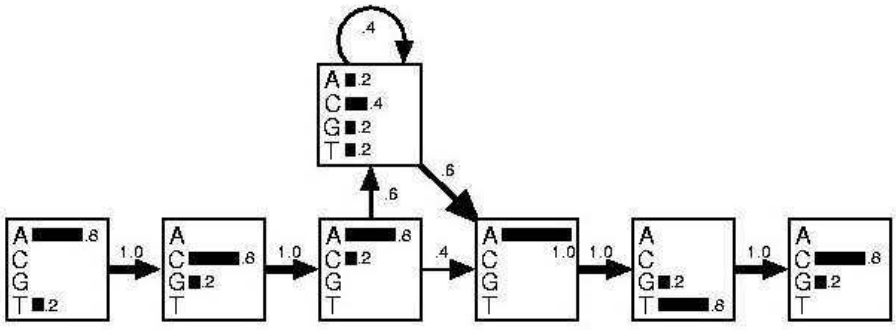


Figura 15: Esempio di modello di Markov

Nella Figura 16 sono mostrate le probabilità di tutte le sequenze originali, della sequenza consensus e della sequenza exceptional. Queste ultime rappresentano le sequenze rispettivamente più probabile e meno probabile che il modello può avere in uscita. In particolare il consensus è formato da tutti i caratteri che sono i più probabili in ogni posizione dell'allineamento per cui, analogamente al profilo, rappresenta in maniera schematica le caratteristiche essenziali della famiglia.

	Sequenze	**Prob. × 100**
Consensus	ACAC--ATC	4.7
Sequenze originali	ACA---ATG	3.3
	TCAACTATC	0.0075
	ACAC--AGC	1.2
	AGA---ATC	3.3
	ACCG--ATC	0.59
Exceptional	TGCT--AGG	0.0023

Figura 16: Statistica delle sequenze

Capitolo 3
Algoritmo di Allineamento per Astrazione

Uno dei limiti fondamentali dei metodi classici per l'allineamento di biosequenze consiste nel considerare esclusivamente relazioni di tipo "puntuale"; cioè il confronto fra sequenze discende direttamente da un confronto fra singole coppie aminoacidi in una certa posizione trascurando le eventuali relazioni fra aminoacidi in posizioni diverse. Infatti anche le euristiche di preallineamento basate su sottosequenze in realtà trattano queste sottosequenze come stringhe di caratteri indipendenti e perciò non tengono conto delle loro interazioni.

In altre parole i metodi classici sono basati quasi esclusivamente sulla conoscenza della struttura primaria delle sequenze senza tener conto delle informazioni derivanti dalla struttura secondaria e terziaria le quali sono un riflesso diretto dalle iterazioni fra aminoacidi sopra citate.

In questo capitolo viene presentato un nuovo modello di allineamento adatto a utilizzare tutte le informazioni di questo tipo. In realtà, a causa della scarsa disponibilità di informazioni generalizzabili sulla struttura terziaria e di database con strutture terziarie note, allo stato attuale è possibile sviluppare soltanto il livello secondario. Queste ultime infatti sono sempre reperibili, eventualmente tramite un'operazione di predizione.

Chiaramente la struttura primaria rimane sempre l'informazione più importante di tutta la biosequenza, sia perché è l'unica informazione conosciuta con precisione e completezza, sia perché tutte le altre informazioni possono essere in qualche modo derivate da essa. Per questo motivo il modello proposto si baserà essenzialmente su allineamenti di sequenze che verranno però guidati dalle informazioni addizionali.

3.1 Astrazione

In questo paragrafo viene introdotto il concetto dell'astrazione utilizzato in molti problemi di apprendimento [Saitta98], di theorem proving [Plaisted81] [Giunchiglia97] e di planning [Knoblock91].

Intuitivamente per astrazione si intende la mappatura di un problema da una rappresentazione concreta ad una nuova rappresentazione, detta astratta, in cui vengono tralasciati alcuni dettagli e conservate solo alcune proprietà desiderate allo scopo di generalizzare il problema e ridurne la complessità della ricerca.

Le teorie dell'astrazione possono essere rappresentate differentemente a seconda dell'ambiente in cui il problema viene definito e risolto. Due rappresentazioni molto usate sono "STRIP system" [Sacerdoti74] nei problemi di planning e "formal systems" [Plaisted81] nei problemi di theorem proving. Sebbene le varie rappresentazioni sono diverse, presentano almeno tre caratteristiche in comune che sono l'utilizzo di un linguaggio di descrizione del problema, un insieme di dati scritti in questo linguaggio e un insieme di regole da applicare ai dati.

Si farà riferimento alla prima in quanto è più adatta a rappresentare i problemi di ricerca e perciò anche quelli di allineamento.

Lo spazio dei problemi è definito da una coppia $\langle S, O \rangle$, dove *S* è l'insieme degli stati che descrivono l'ambiente e *O* è l'insieme degli operatori che permettono di passare da uno stato all'altro. Un problema è dato da una tripla $\langle P, s_0, G \rangle$ dove $P = \langle S, O \rangle$ è un spazio di problemi, $s_0 \in S$ è lo stato iniziale e $G \subset S$ è l'insieme degli stati obiettivo.

Il problema è risolto trovando una sequenza di operatori che partendo dallo stato iniziale portano ad uno stato obiettivo.

Anziché risolvere un dato problema $p = \langle P, s_0, G \rangle$ nello spazio in cui viene dato (ground space) si cerca prima una soluzione in uno spazio astratto in cui il problema sarà rappresentato nella forma $\pi^{(N)} = \left\langle \Pi^{(N)}, \sigma_0{}^{(N)}, \Gamma^{(N)} \right\rangle$.

Questo rappresenta il livello *N* di astrazione ovvero il livello più alto (spazio più astratto), dove *N* è un parametro che indica la profondità del processo di astrazione.

A partire dal livello *N* si ritorna al livello concreto (ground) passando per una successione di livelli intermedi di astrazione nel seguente modo:

1. al livello k=N, N−1, …,1 si risolve il problema $\pi^{(k)}$ trovando lo stato ottimo $\sigma_{opt}{}^{(k)}$;
2. si formula il problema al livello più concreto k−1 che si ottiene dal livello k aggiungendo gli operatori e gli stati che erano stati tralasciati nel processo di astrazione:
$$\pi^{(k)} \rightarrow \pi^{(k-1)} = \left\langle \Pi^{(k-1)}, \sigma_0{}^{(k-1)}, \Gamma^{(k-1)} \right\rangle$$
3. si utilizza $\sigma_{opt}{}^{(k)}$ per separare il problema al livello k−1 in due sottoproblemi: il primo ha parte dallo stato iniziale e ha come obiettivo lo stato $\sigma_{opt}{}^{(k-1)}$ ottenuto concretizzando $\sigma_{opt}{}^{(k)}$, mentre il secondo parte da $\sigma_{opt}{}^{(k-1)}$ e termina nello stato obiettivo $\Gamma^{(k-1)}$:
$$\pi_a{}^{(k-1)} = \left\langle \Pi^{(k-1)}, \sigma_0{}^{(k-1)}, \sigma_{opt}{}^{(k-1)} \right\rangle \qquad \pi_b{}^{(k-1)} = \left\langle \Pi^{(k-1)}, \sigma_{opt}{}^{(k-1)}, \Gamma^{(k-1)} \right\rangle$$
4. si passa ora a risolvere il problema k−1 composto eventualmente da diversi sottoproblemi (punto 1). L'algoritmo termina al livello ground k=0.

In Figura 1 è mostrato graficamente un problema di ricerca in cui il generico stato è rappresentato da un quadrato e lo spazio di ricerca è una griglia 8x8 di stati. Gli stati iniziali e obiettivo sono indicati rispettivamente con una croce e un cerchio. Il processo di ricerca è rappresentato da percorsi all'interno della griglia.

In Figura 2 è mostrato il processo di astrazione a due livelli dove ogni stato è ottenuto raggruppando quattro stati del livello precedente. Il livello 2 ha uno spazio di dimensione 2x2 e un solo stato obiettivo per cui la soluzione è banale. Il livello 1 ha uno spazio di dimensione 4x4 e tre stati obiettivo di cui due sono l'astrazione degli stati obiettivo concreti mentre il terzo è lo stato derivante dalla concretizzazione dello stato obiettivo al livello 2; quest'ultimo (tratteggiato) può essere considerato come uno stato di appoggio. In tal caso la soluzione è data dall'unione delle soluzioni dei due sottoproblemi: la prima va dallo stato iniziale allo stato d'appoggio mentre la seconda va da quest'ultimo allo stato obiettivo.

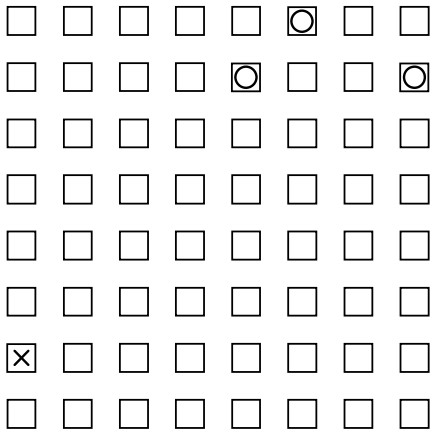


Figura 1: Rappresentazione di un problema di ricerca

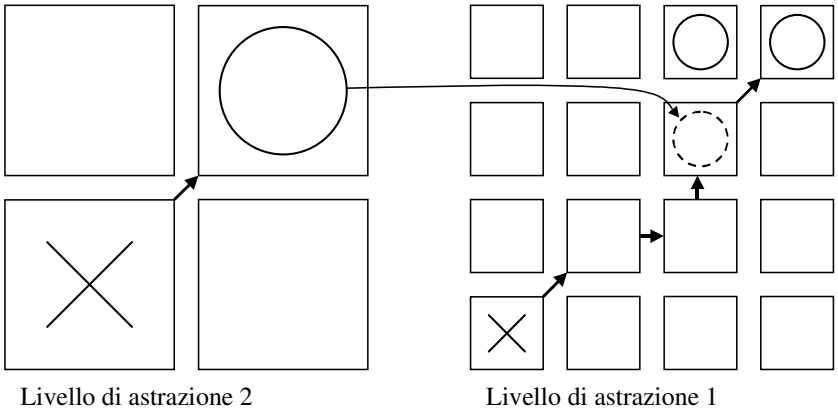


Figura 2: Ricerca nei livelli astratti

Con un procedimento analogo si passa dal livello concreto 0 dove la soluzione passa per due punti d'appoggio avendo suddiviso il problema in tre sottoproblemi (Figura 3).

L'esempio ha mostrato come l'astrazione ha semplificato la ricerca: la ricerca ad un livello astratto è più semplice in quanto lo spazio è ridotto, inoltre la soluzione a livello astratto guida e semplifica quella a livello concreto.

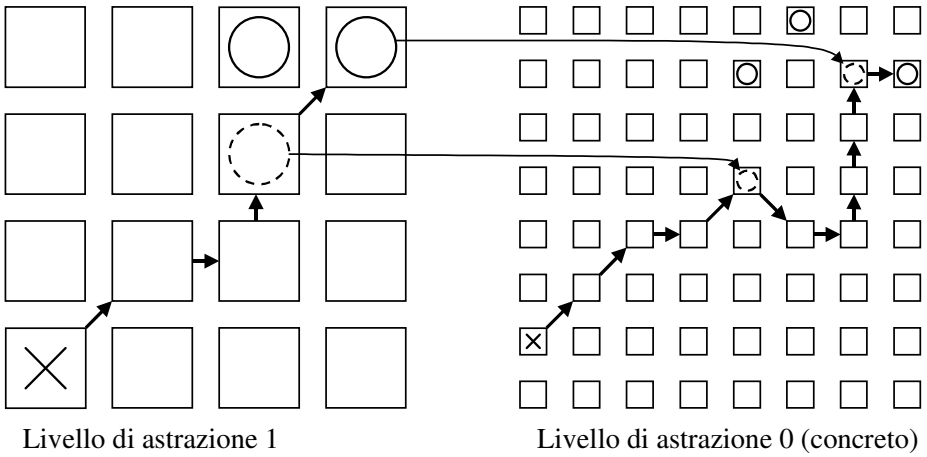


Figura 3: Ricerca nel livello concreto

La bontà di un algoritmo di astrazione dipende in gran parte dalle tecnica di mappatura e dal metodo di conservazione delle proprietà che, suggerite dalle caratteristiche particolari del problema in esame, rappresentano in maniera formale all'interno del modello l'euristica che si vuole utilizzare.

Per decidere quali sono proprietà le importanti e più rappresentative ad un certo livello di astrazione che è giusto conservare al livello successivo non si può fare a meno di considerare il problema particolare e il livello di dettaglio che si vuole ottenere. Un livello di dettaglio troppo elevato può, infatti, vanificare i vantaggi dell'astrazione mentre trascurando troppi dettagli posso ottenere livelli troppo generici.

3.2 Architettura a due livelli

Il modello proposto è sostanzialmente formato da due livelli (Figura 4) indipendenti e interagenti che lavorano a livelli di astrazione differenti: il livello della struttura primaria e il livello della struttura secondaria.

L'indipendenza è garantita dal fatto che ogni livello si occupa in maniera diretta di un solo tipo di struttura e utilizza solo le informazioni di sua competenza; risulta perciò molto importante l'interazione fra i due livelli che permette di sfruttare in maniera organica i due tipi di informazione.

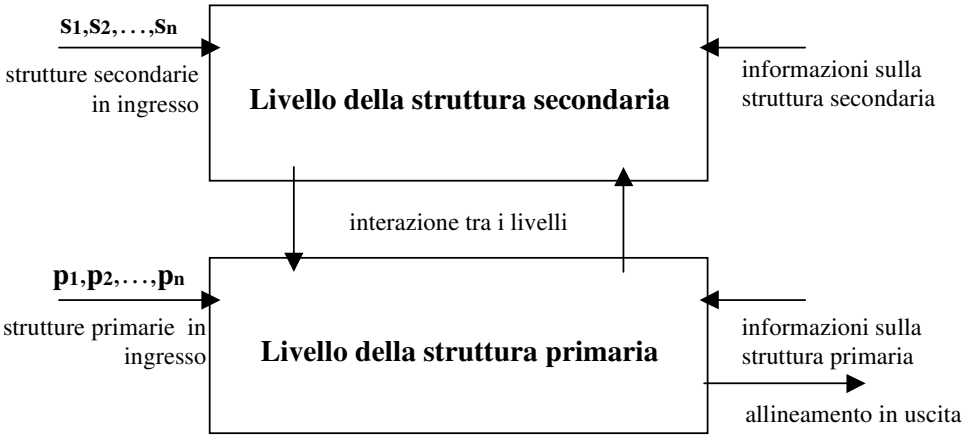


Figura 4: Modello a due livelli

Il livello principale del modello è quello primario in quanto deve trattare la parte dell'informazione più importante e deve implementare l'algoritmo di confronto fra le sequenze lavorando direttamente sui residui. Il livello secondario invece, lavorando ad un livello di astrazione maggiore, ha lo scopo di guidare il primario in modo da diminuire la complessità.

3.2.1 Livello della struttura primaria

Utilizzando la notazione già vista, le sequenze sono rappresentate da stringhe su un alfabeto *S* di venti simboli ognuno dei quali rappresenta un aminoacido:

$$\mathbf{s} = s_1 s_2 s_3 ... s_n \qquad s_i \in S$$

Il raggruppamento di residui può essere fatto secondo diversi criteri di classificazione (struttura secondaria, idrofobicità, idrofilicità, carica elettrica, ecc.) che portano a lavorare in spazi astratti.

Il livello della struttura primaria si occupa dell'allineamento vero e proprio fra sequenze di aminoacidi avendo come ingressi una lista di sequenze $\mathbf{P=p_1,p_2,\ldots,p_N}$ e come uscita le sequenze allineate.

Il modello non impone nessun vincolo riguardo il meccanismo di punteggio (score model) che può essere rappresentato nella forma più generale come una funzione $f(i_1,i_2,\ldots,i_N)$ che calcola il punteggio dell'allineamento ottimo fra le sottosequenze $\mathbf{p_1}[1\ldots i_1]$, $\mathbf{p_2}[1\ldots,i_2]$, …, $\mathbf{p_N}[1\ldots i_N]$ tenendo conto eventualmente delle relazioni

non puntuali. In generale la funzione *f* utilizzerà informazioni provenienti dai livelli superiori.

3.2.2 Livello della struttura secondaria

Il livello della struttura secondaria opera ad un livello più astratto in cui gli elementi fondamentali non sono gli aminoacidi ma sottosequenze etichettate con elementi di struttura secondaria.

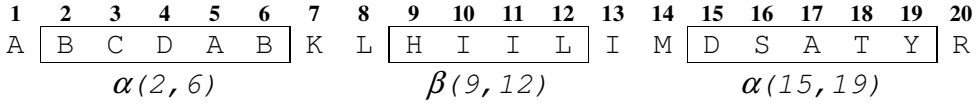
In tal caso una struttura secondaria $\boldsymbol{\sigma}$ può essere espressa come una stringa su un alfabeto Σ di simboli ognuno dei quali rappresenta un elemento di struttura secondaria:

$$\boldsymbol{\sigma} = \sigma_1\sigma_2\sigma_3...\sigma_m \qquad \sigma_i \in \Sigma$$

Considerando per semplicità un alfabeto che include gli elementi di struttura "α-helix", "β-sheet", "turn" un esempio di struttura secondaria è il seguente:

$$\boldsymbol{\sigma} = \alpha\beta\tau\beta \qquad \Sigma = \{\alpha, \beta, \tau\}$$

L'elemento di struttura secondaria non può essere considerato come un elemento isolato ma è rappresentativo di una certa sottosequenza della proteina per cui sarà caratterizzato da una posizione e una lunghezza relativi alla struttura primaria. Come mostrato nel seguente esempio, la notazione $\sigma_i(h,k)$ indica una struttura secondaria delimitata dalle posizioni h e k:



Notare come la struttura secondaria sia una mappatura in uno spazio più astratto della struttura primaria che consente di ridurre la complessità in quanto si opera con un alfabeto e con dimensioni ridotte.

Il modello accetta anche casi di incertezza utilizzando il simbolo particolare "c" (coil) per le zone di cui non si conosce con esattezza la struttura secondaria oppure per le zone che non assumono nessuna particolare struttura.

Il generico elemento $\sigma_i(h,k)$ è detto "residuo generalizzato per la struttura secondaria" o semplicemente "residuo generalizzato"; esso si differenzia dai

residui di aminoacidi per il fatto di avere una lunghezza. E' possibile anche utilizzare una notazione equivalente che etichetta tutti gli aminoacidi:

s =	A	B	C	D	A	K	L	H	I	I	B	L	M	S	R	D	F	D	S	A
	α	α	α	α	α	β	β	β	β	β	β	τ	τ			β	β	β	β	β

Le definizioni di inserimenti/cancellazioni si mantengono formalmente identiche alle analoghe operazioni fatte sulla struttura primaria. In questo caso il "gap generalizzato" sarà indicato con un simbolo "ε" che rappresenta l'inserimento di un elemento di struttura secondaria in una sequenza (o la cancellazione dello stesso elemento nell'altra sequenza). Ovviamente anche il gap generalizzato, come gli altri elementi di struttura secondaria, è rappresentativo di una porzione di sequenza primaria per cui sarà caratterizzato da una lunghezza.
A questo punto è possibile definire un "allineamento generalizzato" di due strutture secondarie come la coppia di strutture che si ottengono da quella di partenza aggiungendo eventualmente dei gap generalizzati e facendo in modo che le lunghezze siano uguali, dove per lunghezza si fa riferimento al numero di elementi della struttura secondaria e non al numero di aminoacidi coivolti.

Vediamo un esempio di strutture secondarie e di un possibile allineamento:

$$\begin{cases} \boldsymbol{\sigma_1} = \alpha\beta\tau\alpha\beta c\alpha\tau \\ \boldsymbol{\sigma_2} = \alpha\beta\alpha\tau\beta\tau \end{cases} \quad \Rightarrow \quad \text{all} = \begin{bmatrix} \alpha & \beta & \tau & \alpha & \beta & c & \alpha & \tau \\ \alpha & \beta & - & \alpha & \tau & - & \beta & \tau \end{bmatrix}$$

Un allineamento ottimo di questo tipo può essere calcolato con il solito metodo della programmazione dinamica usando un opportuno modello di punteggio relativo alle strutture secondarie.
In realtà come vedremo in seguito, le strutture secondarie non possono essere trattate come semplici strutture primarie per cui sarà necessario definire un meccanismo di punteggio più articolato, che non sarà necessariamente espresso in termini di matrice di sostituzione.

3.2.3 Interazione tra i livelli

I concetti di sostituzione, adattamento, inserimento e cancellazione di elementi di struttura secondaria sono simili dal punto di vista formale agli stessi concetti della struttura primaria, però assumono un significato diverso dal punto di vista biochimico.

Per esempio l'inserimento di un singolo aminoacido in una proteina spesso è una modifica che può avere un'importanza relativa in quanto può rappresentare una naturale evoluzione che non modifica sostanzialmente le proprietà della due sequenze.

In generale possiamo dire che il meccanismo di punteggio come somma di punteggi associati ai singoli aminoacidi è un meccanismo "debole" (poco selettivo) nel senso che per affermare che due sequenze abbiano funzioni simili devo richiedere che i punteggi di similarità siano molto elevati, ovvero che ci siano molti adattamenti. Consideriamo invece l'inserimento di una struttura secondaria in una proteina; questa operazione può modificare pesantemente la funzionalità della struttura stessa e non può essere vista come una conseguenza diretta dell'evoluzione in quanto è un'operazione che influisce su diversi aminoacidi contemporaneamente e sulle relazioni che intercorrono fra di essi. Possiamo dire allora che i confronti fra strutture secondarie sono dei confronti più "forti" (selettivi) rispetto a quelli relativi alle strutture primarie in quanto vengono confrontati proprio quegli elementi che caratterizzano le proteine dal punto di vista funzionale.

In altre parole una somiglianza/diversità nella struttura secondaria può essere un informazione molto più forte rispetto alla somiglianza/diversità nella struttura primaria. Nonostante questo l'informazione secondaria, essendo derivata da un procedimento di astrazione (semplificazione) può soltanto completare e arricchire l'informazione primaria ma non sostituirla.

In Figura 5 è mostrato un esempio di come può avvenire l'interazione fra i due livelli nell'allineamento di due sottosequenze. Il livello secondario ha allineato due strutture alfa di lunghezza differente e il primario si occuperà di allineare le due sottosequenze corrispondenti in base alla somiglianza delle coppie di aminoacidi.

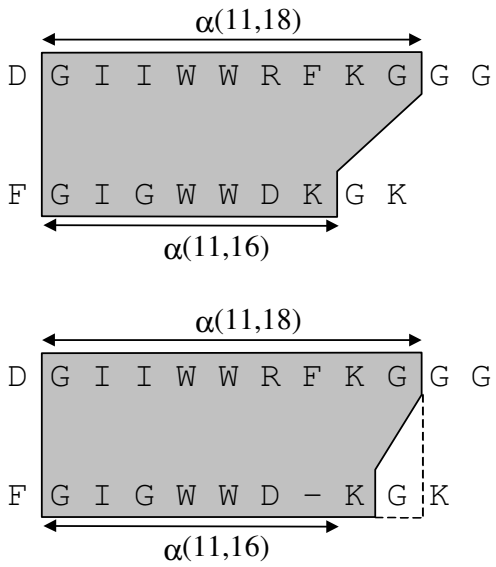


Figura 5: Interazione tra i due livelli

L'interazione avviene se per esempio il livello secondario tiene conto anche della lunghezza fra le due sottosequenze e del tipo di aminoacidi presenti. Infatti se le lunghezze sono molto differenti oppure se gli aminoacidi della prima sequenza sono presenti in minima parte nella seconda il punteggio sarà inferiore anche a livello primario.

Un altro aspetto molto importante è quello della flessibilità che viene gestita in termini di interazione fra i livelli. Eseguendo un confronto secondario troppo "rigoroso", infatti c'è il rischio di condurre ad un allineamento con elevata somiglianza secondaria ma con poco significato dal punto di vista del confronto primario. Il rischio di introdurre questi "falsi teoremi" va bilanciato con il vantaggio che si ottiene dall'utilizzo delle informazioni secondarie nell'algoritmo. La soluzione migliore sarà una via di mezzo fra rigore e flessibilità permettendo per esempio alle zone secondarie di allargarsi e restringersi a seconda della situazione.

Nell'esempio in Figura 5 l'allargamento del raggio d'azione rispetto al vincolo posto dal secondario porta dei benefici sia in termini di differenza di lunghezza che in termini di confronto primario. Questo equivale a consentire una certa flessibilità nelle sovrapposizioni fra gli elementi di struttura secondaria.

3.3 Allineamento a coppie

In questo paragrafo viene descritto l'algoritmo allineamento fra due sequenze che utilizza l'astrazione a livello secondario per guidare il confronto primario.

In Figura 6 è mostrata la matrice di allineamento "vincolata". Si tratta di una matrice in cui è definita soltanto una banda relativamente stretta di elementi che, vincolando l'allineamento a seguire un particolare percorso, permette un notevole risparmio computazionale e di memoria soprattutto per sequenze molto lunghe. La struttura della matrice è costruita a partire dai vincoli imposti dal livello astratto.

Consideriamo due sequenze proteiche etichettate con elementi di struttura secondaria:

$$\begin{cases} \mathbf{a} = a_1 a_2 a_3 ... a_n \\ \mathbf{A} = A_1 A_2 ... A_N \end{cases} \qquad \begin{cases} \mathbf{b} = b_1 b_2 b_3 ... b_m \\ \mathbf{B} = B_1 B_2 ... B_M \end{cases}$$

dove *n* e *m* sono il numero di residui primari e *N* e *M* il numero di residui secondari generalizzati.

Il primo passo consiste nell'effettuare un allineamento sulle sequenze secondarie **A** e **B**, usando l'algoritmo di programmazione dinamica.

Il meccanismo di punteggio deve tener conto dell'effettiva somiglianza puntuale fra elementi di struttura secondaria, e questo può essere fatto attraverso una matrice si sostituzione secondaria; il costo dei gap sarà dato da un contributo costante dovuto alla presenza e un contributo proporzionale alla lunghezza del gap stesso:

$$C = G_{opening} + \lambda \cdot G_{elong}$$

In Figura 7 è mostrato un esempio di modello di punteggio ricavato in base ai punteggi di allineamenti di strutture tridimensionali note (3D_Ali) che presentano una somiglianza strutturale di almeno il 70%.

In Figura 8 è mostrato un esempio di allineamento secondario con matrice di allineamento.

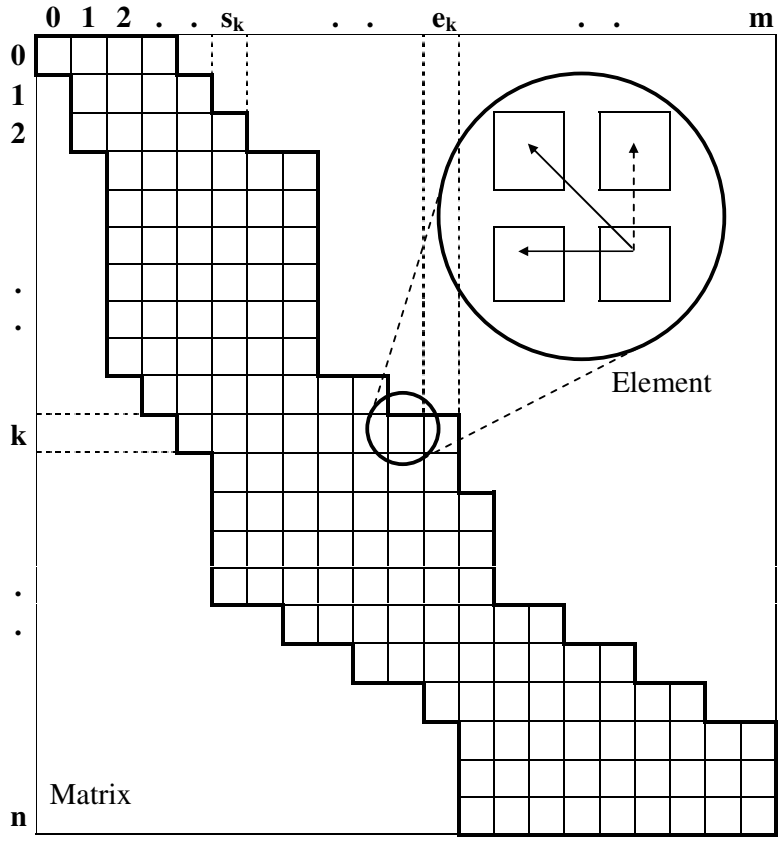


Figura 6: Esempio di matrice vincolata

	H	E	L
H	2		
E	−15	4	
L	−4	−4	−2

Go=−12 **Ge**=−2

H helix
E sheet
L loop (coil)

Figura 7: Esempio di score model secondario

Passiamo ora all'algoritmo di allineamento primario in cui il punto cruciale è la determinazione della struttura della matrice vincolata primaria in base alle informazioni derivanti dal confronto secondario.

Il primo passo consiste nell'espandere il percorso dell'allineamento secondario in modo da ottenere una matrice primaria vincolata a blocchi come mostrato in Figura 9. La struttura a blocchi introduce dei vincoli troppo forti sull'allineamento primario per cui è necessario smussare gli angoli con un'operazione di filtraggio che elimina o aggiunge alcuni elementi. In Figura 10 è mostrata la struttura della matrice dell'allineamento dopo lo smussamento.

Il meccanismo di punteggio primario non prevede ulteriori vincoli sulla matrice ma si basa esclusivamente sui confronti fra aminoacidi consentiti dalla matrice stessa. L'utilizzo di una matrice di sostituzione classica come PAM o BLOSUM è la scelta più semplice e spesso porta a dei risultati abbastanza accurati. In Figura 11 è mostrata il percorso lungo la matrice corrispondente all'allineamento primario.

$$\begin{cases} \mathbf{a} = \text{ARQEFGFDACKLKGGFDASR} \\ \mathbf{A} = \beta(1,3)\,\alpha(4,6)\,\beta(7,12)\,\alpha(13,18)\,c(19,20) \end{cases}$$

$$\begin{cases} \mathbf{b} = \text{ARARDAREQFGCLKLKGSRA} \\ \mathbf{B} = \alpha(1,5)\beta(6,8)\alpha(9,11)\beta(12,17)c(18,20) \end{cases}$$

		α	β	α	β	c
	0	−1	−3	−4	−5	−6
β	−1	**2**	1	−1	−2	−3
α	−3	**3**	2	1	0	−1
β	−4	−1	**4**	4	−1	−2
α	−5	1	3	**5**	**2**	4
c	−6	−4	3	7	6	**10**

Allineamento

β	α	β	α	–	c
–	α	β	α	β	c

Figura 8: Esempio di allineamento secondario con matrice piena

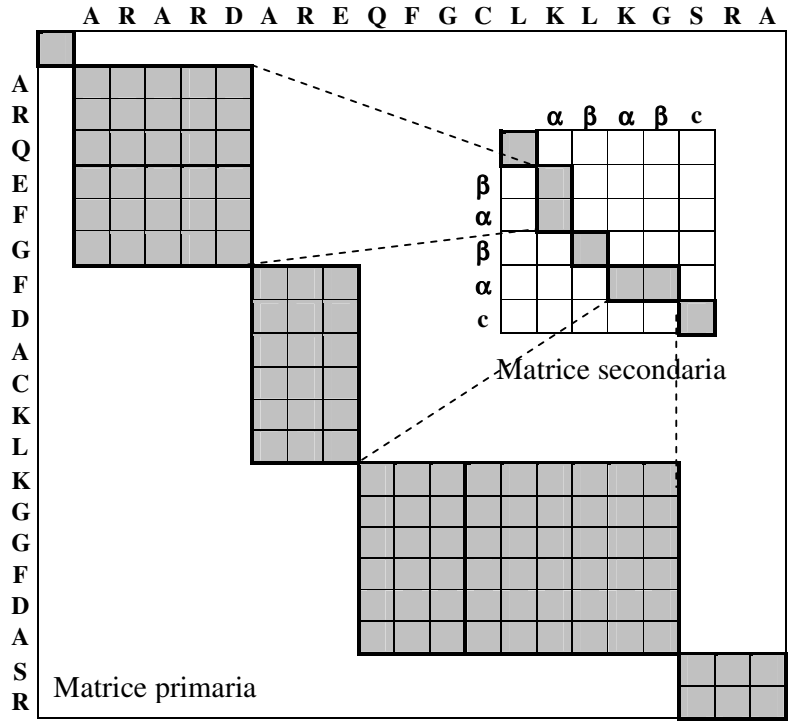


Figura 9: Espansione della matrice secondaria

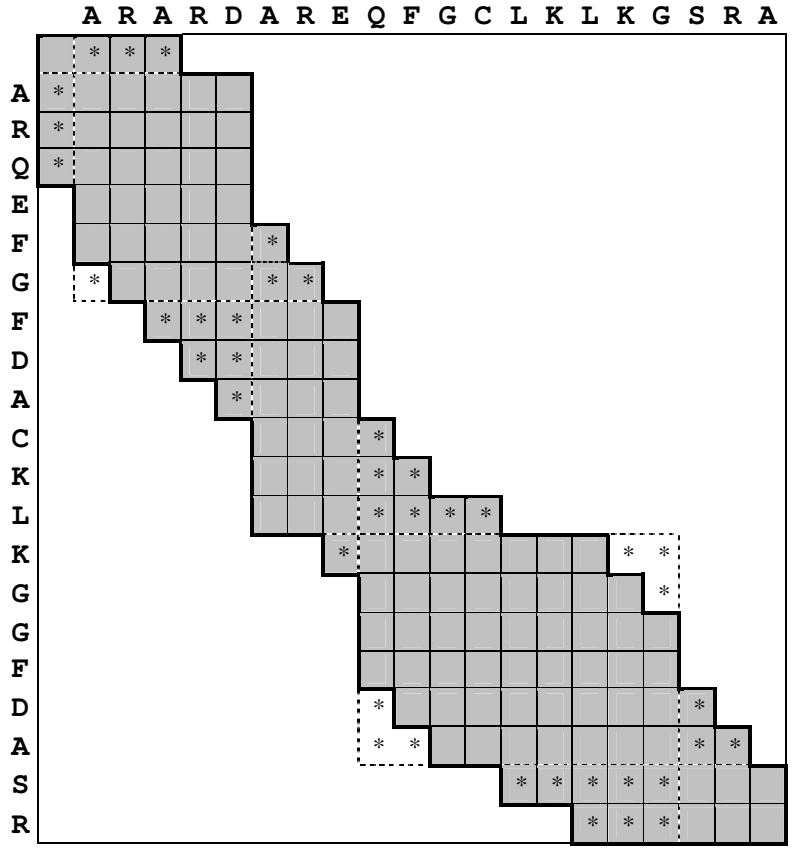


Figura 10: Struttura della matrice primaria dopo lo smussamento; gli elementi aggiunti o cancellati rispetto alla struttura iniziale sono indicati con un asterisco

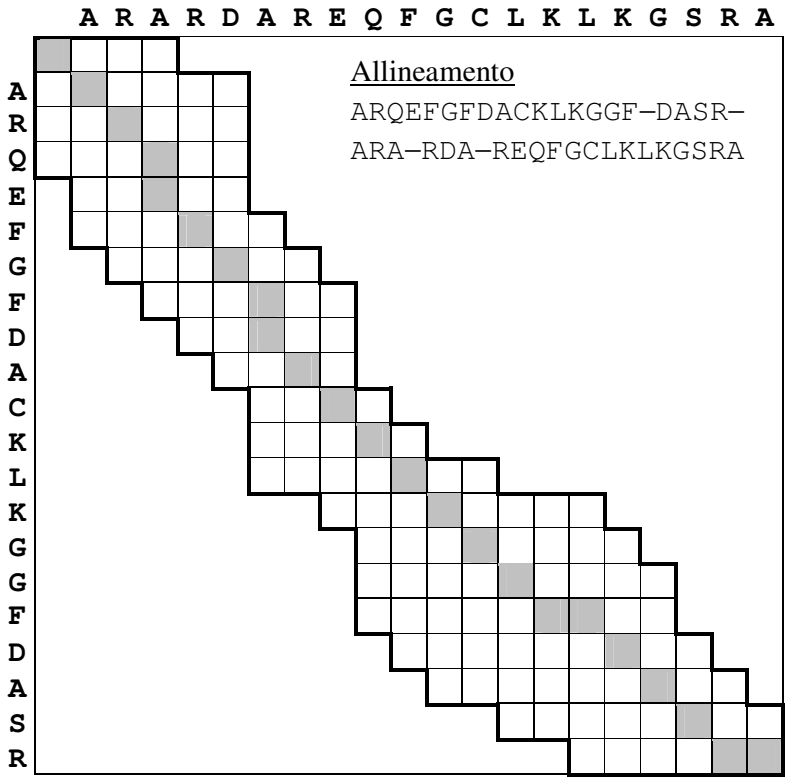


Figura 11: Allineamento primario

3.4 Allineamento multiplo

Dal punto di vista concettuale l'algoritmo di allineamento a coppie visto in precedenza può essere generalizzato al caso di allineamenti multipli anche se la complessità computazionale rende il problema intrattabile.

Come abbiamo visto in precedenza le tecniche progressive (ClustalW), che costruiscono l'allineamento unendo mano a mano gruppi sempre più grandi di sequenze, sono limitate da una consistente propagazione degli errori ma hanno un basso costo computazionale essendo basate sostanzialmente su confronti a coppie; viceversa le tecniche iterative (SAGA), permettendo il feedback, danno risultati più accurati ma hanno una complessità computazionale maggiore.

Con l'algoritmo di allineamento multiplo proposto si rinuncia alla ricerca della soluzione ottima ma si utilizzano una serie di processi progressivi e iterativi per costruire un allineamento multiplo subottimo. I processi progressivi hanno lo scopo di posizionare velocemente l'allineamento in un punto possibilmente vicino alla

soluzione ottima, mentre quelli iterativi effettuano una ricerca locale modificando piccole sottosequenze.

In Figura 12 è mostrato uno schema molto generale dell'algoritmo di allineamento dove è evidenziato un ciclo di feedback interno per il processo iterativo, e un ciclo più esterno per il processo progressivo.

In generale la distinzione fra processo progressivo e iterativo è trasversale rispetto alla suddivisione in livello primario e secondario nel senso che entrambi i livelli possono dare il loro contributo utilizzando entrambe le tecniche. Però in realtà, risulta naturale assegnare al livello secondario soltanto i processi progressivi in quanto questo livello definisce una linea guida per il livello primario, mentre quest'ultimo si occupa dell'allineamento vero e proprio.

In questo modo si ha una separazione più netta fra le competenze dei due livelli e viene evidenziata maggiormente l'astrazione secondaria come processo a sé stante. Questo non pregiudica l'interazione fra i livelli che è garantita nel modello di punteggio.

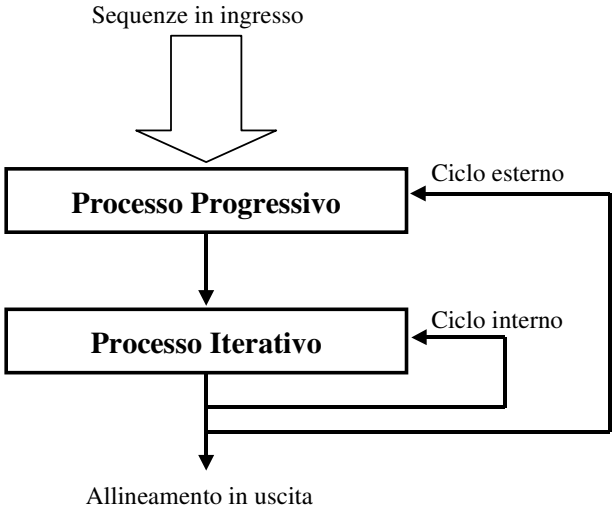


Figura 12: Schema generale dell'algoritmo di allineamento multiplo

In Figura 13 viene mostrato l'algoritmo con i due livelli separati. Il blocco di espansione è l'interfaccia fra i due livelli e serve per creare un allineamento primario a partire da quello secondario attraverso un espansione dei residui generalizzati.

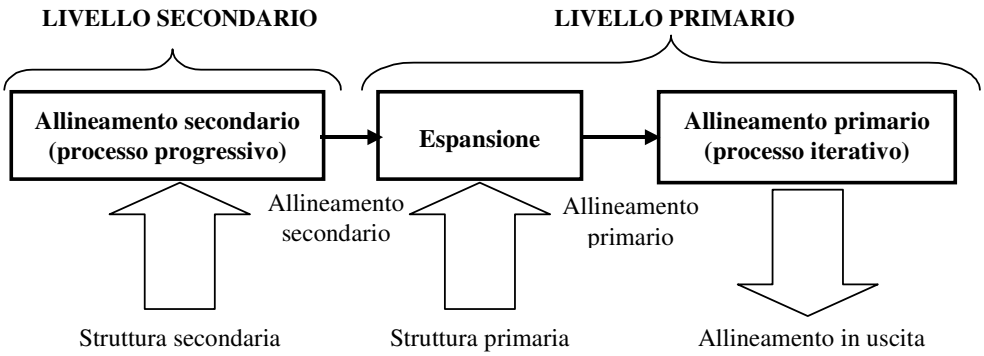
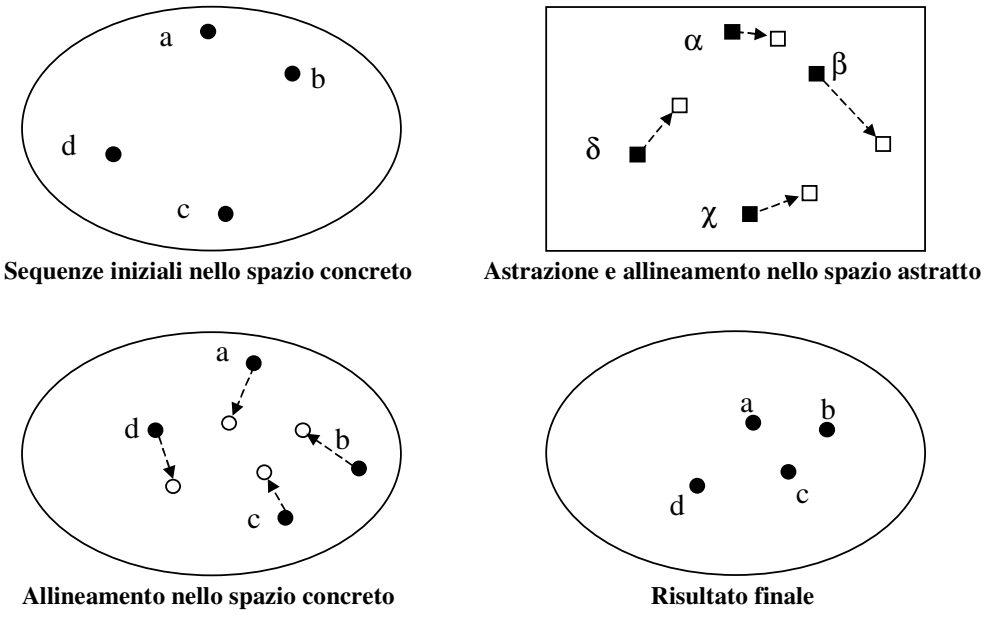


Figura 13: Schema dell'algoritmo di allineamento

In Figura 14 è mostrato graficamente l'algoritmo: le sequenze dell'allineamento, indicate come punti di uno spazio, sono rappresentate nello spazio concreto e nello spazio astratto mentre il processo di allineamento è indicato con spostamenti che servono per assegnare le corrette relazioni esistenti fra le sequenze. Infatti ricordiamo che un'allineamenento fra due sequenze può essere interpretato in termini di distanza/somiglianza.



Sequenze iniziali nello spazio concreto **Astrazione e allineamento nello spazio astratto**

Allineamento nello spazio concreto **Risultato finale**

Figura 14: Rappresentazione grafica dell'algoritmo

3.4.1 Allineamento secondario

Il primo passo dell'algoritmo consiste nel generare un allineamento multiplo secondario utilizzando una tecnica progressiva in cui si parte da un allineamento a coppie e si aggiungono mano a mano le altre sequenze allineandole con l'allineamento multiplo attuale. L'algoritmo progressivo presentato segue la strategia dell'albero guida evolutivo simile a quella utilizzata da ClustalW per definire l'ordinamento con il quale verranno aggiunte le sequenze.

A tale scopo vengono introdotti i concetti di sequenza e allineamento generalizzati: una sequenza generalizzata **s** non è altro che un allineamento multiplo che viene visto come un array i cui elementi s_i sono le colonne dell'allineamento stesso. Indicando con N il numero di sequenze dell'allineamento e con L la sua lunghezza, avrò L colonne e ogni colonna s_k sarà una sequenza di lunghezza N:

$$\mathbf{s} = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_L \\ \hline s_{11} & s_{21} & \cdots & s_{L1} \\ s_{12} & s_{22} & \cdots & s_{L2} \\ \cdots & \cdots & \cdots & \cdots \\ s_{1N} & s_{2N} & \cdots & s_{LN} \end{bmatrix}$$

dove il generico s_{ij} può essere un aminoacido o un gap.

Data una matrice di sostituzione M il punteggio fra due elementi (colonne) di sequenze generalizzate è definito in maniera analoga al punteggio negli allineamenti multipli come somma dei punteggi associati a tutte le coppie. Per i gap avrò un modello analogo a quelli già visti.

Il meccanismo di punteggio in realtà è più complicato nel caso di confronto secondario in quanto tiene conto delle differenze di lunghezze degli elementi come abbiamo già visto nei paragrafi precedenti.

La relazione utilizzata è la seguente:

$$\text{score}(A_i, B_j) = M(A_i, B_j) - \lambda \cdot \frac{\left|\text{length}(A_i) - \text{length}(B_j)\right|}{\text{length}(A_i) + \text{length}(B_j)}$$

dove A_i e B_j sono elementi di struttura secondaria.

Applicando l'algoritmo di allineamento (per esempio la programmazione dinamica) alle fra due sequenze generalizzate con le solite regole formali e utilizzando il confronto visto in precedenza si ottiene un allineamento generalizzato ovvero un "allineamento fra allineamenti".

In Figura 15 è mostrato l'allineamento fra le sequenze generalizzate secondarie:

A =	c	–	α	c	α	β	c	
	c	β	α	c	α	–	–	

B =	α	β	–	–	α	c	η	c	β
	α	β	c	β	α	β	η	c	–

Notare che i gap aggiunti dall'allineamento, evidenziati in Figura 15, vanno inseriti in tutte le sequenze che compongono la sequenza generalizzata.

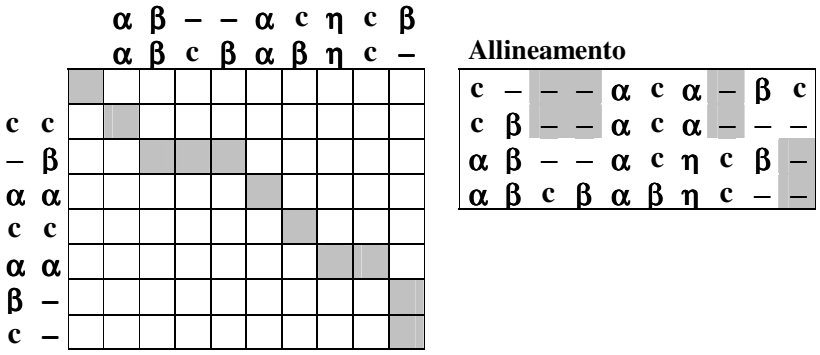


Figura 15: Allineamento fra sequenze generalizzate secondarie

Vediamo ora come si può costruire un allineamento multiplo applicando ripetutamente l'algoritmo di allineamento generalizzato visto in precedenza ad un set di sequenze iniziali $\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3,\ldots,\mathbf{p}_N$.

La tecnica generale consiste nell'allineare alcuni gruppi disgiunti di queste sequenze in modo da ottenere un set ridotto di sequenze generalizzate; applicando tante volte questo procedimento si ottiene alla fine una sola sequenza generalizzata che rappresenta l'allineamento multiplo finale. Questo procedimento può essere rappresentato graficamente con un albero in cui le foglie sono le sequenze iniziali, ogni nodo rappresenta un allineamento multiplo ottenuto allineando le sequenze figlie, e il nodo radice rappresenta il risultato finale. Un esempio è mostrato in Figura 16.

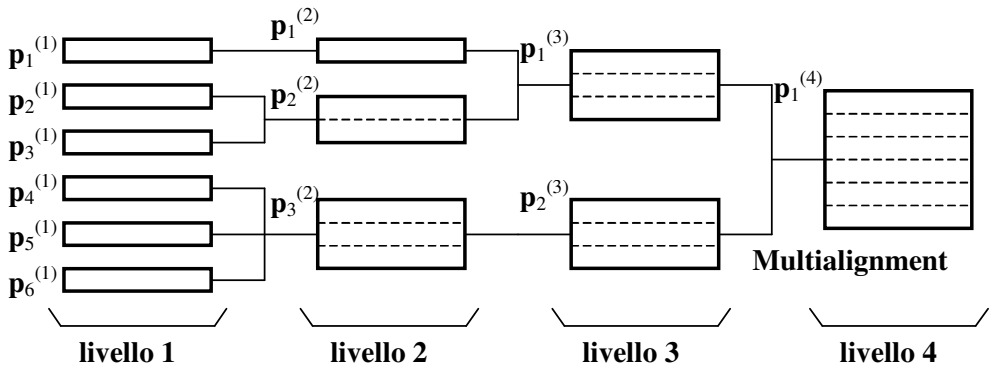


Figura 16: Costruzione dell'allineamento multiplo secondario

Chiaramente la bontà dell'allineamento è determinata in gran parte dal criterio usato per raggruppare le sequenze da allineare che in generale sarà basato sulla somiglianza, cioè si allineano gruppi di sequenze simili. E' possibile definire diversi criteri di raggruppamento a seconda delle esigenze, ma per avere una visione più completa conviene fare una classificazione generale.

Possiamo innanzitutto suddividere i criteri in due grandi classi:

- Criteri statici: la struttura dell'albero è definita subito al primo passo a partire da un confronto diretto fra le sequenze primarie iniziali $\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3,\ldots,\mathbf{p}_N$. Gli allineamenti sono effettuati seguendo staticamente tale struttura fino ad arrivare al risultato finale.
- Criteri dinamici: la struttura dell'albero è definita passo per passo, cioè al generico passo k vengono decisi i raggruppamenti del livello k da un confronto fra $\mathbf{p}_1^{(k)},\mathbf{p}_2^{(k)},\mathbf{p}_3^{(k)},\ldots,\mathbf{p}_{Nk}^{(k)}$, e poi si eseguono i relativi allineamenti che generano le sequenze in ingresso al livello k+1.

Il primo metodo è decisamente più efficiente computazionalmente ma può portare a risultati peggiori.

Una seconda classificazione riguarda il numero di raggruppamenti nel passare da un livello al livello successivo:

- Criteri di raggruppamento singolo: si accetta un solo raggruppamento per livello
- Criteri di raggruppamento multiplo: è possibile avere più di un raggruppamento.

I criteri del primo tipo effettuano un confronto più preciso ma possono portare ad alberi di notevole dimensione. Per definire i singoli raggruppamenti viene eseguito un confronto fra tutte le sequenze in gioco ad un certo passo in modo da avere un punteggio s_{ij} associato alla coppia di sequenze i e j. Una coppia di sequenze diventa un gruppo se ha un punteggio s_{ij} superiore ad una certa soglia s_{th}.
Vediamo ora l'algoritmo di allineamento multiplo passo per passo applicato alle sequenze in ingresso $\mathbf{p}_1^{(1)},\mathbf{p}_2^{(1)},\mathbf{p}_3^{(1)},\ldots,\mathbf{p}_{Nk}^{(1)}$:

1	k=1
2	Si effettua un confronto a coppie fra tutte le sequenze $\mathbf{p}_1^{(k)},\mathbf{p}_2^{(k)},\mathbf{p}_3^{(k)},\ldots,\mathbf{p}_{Nk}^{(k)}$ in modo da calcolare i punteggi $s_{ij}^{(k)}$.
3	Si effettuano gli allineamenti fra le coppie di sequenze che hanno un punteggio $s_{ij}^{(k)}$ superiore alla soglia s_{th}. Si ottiene così un nuovo insieme di sequenze generalizzate che rappresenta l'uscita al livello k e l'ingresso al livello k+1: $\mathbf{p}_1^{(k+1)},\mathbf{p}_2^{(k+1)},\ldots,\mathbf{p}_{N(k+1)}^{(k+1)}$ dove $N_{k+1}<N_k$
4	k=k+1
5	Se $N_k=1$ l'algoritmo termina dando con uscita $\mathbf{p}^{(k)}$, altrimenti si riparte dal punto 2.

Per concludere osserviamo che la scelta di un criterio di allineamento molto accurato è indispensabile qualora un algoritmo progressivo di questo tipo venga utilizzato per generare un allineamento multiplo definitivo. Nel nostro caso non è richiesta grande precisione perché, come vedremo in seguito, l'allineamento in uscita del blocco secondario verrà sottoposto ad una serie di operatori che possono modificarlo notevolmente. L'utilizzo di un criterio di raggruppamento statico e singolo è più che sufficiente per i nostri scopi e consente di avere, oltre un risparmio computazionale notevole, un buon posizionamento per le modifiche successive.

3.4.2 Espansione

L'espansione dell'allineamento multiplo secondario appena ottenuto fornisce l'ingresso per il livello primario. L'espansione consiste nel sostituire ad ogni elemento secondario la relativa sottosequenza primaria e riempire con gap gli spazi che necessariamente rimangono liberi. Consideriamo per esempio un elemento secondario A_i e indichiamo con l_{max} la lunghezza massima degli elementi che si trovano nella stessa colonna di A_i. Se A_i è un gap devo sostituirlo con l_{max} gap nel livello primario, altrimenti lo sostituisco con la sottosequenza primaria corrispondente subseq(A_i) seguita da un numero di gap pari a l_{max}–length(A_i).

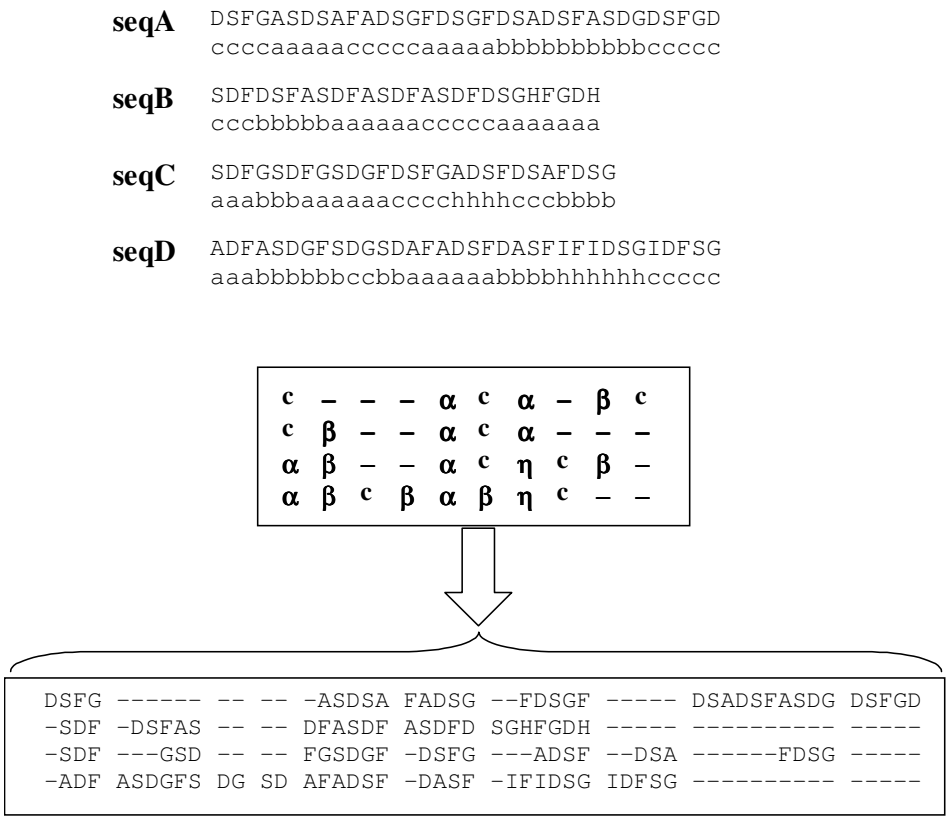


Figura 17: Espansione dell'allineamento secondario

Osserviamo che l'allineamento multiplo ottenuto per espansione può presentare un punteggio a livello primario molto basso. Questo non è un problema a questo punto dell'algoritmo in quanto la cosa importante è avere un allineamento ben posizionato cioè tale che le zone che devono sovrapporsi si trovino in posizioni

vicine. Sarà compito del livello primario allineare correttamente queste zone con operatori locali.

In Figura 17 è mostrato un esempio di allineamento secondario e la relativa espansione dove le sottosequenze appartenenti a strutture secondarie adiacenti sono state separate da spazi.

3.4.3 Allineamento primario

Vediamo ora il livello primario che, come abbiamo visto, è composto da un ciclo principale progressivo e un ciclo interno iterativo (Figura 18). Il processo progressivo si occupa di selezionare le sequenze da modificare mentre il processo iterativo esegue delle modifiche locali.

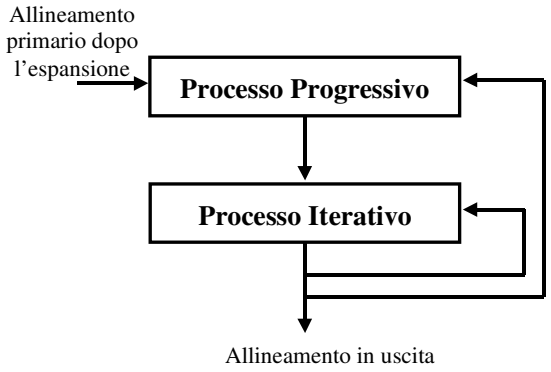


Figura 18: Schema dell'allineamento primario

Lo scopo principale è quello di ottimizzare una funzione obiettivo che indica la qualità dell'allineamento, ovvero la funzione di punteggio del tipo:

$$\text{Score}(s_1, s_2, ..., s_N) = C_{go} \cdot N_{go} + C_{gc} \cdot N_{gc} + \sum_{s_i, s_j \neq \text{gap}} M(s_i, s_j)$$

dove M è una matrice di sostituzione, N_{go} e N_{gc} sono rispettivamente il numero di gap presenti e il numero complessivo di gap symbol e C_{go}, C_{gc} i rispettivi costi. In questo schema non vengono considerati i gap presenti all'inizio o alla fine dell'allineamento in quanto non rappresentano inserimenti veri e propri.

Descriviamo ora in dettaglio i due processi.

Il **processo progressivo** si occupa di selezionare una o più sequenze dell'allineamento che saranno soggette alle modifiche iterative e di attivare le sequenze che entreranno in gioco nel calcolo del punteggio. Ad ogni passo infatti non tutte le sequenze danno il loro contributo nel punteggio ma solo quelle che vengono attivate, consentendo di ridurre il rischio, tipico degli algoritmi progressivi, di cadere in massimi locali.

In Figura 19 è mostrato un allineamento con un algoritmo progressivo molto semplice ma che porta spesso a buoni risultati. Vengono attivate inizialmente solo due sequenze per esempio le prime due, e si applica un processo iterativo di allineamento prima a s_1 e poi a s_2. Poi si attiva anche s_3 e si applica su di essa il processo iterativo e così via per le altre sequenze. L'allineamento viene costruito allineando una ad una tutte le sequenze mantenendo attivate solo le sequenze precedentemente allineate.

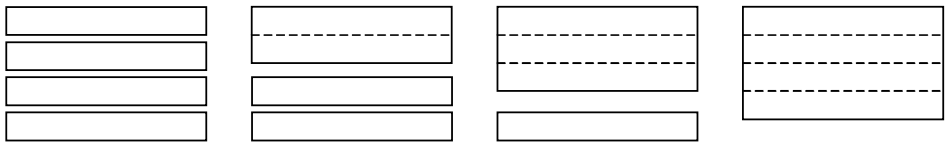


Figura 19: Processo progressivo per l'allineamento multiplo primario

Il **processo iterativo** si occupa di applicare degli operatori locali di modifica ad una singola sequenza in maniera da massimizzare la funzione Score sulle sequenze attive. Formalmente il generico operatore è una funzione che agisce su una o più sottostringhe appartenenti ad una sequenza **s** dell'allineamento:

$$O = O\left(\mathrm{sub}_1(\mathbf{s}), \mathrm{sub}_2(\mathbf{s}), ..., \mathrm{sub}_L(\mathbf{s})\right)$$

dove sub(**s**) rappresenta una certa sottosequenza di **s** e L è il numero di sottosequenze che può modificare.

Il parametro *L* rappresenta l'estensione del raggio d'azione dell'operatore e perciò è un indice della sua complessità ma anche della sua efficacia nell'allineamento.

Per come è stato definito il modello il livello primario deve eseguire soltanto operazioni di tipo locale in quanto si suppone che l'allineamento proveniente dal livello secondario sia già posizionato bene. Di conseguenza sono stati utilizzati

soltanto operatori a corto raggio in modo da avere una complessità computazionale relativamente bassa.

Vediamo innanzitutto due operatori semplici che effettuano una modifica elementare su una singola sottosequenza: l'operatore di movimento e l'operatore di arrangiamento dei gap.

L'operatore di movimento (Figura 20) sposta la sottosequenza in tutto il suo raggio d'azione fino a trovare la posizione con punteggio massimo. Un movimento verso sinistra può avvenire soltanto se a sinistra della sottosequenza ho dei gap e analogamente per la parte destra. Questo tipo di operatore può applicarsi soltanto alle sottosequenze che si presentano come "isole" circondate da gap.

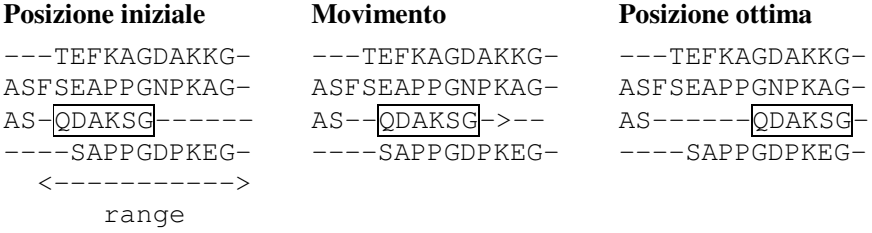


Figura 20: operatore di movimento

L'operatore di arrangiamento tenta di arrangiare i gap della sottosequenza spostandoli in posizioni differenti cercando di ottimizzare il punteggio. Per motivi di efficienza la ricerca non è completa ma si limita alle configurazioni più probabili in cui i gap symbol non spezzano troppo la sottosequenza in esame.

E' un operatore più complesso rispetto a quello di movimento e può portare a risultati migliori soprattutto nell'inserimento di gap. Nel caso in cui la sottosequenza è lunga o con molti gap può diventare complesso computazionalmente, per cui è utile suddividere la sottosequenza in più parti da trattare separatamente. In Figura 21 è mostrato un esempio.

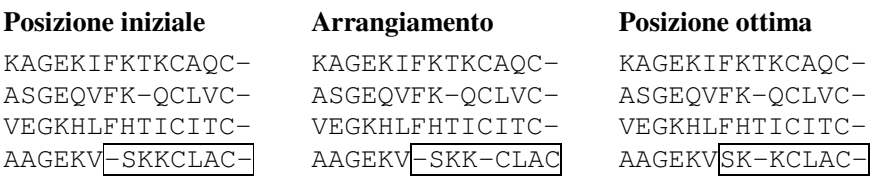


Figura 21: Operatore di arrangiamento dei gap

I due operatori appena mostrati possono essere applicati contemporaneamente in maniera mista: mentre viene fatta muovere la sottosequenza, ad ogni passo si applica l'operatore di arrangiamento alla stessa e si memorizza il risultato migliore trovato.

In Figura 22 è mostrato un esempio di applicazione di questo operatore dove è evidenziato l'allineamento ottimo trovato (arrangiamento 3). Notare che ad ogni movimento viene memorizzata la posizione iniziale in modo che dopo l'arrangiamento è possibile tornare indietro per effettuare il movimento successivo.

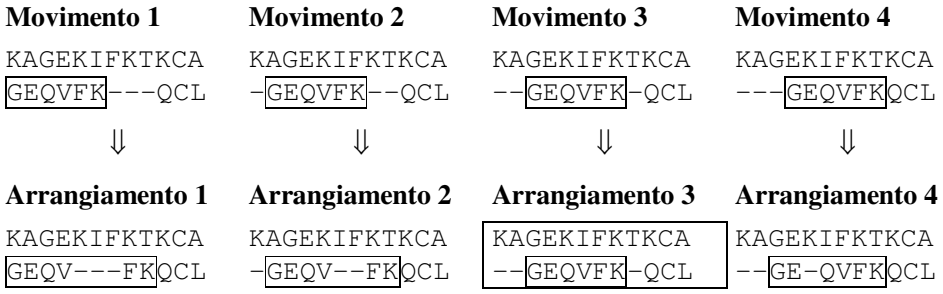


Figura 22: Esempio di operatore misto

3.4.4 Risultati sperimentali

L'algoritmo di allineamento descritto in questo capitolo è stato implementato nel programma A3 "Alignment Algorithm with Abstraction".

Per valutare le performance del programma rispetto agli algoritmi disponibili in letteratura è stato scelto di utilizzare il dataset *BAliBASE* [Thompson99b]. Si tratta di un database composto da 142 allineamenti elaborati manualmente e che pertanto possono essere considerati corretti. Il dataset è diviso in 5 gruppi, ogni dei quali ha lo scopo di testare una particolare caratteristica del programma in esame (Figura 23): la capacità di allineare sequenze equidistanti (gruppo1), capacità di trattare sequenze in cui è presente un elemento "intruso" (gruppo2), raggruppamento delle sequenze in famiglie (gruppo3) e infine sequenze con presentano gap estesi (gruppo4-5).

Le strutture secondarie sono state estratte dal database PDB [Berman00] e in loro assenza sono state predette con il programma SSPRO [Pollastri02]. Per la misura della qualità di ogni allineamento è stato utilizzato il parametro *CS* (column score)

che indica la percentuale di colonne di un'allineamento che risultano corrette rispetto all'allineamento di riferimento di *BAliBASE*.

Gruppo	Contenuto
gruppo 1	Allineamenti di sequenze equidistanti che variano per lunghezza e similarità delle sequenze.
gruppo 2	Insiemi di sequenze strettamente correlate con l'aggiunta di una sequenza orfana che tende a corrompere l'allineamento.
gruppo 3	Sequenze prese da un numero limitato di famiglie differenti.
gruppo 4	Allineamenti con gap molto lunghi nelle zone terminali delle sequenze.
gruppo 5	Allineamenti con gap molto lunghi nelle zope interne alle sequenze.

Figura 23: Descrizione del dataset *BAliBASE*

I risultati sono stati comparati con i programmi ClustalW [Thompson94], Praline [Heringa99], DiAlign2 [Morgenstern96], TCoffee [Notredame00]. In particolare ClustalW e DiAlign2 sono stati inclusi in quanto rappresentano una tipica implementazione di algoritmi progressivi e iterativi rispettivamente. Praline invece è stato incluso per la sua capacità di utilizzare informazioni sulle strutture secondarie e da questo punto di vista è simile all'algoritmo proposto.

Il primo esperimento (Figura 24) mostra il comportamento di A3 su tutto il dataset *BAliBASE* rispetto agli altri programmi. In tutti i gruppi considerati A3 presenta dei risultati paragonabili con lo stato dell'arte e in certi casi superiori (gruppi 1,2).

Nel primo gruppo 1 le prestazioni di A3 sono dovute al fatto che l'informazione sulla struttura secondaria permette di selezionare con accuratezza le zone ad elevata somiglianza funzionale anche se queste presentano una similarità bassa a livello primario. Questo aspetto verrà analizzato meglio negli esperimenti successivi. Anche nel gruppo 2 A3 risulta essere il programma con prestazioni migliori. Si tratta di un gruppo di allineamenti rappresentativo di una situazione sperimentale molto frequente: è stato commesso un errore nella selezione delle sequenze della famiglia proteica per cui è presente una proteina "intrusa" che destabilizza l'allineamento. La proteina presenta una somiglianza primaria elevata con le altre e inoltre potrebbe avere delle caratteristiche comuni a certe sequenze che però non

sono caratteristiche della famiglia in esame. Questo fenomeno confonde la maggior parte dei programmi che raggiungono appena il 40%. Anche se con A3 si ottengono risultati migliori, in quanto il preallineamento a livello secondario permette di filtrare l'effetto di disturbo derivante dall'aggiunta della sequenza, il tasso di errore è ancora molto elevato. Allo stato attuale nemmeno algoritmi basati su accurati modelli di Markov risecono a trattare questo problema.

Il gruppo 3 contiene sequenze raggruppate per famiglie e sotto certi punti di vista presenta gli stessi problemi del gruppo 2 anche se molto meno accentuati. In questo caso i risultati si aggirano intorno al 50% di precisione per Praline, TCoffee e A3.

Programma	***1***	***2***	***3***	***4***	***5***
ClustalW	78.59	39.61	47.73	51.83	61.00
DiAlign2	70.79	31.83	35.45	**76.67**	83.33
Praline	77.41	37.70	**50.00**	27.92	59.42
T-Coffee	78.85	40.35	48.64	69.25	**89.33**
A3	**81.42**	**43.23**	49.45	76.18	88.92

Figura 24: Risultati dei programmi sui 5 gruppi di *BAliBASE*

Gli ultimi due gruppi rappresentano dei tipici casi in cui le sequenze hanno lunghezze molto differenti per cui la costruzione dell'allineamento implica l'aggiunta di gap molto lunghi. In questo caso il confronto fra le strutture secondarie effettuato da A3 permette di posizionare correttamente dei gap anche molto lunghi nel primo step dell'allineamento per cui gli operatori locali possono ottimizzare l'allineamento senza il rischio di spostarlo oltre i vincoli imposti dal livello secondario.

L'esperimento seguente permette di valutare le presetazioni di A3 nel caso in cui le sequenze in esame presentino delle una similarità primaria molto bassa. E' stato stimato infatti che esiste una soglia del 30% di percentuale di identità al di sotto della quale il calcolo di un'allineamento multiplo risulta particolarmente complesso a causa dell'elevato numero e valore dei massimi locali presenti nelle funzioni di punteggio [Rost99]. Mediando i risultati dei programmi proposti in esame sugli allineamenti di *BAliBASE* che soddisfano questo criterio si ottengono i risultati riportati in Figura 25.

Programma	*CS*
ClustalW	53.62
DiAlign2	48.44
Praline	47.50
T-Coffee	60.01
A3	**69.12**

Figura 25: Risultati per gli allineamenti con percentuale di identità <30%

In questo caso è evidente come i deboli segnali di omologia delle sequenze possano essere trattati con una maggiore efficacia utilizzando l'informazione strutturale delle proteine. Spesso infatti una somiglianza a livello strutturale e funzionale non si ripercuote a livello primario per cui l'utilizzo dell'informazione secondaria può venire incontro per la soluzione di questo problema.

Infine in figura Figura 26 è mostrato un esempio evidente in cui in un'allineamento fra sequenze con similarità minore del 15%, l'algoritmo A3 riesce meglio di TCoffee a allineare zone che presentano una struttura secondaria ben evidente.

A3

```
..NLFVALYDfvasgdntlsitkGEKLRVLgynhn.............gEWCEAQt..kngqGWVPSNYITPVN......
kGVIY.ALWDyepqnddelpmkeGDCMTIIhrede.............deiEWWWARl..ndkeGYVPRNLLGLYP......
.gYQYRALYDykkereedidlhlGDILTVNkgslvalgfsdgqearpeeiGWLNGYnettgerGDFPGTYVEYIGrkkisp
...NFRVYYRDsrd.....pvwkGPAKLLWkg.................eGAVVIQd..nsdiKVVPRRKAKIIRd.....
...drvrkksga........awqGQIVGWYctnlt.............peGYAVESeahpgsvQIYPVAALERIN......
```

TCoffee

```
.NLFVALYDfvasgdntlsitkGEKLRV...................Lgynhng EWCEA.........Qtkngq GWVPSNYIT
kGVIYALWDyepqnddelpmkeGDCMTIIhrededeiEWWWARlndkeGYVP..................RNLLGLYP.....
gYQYRALYDykkereedidlhlGDILTV.............Nkgslvalgfsdgqearpeei GWLNGYnettgerGDFPGTYVE
..NFRVYY.....RDsrdpvwkGP..........AKLLW.......kgeGAVVI.............Qdnsd...........iK
..........drvrkksgaawqGQIVGW.............Yctnltpe GYAVESeahp.............gsvQIYPVAALE
```

Figura 26: Esempi di allineamenti fra sequenze con identità del 15%

Seconda Parte

Predizione di Strutture Secondarie Proteiche

Capitolo 4
Predizione di strutture proteiche

In questo capitolo verrà introdotto il problema della predizione di strutture proteiche e dello stato dell'arte sulle tecniche di apprendimento automatico per la predizione di strutture secondarie. La predizione e la modellazione di strutture proteiche risulta di fondamentale importanza in bioinformatica a causa dello stretto legame che esiste fra la struttura di una proteina e la sua funzione.

4.1 Introduzione

A causa dello stretto legame che esiste fra la funzione di una proteina all'interno di un organismo e la sua struttura, la predizione di strutture proteiche è diventato uno dei task più imporntanti della proteomica e della bioinformatica. Sebbene esistano accurate tecniche sperimentali come la Risonanza Nucleare Magnetica e la Cristallografia a raggi X, queste sono ancora troppo dispendiose in termini economici e di tempo per applicazioni in larga scala. D'altra parte nonostante l'enorme crescita del numero di dati disponibile nei database di strutture proteiche, il gap esistente fra le sequenze note (160,000 in Swiss-Prot [Bairoch00]) e le strutture note (29,000 in PDB [Berman00]) è ancora elevato e in costante crescita come mostrato nelle in Figura 1 e Figura 2. Nonostante i progressi negli ultimi anni in questa direzione non sono ancora state trovate metodologie di applicazione generale. Per questo motivo, la necessità di tecniche di predizione automatica e di modellazione di strutture proteiche è di fondamentale importanza per la ricerca nell'ambito della bioinformatica.

La predizione di strutture terziare, ovvero del folding completo della proteina, è un problema intrattabile per cui molti studi si sono concentrati nel compito più semplice della predizione di strutture secondarie. La struttura secondaria

rappresenta il folding locale della proteina ed è per questo meno informativo. D'altra parte molti modelli di predizione/modellazione terziaria e di studio funzionale prendono come punto di partenza proprio la struttura secondaria. Non è raro infatti che siti funzionali e domini proteici siano fortemente caratterizzati da un punto di vista secondario per cui possono essere analizzati senza dover ricorrere allo studio del folding completo.

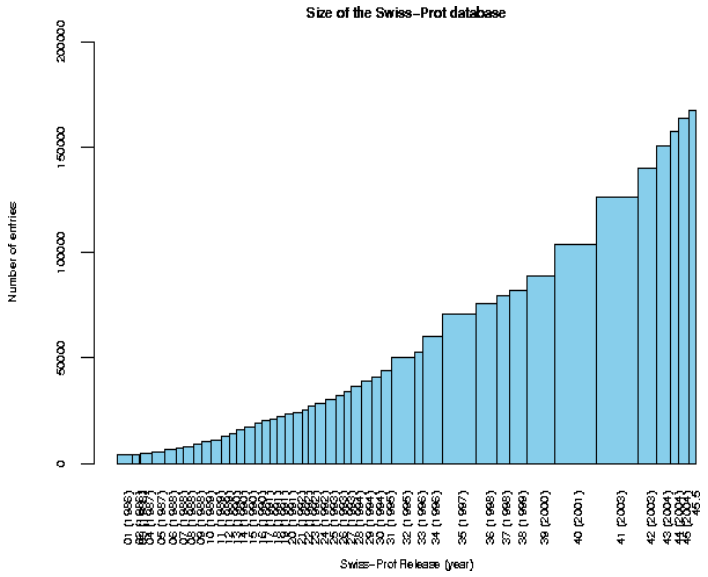


Figura 1: Crescita negli anni del numero di entry in SwissProt

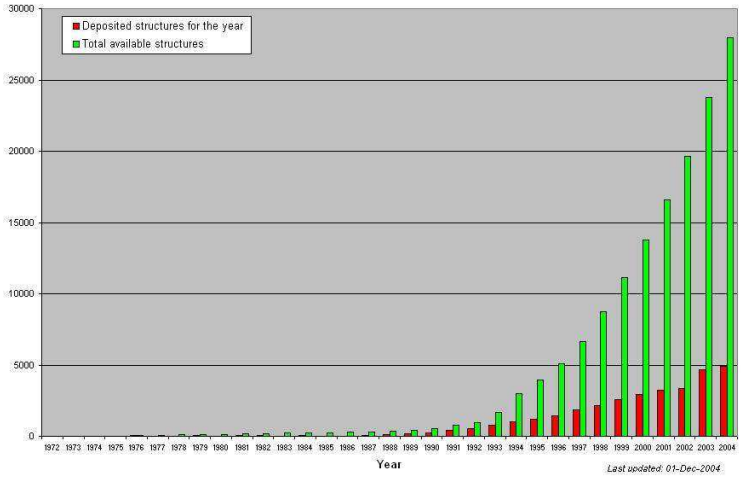


Figura 2: Crescita negli anni del numero di entry in PDB

4.2 Stato dell'arte

In questo paragrafo verranno illustrati i vari metodi presenti nello stato dell'arte della predizione di strutture proteiche facendo riferimento soprattutto alle tecniche di predizione di strutture secondarie.

Il problema della determinazione delle strutture proteiche è molto complesso in quanto sono coinvolti processi e interazioni biologiche, chimiche e fisiche. In maniera molto schematica è possibile classificare le metodologie di predizione in tre rami principali: i modelli comparativi, i metodi ab-initio, e i metodi di fold-recognition.

I *modelli comparativi* sono basati sull'ipotesi di base che proteine con sequenze simili avranno simile anche la struttura e perciò la funzione [Chothia86]. La predizione può essere effettuata tramite confronti fra la sequenza primaria della proteina in esame (target) con proteine a struttura nota (template). Sebbene i modelli comparativi siano in assoluto i più accurati, la loro applicabilità è limitata ai casi in cui la similarità è molto elevata (>30%) e in cui è possibile costruire un preciso allineamento fra target e template. Una procedura tipica per la costruzione di un modello comparativo è la seguente [Sali95] [Sanchez97]:

1. Identificazione delle proteine a struttura nota che sono in qualche modo correlate con la proteina in esame
2. Allineamento fra le proteine trovate e la proteina in esame e costruzione di un template per ogni allineamento che presenta uno score sufficientemente alto.
3. Costruzione di un modello 3D della proteina target basato sugli allineamenti con i template.

Diversi algoritmi sono stati studiati per costruire il modello corrispondente ad un allineamento. In particolare ricordiamo i metodi basati sull'assemblaggio delle sottosequenze [Greer90], il matching di segmenti [Unger89] e l'utilizzo di vincoli spaziali [Havel93].

I *metodi ab-initio* invece si basano sul principio che la struttura delle proteine all'equilibrio termodinamico corrisponde al suo stato di minima energia libera. A partire da un modello di energia si può applicare un adeguato algoritmo di ottimizzazione per trovare la configurazione ottimale corrispondente alla struttura nativa.

Un tipico algoritmo ab-initio è composto dai seguenti step:

1. costruzione di un modello biologico che rappresenta le interazioni fra le differenti parti delle proteine
2. definizione di una funzione che rappresenta l'energia libera
3. applicazione di un algoritmo di ottimizzazione per trovare la configurazione ottimale a minima energia che corrisponde alla struttura predetta

Accurati modelli a livello atomico [Boczko96] sono applicabili soltanto a sequenze corte perché molto complessi computazionalmente per cui spesso sono più utili modelli semplificati che lavorano a livello di residui [Levitt76] o a livello di griglie (lattice-model) [Skolnick90]. Per quanto riguarda le funzioni di energia adottate in letteratura ricordiamo le funzioni basate su potenziali atomici [Roterman89], su potenziali statistici [Vajda97], e su regole chimiche [Hinds94]. Infine, per quanto riguarda le strategie di ottimizzazione, sono stati proposti algoritmi genetici [Dandekar94], simulazioni Monte Carlo [Covell92], e simulazione di dinamiche molecolari [Levitt83].

I metodi di *fold recognition* partono dall'osservazione che l'enorme numero di proteine presenti in natura codificano in realtà soltanto per un piccolo numero di tipologie strutturali [Chothia92], [Blundell93], inoltre la probabilità di trovare nel database PDB una struttura simile ad una proteina nuova è circa il 70% [Orengo94] anche se alla somiglianza strutturale non è associata una similarità primaria.

Anche in questo caso la struttura della proteina viene predetta attraverso confronti con le sequenze di un database con la differenza che vengono utilizzate delle metriche basate sulla compatibilità della sequenza con un certo tipo di folding e non sulla similarità fra le sequenze. Ai template strutturali che ne derivano viene poi associato uno score e viene poi scelto quello con score maggiore.

Anche per i metodi di fold-recognition sono state proposte diverse strategie in letteratura. Nel fold recognition 3D-1D [Bowie91] le strutture terziarie del database sono codificate con delle stringhe come la struttura secondaria e l'accessibilità al solvente. Queste stringhe mono-dimensionali vengono poi allineate con l'analoga stringa della sequenza terget.

Le difficoltà nella predizione delle strutture proteiche sono dovute soprattutto alle complesse relazioni fra le differenti parti della proteina fra loro e le altre molecole

dell'ambiente circostante. Molti tipi di struttura sono determinati in gran parte da interazioni locali fra residui vicini mentre altre sono dovute a interazioni distanti nella stessa proteina. Infatti, nonostante sia riconosciuto che la sequenza primaria contiene tutte le informazioni necessarie per determinare la struttura proteica [Anfinsen73], recenti studi hanno mostrato che molti proteine assumono la loro struttura nativa con l'aiuto di molecole esterne (chaperones) [Gething92], [Hartl94].

Lo studio della strutture proteiche può essere semplificato considerando soltanto la struttura secondaria cioè le conformazione lineare a cui appartiene ogni residuo della proteina. La struttura secondaria può essere vista come un'etichettatura in cui ad ogni aminoacido è associata uno stato strutturale: alfa elica (α), foglietto beta (β), random coil (c), dove quest'ultima rappresenta lo stato in cui non è presente nessun folding riconoscibile. Le strutture secondarie α e β sono schematizzate in Figura 3.

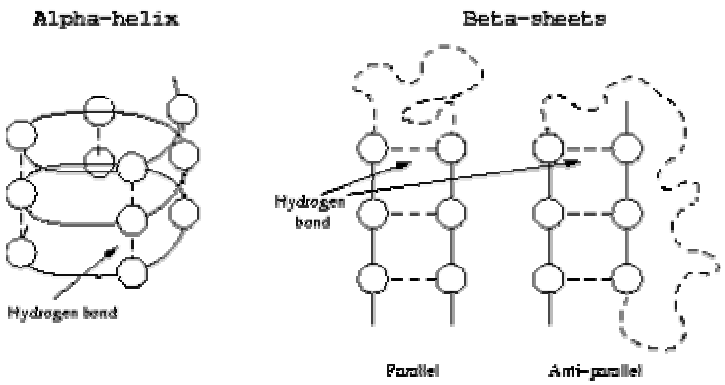


Figura 3: Strutture secondarie (α e β)

Sebbene la struttura secondaria può sembrare molto meno informativa rispetto alla terziaria, in realtà gioca un ruolo importante nella stabilizzazione della struttura terziaria, nella cinetica del folding e nella determinazione di domini funzionali e di siti attivi. Pertanto la struttura terziaria e la funzione proteica in molti casi può essere determinata a partire dalla secondaria.

Nel prossimo paragrafo verrà descritto lo stato dell'arte riguardo agli algoritmi di predizione di strutture secondarie.

4.3 Predizione di strutture secondarie

Esiste una grande varietà di metodi proposti in letteratura che vengono suddivisi per semplicità di tre generazioni. I metodi di prima generazione risalenti ai primi anni 70 sono basate sullo studio statistico della composizione delle strutture proteiche in termini di aminoacidi. Calcolata la propensione degli aminoacidi ad appartenere ad una certa struttura secondaria è possibile definire delle regole empiriche di predizione [Chou74] [Garnier78] e [Lim74]. Lo svantaggio principale di queste tecniche è che non tengono conto delle informazioni contestuali e cioè delle relazioni fra aminoacidi vicini lungo la sequenza, mentre è ormai noto che le strutture secondarie sono fortemente dipendenti da questi tipi di legami.

I metodi di seconda generazione migliorano enormemente le performance sfruttando database proteici e informazioni statistiche su sotto-sequenze di aminoacidi. In questo caso la predizione può essere vista come un problema di machine learning nel quale viene creato un modello che predice la sequenza della struttura secondaria (labeling). Le regole derivanti, applicate a un aminoacido centrato in una finestra di dimensione 11-21, restituiscono la propensione del residuo centrale ad appartenere alla classe strutturale alfa, beta e coil. Diversi metodi rientrano in questa categoria e possono essere classificati in base all'approccio sottostante, il tipo di informazione utilizzata e la tecnica di apprendimento. Per quanto riguarda l'approccio ricordiamo le tecniche basate sull'informazione statistica [Robson76] [Garnier87], la teoria dei grafi [Mitchell92], la statistica multivariata [Kanehisa88]; il tipo di informazione è relativo alle proprietà della proteina da utilizzare per estrarre le feature dell'apprendimento, fra queste ricordiamo le proprietà chimico-fisiche [Ptitsyn83], e i pattern di sequenza [Taylor83]. Per quanto riguarda le tecniche di apprendimento, le più diffuse sono quelle basate su reti neurali [Holley89] [Qian88], anche se bisogna ricordare altri approcci come il k-Nearest Neighbors [Levin86] [Lander93].

Probabilmente l'innovazione più significativa introdotta nei metodi di predizione di terza generazione riguarda l'utilizzo delle informazioni evolutive e a lungo raggio contenute negli allineamenti multipli. E' ben noto il fatto che anche singole variazioni nella sequenza di aminoacidi possono compromettere la funzionalità della proteina per cui per evidenziare quali sostituzioni sono invece "accettate"

dall'evoluzione è possibile allineare fra loro sequenze appartenenti alla stessa famiglia oppure con similarità molto elevata. In questo modo è possibile costruire un profilo che indica per ogni posizione della sequenza in esame le propensioni dell'aminoacido a sostituirsi con gli altri preservando la funzionalità proteica. L'allineamento multiplo è perciò un'informazione aggiuntiva che indica le regioni maggiormente conservate della sequenza e per questo permette di localizzare le zone che con maggiore probabilità avranno un particolare folding. Per quanto riguarda gli aspetti le tecniche utilizzate ricordiamo i metodi basati su statistiche puntuali sulle colonne dell'allineamento [Zvelebil87], l'utilizzo di proprietà biologiche [King96], kNN [Salamov95], e reti neurali [Rost93], [Pollastri02], [Jones99]. Va ricordato inoltre il programma Predator [Frishman96] [Frishman97] che utilizza le informazioni sui legami idrogeno per predire le strutture alpha e beta.

<table>
<tr><th>generazione</th><th>singolo residuo</th><th>regole esplicite</th><th>nearest neighbors</th><th>reti neurali</th></tr>
<tr><td>prima (singolo residuo)</td><td>Chou74
Garnier78</td><td>Lim74</td><td></td><td></td></tr>
<tr><td rowspan="2">seconda (interazioni locali)</td><td rowspan="2">Garnier87</td><td></td><td>Levin86</td><td>Holley89
Qian88</td></tr>
<tr><td rowspan="2">Frishman96</td><td colspan="2">Lander93</td></tr>
<tr><td rowspan="2">terza (omologia)</td><td>Zvelebil87</td><td colspan="2">Salamov95</td></tr>
<tr><td colspan="2">King96</td><td></td><td>Rost93
Jones99
Pollastri02</td></tr>
</table>

Figura 4: **Panoramica sui metodi per la predizione di strutture secondarie**

In Figura 4 vengono mostrati i metodi citati classificati per generazione. I metodi più innovativi (in particolare quelli di terza generazione) verranno descritti più in dettaglio nel paragrafo 4.6.

4.4 Codifica degli aminoacidi

La predizione della struttura secondaria avviene associando a ciascun residuo della sequenza una delle classi strutturali che si sceglie di discriminare. Come abbiamo visto la ripartizione più comunemente utilizzata identifica tre strutture: α-eliche, β-strand e random coil. I metodi basati su reti neurali utilizzano tutti una codifica molto simile. La rete neurale riceve in ingresso una finestra di sequenza di $2n+1$ residui, che consiste nel residuo di cui si vuole predire la struttura secondaria e gli n residui che lo fiancheggiano sulla sinistra e sulla destra della sequenza.

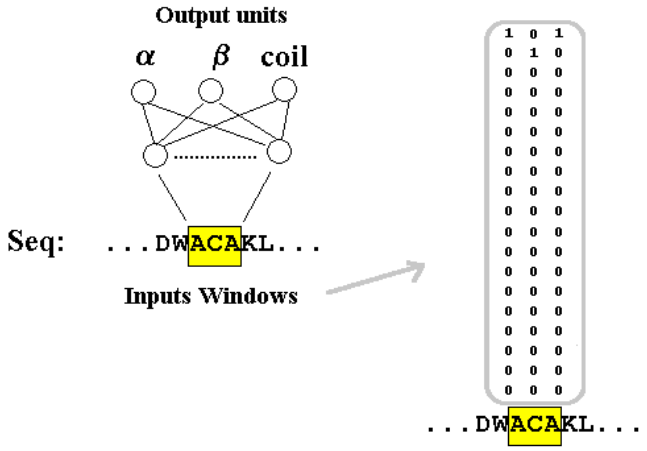


Figura 5: Rete neurale per la predizione di strutture secondarie

Nel caso più semplice, la finestra di ingresso è composta da 20N neuroni di ingresso (N=2n +1 residui), dove ciascun aminoacido è codificato da un gruppo di 20 neuroni. Tali neuroni sono posti tutti a *zero* eccetto quello che codifica il residuo a cui viene assegnato il valore 1 (codifica *one-shot*). Le posizioni fuori dal bordo della proteina possono essere codificate con un pattern tutto nullo oppure con un neurone aggiuntivo.

Nei metodi che utilizzano l'allineamento multiplo, il pattern in ingresso contiene le frequenze dei residui misurate nell'allineamento multiplo stesso.

Per lo strato di uscita la codifica può assumere varie forme, la più comune è comunque quella che codifica un neurone per ciascuna classe (*one-shot*).

In Figura 5 è mostrato un esempio di codifica per rete neurale.

4.5 Metodi di predizione di strutture secondarie

In questo paragrafo verranno illustrati brevemente i metodi di predizione di struttura secondaria più noti.

Per poter valutare le performance di un metodo di predizione e confrontarlo con altri esistenti è necessario introdurre degli indici che forniscano una misura quantitativa delle prestazioni. La più semplice misura è data dalla percentuale degli aminoacidi predetti correttamente:

$$Q_3 = \sum_i \frac{p_C}{N}$$

dove N è il numero totale di residui osservati e p_C è il numero di residui correttamente predetti. Gli indici Q_i indicano la frazione di predizioni corrette per ciascun tipo di struttura:

$$Q_i = \frac{p_i}{N_i} = \frac{p_i}{p_i + u_i}$$

dove N_i è il numero di residui osservati e u_i è il numero di residui predetti in maniera errata (sovrapredetti) nella classe i e p_i il numero di residui predetti correttamente. Per quanto molto comunemente usati Q_3 e Q_i essi non tengono in considerazione le sovrapredizioni e possono essere affetti dalla relativa abbondanza del tipo di strutture secondarie del data base. Perciò un indice più significativo è il coefficiente di correlazione C_i:

$$C_i = \frac{p_i n_i - u_i o_i}{\sqrt{(p_i + u_i)(p_i + o_i)(n_i + u_i)(n_i + o_i)}}$$

dove n_i è il numero di residui che non appartenendo alla struttura i sono stati correttamente assegnati ad altre strutture ed o_i è il numero di residui sovrapredetti in struttura i (cioè quei residui che pur non appartenendo alla classe i vengono classificati come classe i). Questo coefficiente assume i valori nell'intervallo [-1,1]; 1 indica la predizione completamente corretta, mentre 0 indica una predizione non migliore di quella casuale.

In alcuni casi può essere conveniente prendere in considerazione l'errore quadratico medio della rete per un dato insieme:

$$MSE = \frac{1}{N} \sum_p \sum_i \left({y_i}^p - {t_i}^p \right)^2$$

in cui p è l'indice che rappresenta gli N patterns di ingresso, i scorre su tutte le possibili uscite, mentre y_i^p e t_i^p sono rispettivamente l'uscita e il valore atteso per l'i^{ma} classe e per il pattern p.

In seguito è riportata la descrizione dei più importanti metodi di predizione di struttura secondaria insieme con la loro precisione complessiva.

4.5.1 DSC

Il programma DSC [King96] è caratterizzato dall'utilizzo di statistiche lineari per combinare i risultati ottenuti dall'applicazione di regole. Per prima cosa vengono selezionati i "concetti" importanti da un punto di vista biologico per la predizione di strutture secondarie proteiche fra cui:

- le propensioni conformazionali dei residui
- idrofobicità
- posizioni dei gap nell'allineamento
- proprietà di conservazione dei residui
- autocorrelazione

L'importanza di questi contributi è poi pesata singolarmente con un metodo statistico per ottenere la struttura complessiva della sequenza.

4.5.2 PHD

PHD [Rost93] è uno dei primi metodi basati su reti neurali che fa uso dell'informazione evolutiva per effettuare la predizione di strutture secondarie.

L'aspetto più innovativo è quello della codifica dei pattern da fornire alla rete neurale che è stato ripreso da tutti i metodi successivi. Viene prima effettuata una ricerca della sequenza in esame in un database di sequenze note usando BLASTP [Altschul90]. In questo modo viene estratto un insieme di sequenze dal database che hanno una somiglianza elevata con quella in esame e viene creato un allineamento multiplo con il programma ClustalW [Thompson94].

L'addestramento della rete neurale e la predizione della struttura secondaria è ottenuta fornendo alla rete l'intero profilo che contiene informazioni aggiuntive che non sono presenti direttamente nella sequenza originale. Infatti il profilo contiene

le informazioni sulla capacità di ogni di ogni residuo della sequenza di sostituirsi con gli altri aminoacidi senza compromettere la struttura e la funzione proteica. Ogni residuo è codificato con un pattern di lunghezza 20+1 che contiene in ogni posizione la frequenza con cui ogni aminoacido si sostituisce con quello della sequenza in esame, ottenendo perciò un profilo più smussato rispetto a quello della one-shot (Figura 6).

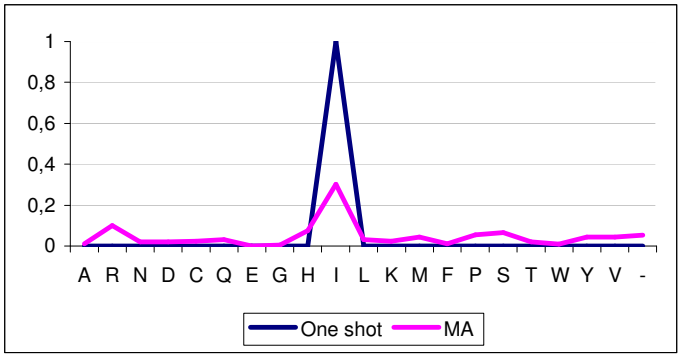


Figura 6: Codifica di un residuo con one-shot e con allineamento multiplo

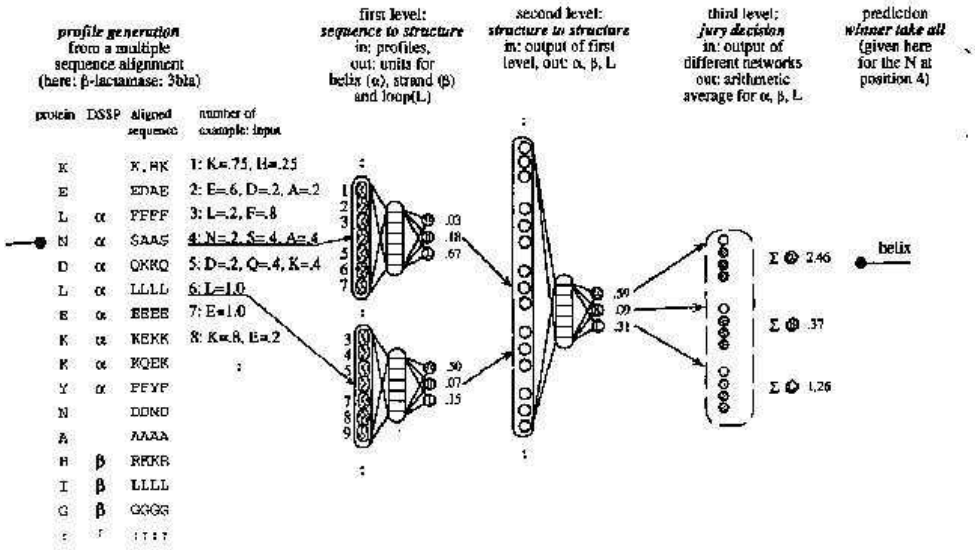


Figure 2. Our network system for secondary structure prediction. Our network system for predicting secondary structure consists of 3 layers: 2 network layers and 1 layer averaging over independently trained networks. ●, Basic cell containing 20 + 1 units to code residues at position 1 to *w* of the input window; here, *w* = 7. ●, Hidden units. Circled α, β and L, output units for helix, strand and loop. Stippled circles, output from architectures not shown here. —●, Example: residue N at position 4 predicted to be in helix ●—.

Figura 7: Il sistema PHD

Come per altri sistemi che utilizzano le reti neurali anche in questo caso il pattern complessivo è ottenuto concatenando fra loro i pattern codificati con il profilo relativi ai residui di una finestra.

L'accuratezza complessiva del sistema è di circa $Q_3 = 70\%$ con miglioramenti rilevanti soprattutto per le strutture beta con $Q_{3\beta} = 65{,}4\%$ contro 46% dello stato dell'arte precedente.

Il sistema complessivo è formato da tre livelli di rete neurale (Figura 7). Il primo livello è alimentato da un blocco di allineamento multiplo di ampiezza *w* con la codifica appena descritta e ha lo scopo di effettuare una predizione della conformazione strutturale del residuo centrale. Il secondo livello è alimentato da una finestra di residui codificati con l'output del livello precedente cioè con le strutture secondarie predette con lo scopo di mediare la predizione tenendo conto della forte correlazione nel segnale secondario. Infine, il terzo livello si propone di combinare i risultati derivanti da 12 reti neurali differenti costruite con il procedimento descritto utilizzando diverse scelte dei parametri (pesi iniziali, bilanciamento delle classi, …).

4.5.3 PSIPRED

Ulteriori incrementi di performance sono stati ottenuti sia con miglioramenti della qualità degli allineamenti multipli, sia con nuove architetture di reti neurali più adatte al compito della predizione delle strutture secondarie.

Rispetto al programma PHD che utilizza allineamenti di proteine filtrate con BLAST, il programma PSIPRED [Jones99] utilizza il metodo di ricerca di similarità più accurato basato su PSIBLAST [Altschul97] che permette di selezionare proteine che presentano livelli di similarità anche molto bassa.

L'output della rete (in termini di struttura secondaria) viene poi filtrato con una seconda rete neurale in modo da correggere gli eventuali errori di predizione dello strato precedente.

In Figura 8 è mostrata l'architettura complessiva di PSIPRED. Le performance stimate per PSIBLAST sono di circa $Q_3 = 75\%$.

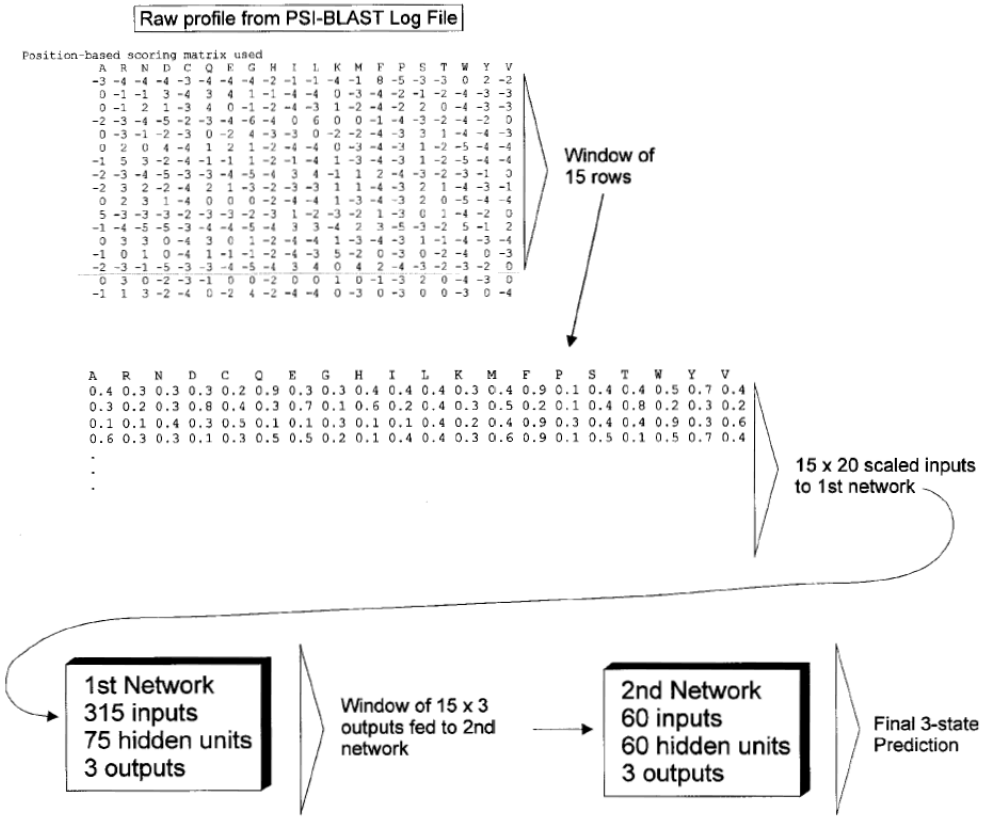


Figura 8: Struttura di PSI-PRED

4.5.4 SSPRO

Il sistema SSPRO [Pollastri02] è caratterizzato dall'utilizzo di reti neurali ricorrenti bidirezionali (BRNN, Bidirectional Recurrent Neural Network), codifica degli input basata sull'output di PSI-BLAST e utilizzo di metodi ensembles per migliorare l'accuratezza del sistema.

Due reti neurali ricorrenti percorrono la sequenza proteica in entrambe le direzioni a partire dai due estremi fino ad arrivare all'input sul quale si vuole effettuare la predizione. L'output complessivo viene prodotto combinando queste due reti neurali con un'ulteriore rete neurale che legge una finestra residui centrata sull'input. Questa architettura (Figura 9) consente di tener conto delle relazioni fra posizioni lontane nella sequenza che è uno dei problemi principali che si incontrano nella predizione di strutture secondarie. Infatti come già detto più volte

la struttura proteica è determinata in gran parte anche dalle complesse interazioni che si formano fra zone della sequenza distanti fra loro.

Un ulteriore incremento nell'accuratezza si ottiene combinando l'output di 11 reti BRNN ottenute con strategie indipendenti. La diversità è ottenuta utilizzando diverse strategie di traning e modificando i parametri della rete neurale quali il numero di neuroni nascosti e il numero di neuroni in ingresso (dimensione della finestra).

L'accuratezza dichiarata da questo sistema supera il 78% (Q_3) nella versione a tre stati d'uscita (è presente anche la versione a 8 stati, SSpro8, che raggiunge il 63%).

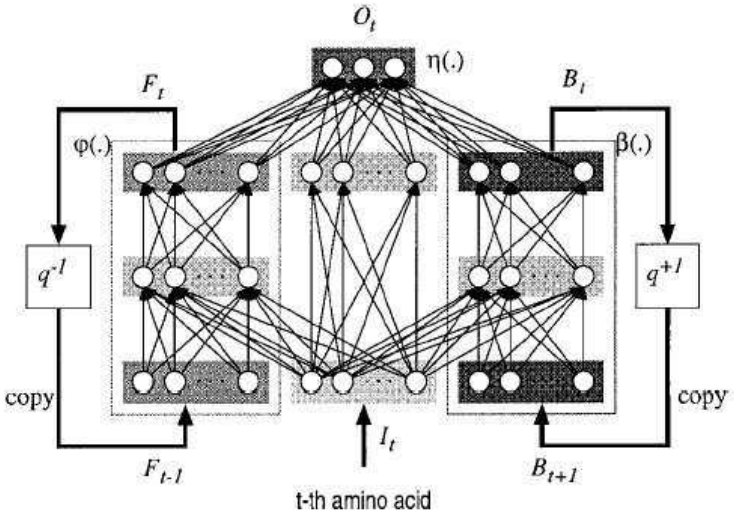


Fig. 1. A BRNN architecture with a left (forward) and right (backward) context, F_t and B_t, associated with two recurrent networks (wheels). The output layer O_t has three normalized exponential units associated with membership in each one of the three secondary structure classes for the current residue at position t. The functions ϕ, β, and η are implemented by feed-forward neural networks.

Figura 9: L'architettura BRNN utilizzata da SSPRO

Capitolo 5
Sistemi ad Esperti Multipli

In questo capitolo verrà introdotta l'architettura ad esperti multipli *NXCS* che presenta delle caratteristiche innovative rispetto alle architetture presenti in letteratura. In particolare permette di sfruttare le potenzialità dei metodi basati su classificatori multipli e nello stesso tempo di inserire nel sistema informazioni legate alla conoscenza del dominio.

5.1 Introduzione

La tecnica del "Dividi et impera" nei problemi di classificazione consiste nel partizionare ricorsivamente lo spazio degli input fino a quando non si raggiungono delle regioni omogenee per classe. Si tratta di una strategia molto utilizzata per controllare la complessità della ricerca nell'ambito dell'intelligenza artificiale e in particolare nel Machine Learning –Decision Lists (DL) [Rivest87], [Clark89], Counterfactuals (CFs) [Vere80], Classification And Regression Trees (CART) [Breiman84]– che si contrappone agli approcci tradizionali in cui si cerca di costruire un modello utilizzando un classificatore monolitico con visibilità globale sullo spazio degli input.

Utilizzando un'interpretazione alternativa la tecnica di partizione può essere vista come uno strumento per generare una popolazione di classificatori (Esperti Multipli) ognuno focalizzato su un particolare sottoinsieme dei dati. L'approccio ad Esperti Multipli è stato adottato, sebbene con da un punto di vista differente, sia dalla comunità dei sistemi evolutivi che da quella dei sistemi connessionisti. Nel primo caso gli studi si sono concentrati sulle architetture e sulle tecniche in grado di implementare un comportamento adattivo su una popolazione di classificatori

–Genetic Algorithms (GAs) [Holland75], [Goldberg89], Learning Classifier Systems (LCSs) [Holland76], [Holland86], e eXtended Classifier Systems (XCSs) [Wilson95]. Nel secondo caso la ricerca si è concentrata soprattutto sulle tecniche di training e di combinazione degli output –Jordan&Jacobs' Mixtures of Experts (MEs) [Jacobs91], [Jordan92] e Weigend's Gated Experts (GEs) [Weigend95].

Altri studi si sono focalizzati sul comportamento di popolazioni di esperti eterogenei rispetto a quello dell'esperto con visibilità globale. Studi teorici e risutalti sperimentali riguardanti soprattutto la teoria dell'aprendimento computazionale e statistico –[Valiant84] e [Vapnik98]– hanno mostrato dei miglioramenti significativi delle performance con l'utilizzo di sistemi basati su esperti multipli.

Altri studi rilevanti riguardano l'utilizzo di classificatori multipli basati su Artificial Neural Network (ANN) [Krogh95], [Breiman96a] e Decision Tree (DT) [Freund97], [Shapire99].

5.2 Architettura Guarded Experts

In questa sessione verrà introdotta formalmente l'architettura a esperti multipli "Guarded Experts" che propone alcuni spunti innovativi rispetto alle precedenti in quanto permette. In particolare l'architettura presenta due livelli il cui apprendimento può essere studiato separatamente: la micro-architettura che rappresenta la struttura interna di un esperto e la macro-architettura che rappresenta la struttura del sistema complessivo intesa come popolazione di esperti.

5.2.1 Micro Architettura

Indichiamo con I e O gli spazi degli input e degli output e con Λ l'oracolo cioè la fuzione in grado di fornire l'output corretto per ogni input: $\Lambda(x) \in O \quad \forall x \in I$. Un esperto globale H è una funzione che approssima l'oracolo nell'intero spazio degli input; viceversa, l'esperto locale h, ha una visione locale dello spazio e perciò approssima l'oracolo solo in un sottoinsieme di I.

Un guarded expert $\Gamma =< g, h >$ (Figura 1) è un esperto caratterizzato da un classificatore h la cui attivazione dipende da una guardia g posta a monte. La guadia è una generica funzione in grado di riconoscere o meno un'input $g : I \rightarrow \{True, False\}$.

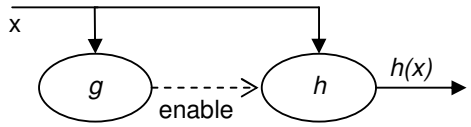


Figura 1: Micro architettura del guarded espert

La funzione della guardia all'interno dell'architettura dell'esperto può essere interpretata intuitivamente nel seguente modo: durante l'apprendimento l'esperto sarà obbligato a vedere solo gli input che la guardia lascia passare per cui si specializzerà nella predizione di tali input tralasciando gli altri. L'ipotesi di base che viene fatta per giustificare l'architettura è quella che il compito dell'esperto locale risulta semplificato rispetto a quello globale in quanto deve trattare una complessità minore. Sarà il meccanismo di apprendimento a garantire che ogni esperto locale effettivamente si specializzi sul dataset di competenza cioè abbia delle performance significativamente maggiori rispetto all'esperto globale.

Dato l'esperto $\Gamma =< g, h >$ si indica con I_g lo spazio degli input riconosciuti dalla guardia, cioè gli input di competenza dell'esperto:

$$I_g = \{x \in I \mid g(x) = True\}$$

In generale gli esperti avranno domini di competenza parzialmente condivisi, come mostrato in Figura 2, e la loro combinazione avverrà proprio su questi input in comune.

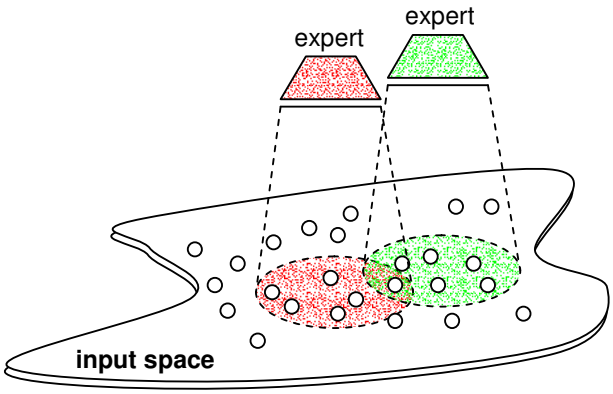


Figura 2: Rappresentazione di due esperti e con il loro dominio di competenza

Una possibile generalizzazione si ottiene definendo la guardia come una funzione che mappa gli input nell'intervallo $[0,1]$ piuttosto che nell'insieme $\{True, False\}$. In questo caso il valore $g(x)$ può essere interpretato come "grado di competenza" dell'esperto per l'input x e permette di definire meccanismi di voting più sofisticati come vedremo in seguito.

5.2.2 Macro Architettura

La macro architettura del sistema ovvero la descrizione della popolazione e delle relazioni fra esperti può essere specificata solo dopo aver fatto delle scelte progettuali che solitamente sono strettamente dipendenti dal particolare task applicativo.
Tali scelte riguardano le strategie e le tecniche di apprendimento, le regole per lo split delle regioni, i meccanismi di selezione, i meccanismi di "outputs blending" e le politiche di voting.

Tecniche e modalità di training. In generale le modalità di training possono essere batch (learned system) oppure online (learning system). Nelle prime si ha una netta separazione fra la fase di addestramento e quella di utilizzo del sistema e rappresenta la modalità più comune nei problemi di classificazione. Nella seconda il processo di apprendimento coincide con quello di predizione per cui il sistema, in

continua fase di apprendimento, deve aggiornare il suo stato in maniera incrementale in base al reward dell'ambiente.

Le tecniche di training, la cui scelta è strettamente dipendente dalla tecnologia utilizzata, hanno come scopo quello di ottimizzare una funzione complessiva di costo (o di merito) associata al reward che il sistema riceve dall'ambiente. Fra le tecniche più comuni ricordiamo le euristiche di information-gain per gli alberi di decisione e l'algoritmo di backpropagation (BP) per le reti neurali. Altre tecniche più sofisticate specifiche per i sistemi ad esterti multipli sono l'algoritmo di Expectation-Maximization (EM) [Jacobs91], il bagging [Breiman96a] e il boosting [Freund97]. Diversi studi sono stati fatti per spiegare il miglioramento di performance che si può ottenere con un sistema ad esperti multpli addestrato con bagging o boosting rispetto all'esperto singolo. Risultati sperimentali e teorici hanno messo in evidenza sia la capacità di questi due metodi di generare esperti indipendenti o addirittura correlati negativamente, sia di ridurre la varianza e migliorare il margine [Domingos00], [Schapire97], [Breiman96b].

Split delle regioni e meccanismi di selezione. In un sistema ad esperti multipli ogni singolo esperto ha una visione locale dello spazio degli input in quanto la guardia riconoscerà soltanto il sottoinsieme $I_g \subseteq I$. Le guardie perciò effettuano uno splitting dello spazio in regioni eventualmente overlappanti i cui bordi sono separati in maniera hard/soft a seconda della tipologia della guardia. La separazione hard, tipica dei sistemi DLs, DTs e CART, si ottiene con una guardia booleana che decide in maniera netta se un input appartiene o meno al range di competenza dell'esperto. Viceversa la separazione soft, tipica dei sistemi MEs e GEs, si ottiene con una guardia con uscita nell'intervallo (0,1) ed in questo caso è possibile definire, per ogni input, il grado di competenza dell'esperto che avrà sempre una visibilità globale. I sistemi con separazione soft sono utili quando nell'applicazione in esame è definito in maniera naturale il concetto di metrica soft e possono essere usati anche quando si vuole forzare ad avere zone overlappanti i sistemi come i DTs che non supportano l'overlapping.

Per quanto riguarda i meccanismi di selezione, l'unico caso interessante da menzionare è quello in cui si ha un overlap delle zone di competenza nello spazio degli input. In questo caso viene a crearsi un match-set definito come l'insieme

degli esperti che riconoscono un certo input e perciò sono "competenti" per trattare quel particolare input: $M_x = \{h_k \mid g_k(x) = True\} \ \forall x \in I$.

Solo gli esperti del match-set concorrono a fornire una predizione per quel particolare input che potrà poi essere pesata in base al grado di competenza o alla fitness.

Meccanismi di outputs blending e politiche di voting. Un sistema ad esperti multipli utilizzato come classificatore deve produrre un unico output per un dato input. Per fare questo deve combinare le uscite che si ottengono dall'interrogazione di tutti gli esperti singoli coinvolti nel match-set. Per semplicità di analisi trattiamo solo i casi più comuni e inoltre consideriamo gli esperti omogenei per quanto riguarda l'encoding in ingresso e uscita:

- *Uscita singola nominale* (classificazione)

  L'uscita è una label nominale su un alfabeto predefinito di *m* classi, per cui l'esperto multiplo deve applicare una politica di voting per combinare tutte le uscite dei singoli per ottenere un'unica label.

  La tecnica più diffusa è quella del majority-voting in cui si sceglie la classe più rappresentata nelle singole uscite. La generalizzazione naturale tiene conto anche della *strength* del esperto e/o dell'*expertise* cioè del grado di competenza dell'esperto per il particolare input.

- *Uscita singola numerica* (classificazione)

  L'uscita è un numero, solitamente normalizzato in [-1,1] (oppure in [0,1]), che codifica la classe in un problema a due classi in modo che valori negativi sono associati ad una classe e valori positivi all'altra classe. Il valore assoluto dell'output può essere interpretato come accuratezza o pseudo probabilità. In questo caso l'*outputs blending* consiste solitamente nella media pesata dei valori di output dei singoli.

- *Uscita multipla numerica* (classificazione)

  L'uscita è un pattern che solitamente è codificato in modo che ogni elemento rappresenta la propensione della classe (normalizzata in [0,1]). In questo caso si esegue prima un *output blending* in modo da ottenere un pattern complessivo rappresentante l'uscita del sistema, e poi si applica un criterio di voting, per esempio il massimo della classe.

In Figura 3 è mostrata la macro architettura del sistema *NXCS*.

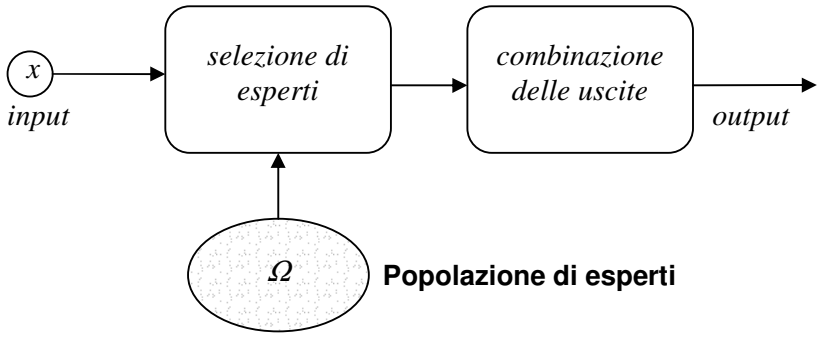


Figura 3: Macro Architettura

5.3 Architettura XCS neurale (*NXCS*)

In questo paragrafo verrà presentata l'architettura *NXCS* come una particolare implementazione dei Guarded Experts in cui si utilizza una strategia *XCS* per l'apprendimento delle guardie g e dei classificatori neurali per implementare i singoli classificatori h.

I classificatori *XCS* [Wilson95], strettamente legati agli *LCS* di Holland e al paradigma dell'apprendimento per rinforzo, sono stati studiati approfonditamente da un punto di vista teorico –[Kovacs99] e [Wilson98]– e l'architettura originale di Wilson è stata estesa con diversi miglioramenti [Lanzi98], [Lanzi99] e [Wilson99].

L'importanza di questa architettura risiede nel fatto che i classificatori *XCS* sono particolarmente adatti a modellare la conoscenza del dominio in una forma del tutto arbitraria utilizzando delle stringhe ternarie (0,1,#). D'altra l'output costante del singolo classificatore *XCS* non permette di estrarre informazioni direttametne dagli input.

Il sistema *NXCS* proposto cerca di trarre i vantaggi dell'architettura *XCS* e nello stesso tempo di migliorare il loro comportamento in fase di predizione utilizzando delle reti neurali che sono sono largamente riconosciute come degli approssimatori universali di funzioni.

Un esperto *NXCS* $\Gamma =< g, h, w >$ (Figura 4 e Figura 5) è caratterizzato da una guardia g che è un classificatore XCS che mappa gli input su $\{True, False\}$, un classificatore h che è una rete neurale feed-forward addestrata e attivata solo sul

sottoinsieme di input riconosciuti dalla guardia, e infine da un peso *w* che rappresenta la *strength* dell'esperto che viene aggiornata in base all'accuratezza nella predizione.

La guardia processa l'input e lo codifica in un messaggio dato da una stringa di simboli '0' e '1' utilizzando opportune feature. Come verrà descritto nei capitoli successivi il matching avviene confrontando la stringa che rappresenta l'input con quella della guardia.

E' importante notare come gli esperti *NXCS* possono essere descritti sia dal punto di vista evolutivo che da quello connessionista. La prima interpretazione vede un esperto *NXCS* come un classificatore *XCS* nel quale l'azione è calcolata con una rete neurale anziché essere una costante. La seconda interpretazione invece considera l'esperto come una ANN equipaggiata con una guardia che gli permette di selezionare un contesto (classificazione) o un regime (predizione).

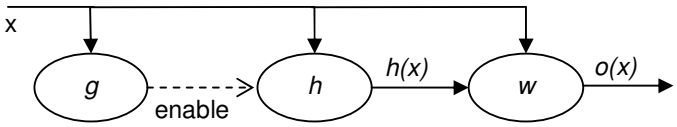


Figura 4: Esperto *NXCS*

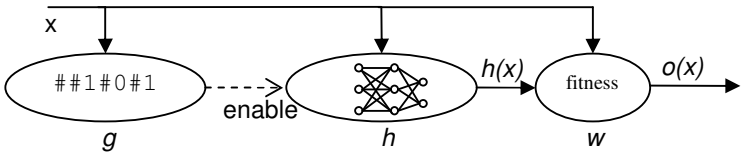


Figura 5: Esperto *NXCS* in dettaglio

La tecnica di training per i sistemi *NXCS* è conforme a quella adottata per gli *XCS* in cui la popolazione è selezionata in base all'accuracy che viene aggiornata a partire dal reward che il sistema riceve dall'ambiente. Un tipico loop di apprendimento di un sistema *NXCS* (classificazione) è riportato in Figura 6.

Gli operatori genetici seguono le classiche regole dei GAs utilizzando come cromosoma la stringa della guardia. La selezione degli esperti avviene in maniera

proporzionale alla fitness nella mutazione e nel crossover e inversamente proporzionale alla fitness per la cancellazione. In questo modo i nuovi esperti generati avranno genitori con fitness elevata mentre gli esperti con fitness bassa avranno una maggior probabilità di essere eliminati. I nuovi esperti generati verranno inseriti nella popolazione solo dopo che il loro classificatore (rete neurale) è stato addestrato sul dataset riconosciuto dalla loro guardia. Dopo l'addestramento i pesi della rete neurale resteranno costanti.

Step 0	Si parte con una popolazione vuota.
Step 1	Si preleva un input *i* dal training set e si costruisce il match-set M formato da tutti gli esperti che riconoscono l'input.
Step 2	Se M è vuoto si crea un nuovo esperto con una guardia tale da coprire l'input in esame (covering) e si addestra la rete neurale sul dataset di training.
Step 3	Per ogni esperto del match-set si calcola la sua predizione e si costruisce il vettore di predizione Pi utilizzando una tecnica di combinazione fra quelle descritte. In questo modo si trova l'azione complessiva del sistema a^*.
Step 4	Si preleva il reward dall'ambiente $r = r(i, a^*)$ e si ,aggiornano i parametri degli esperti del match-set (prediction, prediction error, strength).
Step 5	Se il tempo medio dall'ultima operazione genetica ha superato una certa soglia si applica un operatore genetico (crossover, mutazione, cancellazione) sugli esperti del match-set.
Step 6	Se viene raggiunto un certo criterio di stop l'apprendimento termina, altrimenti si riparte dallo step 1.

Figura 6: Apprendimento di un sistema NXCS

La mutazione (Figura 7) si esegue generando un nuovo esperto identico al genitore ma con un carattere della stringa di guardia modificato. Il crossover invece (Figura 8) si esegue spezzando la stringa delle due guardie dei genitori nello stesso punto e incrociando le due parti. Un metodo alternativo per effettuare il crossover è quello di selezionare le posizioni da scambiare lungo la stringa tramite un campionamento casuale. Il covering invece si ottiene generando inizialmente una guardia identica

all'input che viene poi eventualmente generalizzata con l'aggiunta di dont'care in maniera casuale.

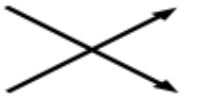
Mutation of a bit vector:

11**0**10110 → 11**1**10110

Figura 7: Mutazione

Crossover of bit vectors

110|10110 → 110|01101



010|01101 → 010|10110

Figura 8: Crossover

Per chiarire meglio il meccanismo di voting per l'*NXCS* consideriamo un matchset *M* costruito a partire da un certo input. Il match-set viene partizionato in sottoinsiemi $M_a \subseteq M$ in base all'azione *a* (output) di ogni singolo esperto

$$M_a = \{c_k \mid c_k \in M \wedge h_k = a\}$$

La stima della prediction (reward) è valutata con la formula seguente, dove la prediction p_c di ogni esperto del match-set contribuisce a quella complessiva ma viene pesata con la strength f_c

$$P_a = \frac{\sum_{c \in M_a} f_c \cdot p_c}{\sum_{c \in M_a} f_c}$$

Nel caso in cui le guardie sono caratterizzate da un matching flessibile è possibile pesare la prediction tenendo conto anche del grado di competenza dell'esperto per il particolare input, indicato con y_c:

$$P_a = \frac{\sum_{c \in M_a} f_c \cdot y_c \cdot p_c}{\sum_{c \in M_a} f_c}$$

In questo caso il match-set potrebbe anche coincidere con tutta la popolazione.

L'azione complessiva dell'esperto multiplo sarà quella che massimizza il reward stimato e cioè: $a^* = \arg\max_a (P_a)$.

Capitolo 6
Sistema MASSP

In questo capitolo verrà descritto un sistema MASSP (MultiAgents Secondary Structure Predictor) derivato dall'architettura *NXCS* descritta in precedenza.

6.1 Rappresentazione del problema

Per rappresentare il problema verrà utilizzata una notazione ad agenti in cui il dominio è gestito da un agente *Environment* (Figura 1) che contiene al suo interno un dataset di catene proteiche. L'ambiente fornisce al sistema coppie input/output ogni volta che ne viene fatta richiesta. Un input è un aminoacido all'interno di una proteina mentre l'output è la struttura secondaria a cui appartiene l'aminoacido in esame: alfa (h), beta (e) e coil (c).

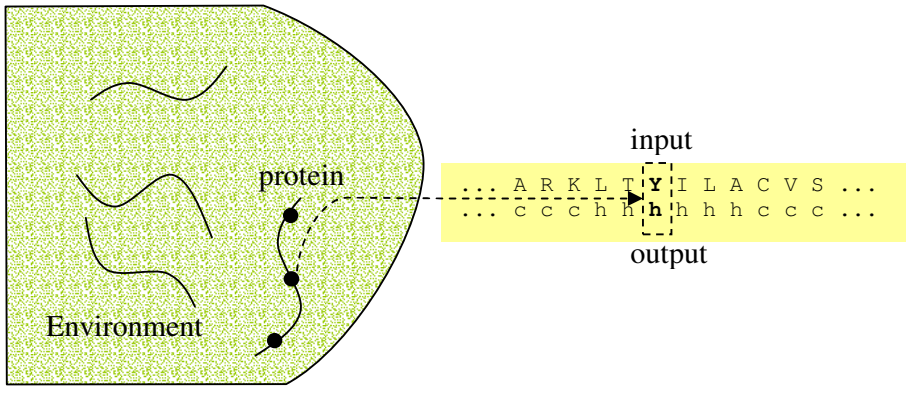


Figura 1: Ambiente per la predizione di strutture secondarie proteiche

Il problema della predizione di strutture secondarie consiste nell'apprendere da un dataset di proteine a struttura nota, le regole che permettono di assegnare la struttura secondaria ad una catena di aminoacidi arbitraria (Figura 2). Come è stato già detto, la struttura secondaria può essere poi utilizzata per creare un modello 3D delle proteina.

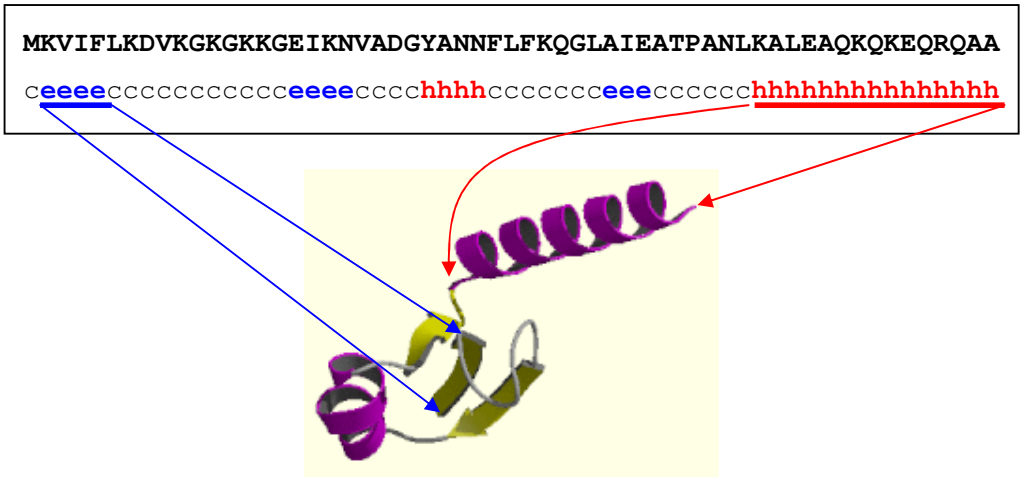


Figura 2: Predizione di struttura proteica

In generale l'ambiente del problema è di tipo simbolico sia in ingresso che in uscita in quanto l'input e l'output sono caratteri definiti negli alfabeti primari e secondari:

$$\{ACDEFGHIKLMNPQRSTVWY\} \text{ e } \{h,e,c\}$$

anche se il sistema, come vedremo, può processare questo input in modo da estrarre feature numeriche molto più adatte alla tecnologia delle reti neurali.

6.2 Architettura

Le guardie rappresentano quella parte del sistema in cui è racchiusa la conoscenza del dominio che in questo caso riguarda tutte le informazioni di tipo chimico-fisico che possono essere utili nel problema della predizione. Indichiamo con γ un vettore di N feature booleane cioè tali che $\gamma_k(x) \in \{0,1\} \quad \forall x \in I$, dove I è lo spazio degli input dell'ambiente.

Una guardia g definita sulle feature γ è data da una stringa di lunghezza N sull'alfabeto '01#' dove ogni carattere è associato ad una feature. La guardia "riconosce" un certo input x se tutte le feature associate al carattere 1 sono vere per l'input mentre quelle associate al carattere 0 sono false:

$$g(x) = True \Leftrightarrow \gamma_k(x) = g_k \qquad \forall k : g_k \neq '\#'$$

In pratica il matching fra la guardia e l'input avviene trasformando l'input in un messaggio m sull'alfabeto '01' dove l'elemento m_i sarà 0 o 1 a seconda del valore della rispettiva feature γ_i. Il messaggio viene poi confrontato con la stringa della guardia. Il match avviene se il messaggio e la guardia sono identici carattere per carattere con l'esclusione dei '#' (dont-care).

Nell'esempio in Figura 3 si ha un match fra la guardia e il messaggio in quanto, tralasciati i #, il resto dei caratteri sono identici.

messaggio	00100111
guardia	0#100#1#

Figura 3: Matching fra guardia e messaggio (input)

Il carattere '#' indica che quella particolare feature non è rilevante per la guardia al fine della valutazione dell'input e verrà perciò trascurata. Il numero di '#' indica la capacità di generalizzazione della guardia ed è una misura indiretta di quanti input è in grado di riconoscere. La guardia formata solo da '#' per definizione riconosce tutti gli input in quanto nessuna delle feature è discriminante. Questa guardia è importante da un punto di vista teorico perché è associata al classificatore monolitico.

La tecnica di matching che abbiamo mostrato è tipica degli algoritmi GAs dove il pattern viene utilizzato come cromosoma durante l'evoluzione. La differenza è che in questo caso il pattern non rappresenta direttamente l'output dell'esperto ma serve solo per selezionare gli input sui quali il classificatore dovrà poi specializzarsi. Inoltre la guardia non riceve un reward diretto ma viene premiata in base al comportamento del classificatore. In questo contesto, valori di fitness

elevata, sono associati a guardie che selezionano un sotto-spazio di input in cui un classificatore è in grado di apprendere meglio del classificatore con visibilità globale.

Il classificatore neurale processa gli input in modo da generare dei pattern numerici lunghezza $20 \times W$ dove W è la lunghezza di una finestra di residui centrata nel input.

In generale il pattern contiene informazioni di omologia della sequenza in esame e si ottiene a partire da un allineamento multiplo (Figura 4) della sequenza con sequenze omologhe . Per ogni colonna dell'allineamento multiplo viene estratto un pattern di lunghezza 20+1 che indica la frequenza di ogni aminoacido e del gap nella colonna. Il problema della scelta della codifica degli input verrà analizzato in maggior dettaglio nel paragrafo successivo.

```
GUX1_TRIRE/481-509   HYGQCGGI...GYSGPTVCASGTTCQVLNPYY
GUN1_TRIRE/427-455   HWGQCGGI...GYSGCKTCTSGTTCQYSNDYY
GUX1_PHACH/484-512   QWGQCGGI...GYTGSTTCASPYTCHVLNPYY
GUN2_TRIRE/25-53     VWGQCGGI...GWSGPTNCAPGSACSTLNPYY
GUX2_TRIRE/30-58     VWGQCGGQ...NWSGPTCCASGSTCVYSNDYY
GUN5_TRIRE/209-237   LYGQCGGA...GWTGPTTCQAPGTCKVQNQWY
GUNF_FUSOX/21-49     IWGQCGGN...GWTGATTCASGLKCEKINDWY
GUX3_AGABI/24-52     VWGQCGGN...GWTGPTTCASGSTCVKQNDFY
GUX1_PENJA/505-533   DWAQCGGN...GWTGPTTCVSPYTCTKQNDWY
GUXC_FUSOX/482-510   QWGQCGGQ...NYSGPTTCKSPFTCKKINDFY
GUX1_HUMGR/493-521   RWQQCGGI...GFTGPTQCEEPYICTKLNDWY
GUX1_NEUCR/484-512   HWAQCGGI...GFSGPTTCPEPYTCAKDHDIY
PSBP_PORPU/26-54     LYEQCGGI...GFDGVTCCSEGLMCMKMGPYY
GUNB_FUSOX/29-57     VWAQCGGQ...NWSGTPCCTSGNKCVKLNDFY
PSBP_PORPU/69-97     PYGQCGGM...NYSGKTMCSPGFKCVELNEFF
GUNK_FUSOX/339-370   AYYQCGGSKSAYPNGNLACATGSKCVKQNEYY
PSBP_PORPU/172-200   RYAQCGGM...GYMGSTMCVGGYKCMAISEGS
PSBP_PORPU/128-156   EYAACGGE...MFMGAKCCKFGLVCYETSGKW
```

Figura 4: Allineamento multiplo. La sequenza target è evidenziata.

Anche se non richiesto direttamente dall'architettura, risulta ragionevole alimentare la guardia e il classificatore con feature differenti. In particolare la guardia, essendo un riconoscitore di "contesto", è in grado di prelevare informazioni sull'input che si trovano in posizioni anche distanti o addirittura associate a l'intera sequenza. La rete neurale invece, seguendo anche la letteratura sulla predizione di strutture secondarie, è più adatta a maneggiare in informazione legata all'omologia di una sottosequenza.

L'algoritmo evolutivo su cui si basa il sistema MASSP rispecchia strettamente quello del NXCS descritto nel capitolo precedente. In particolare la popolazione degli esperti interagirà con l'ambiente prelevando gli input e selezionando gli esperti del match-set tramite il matching delle guardie (Figura 5).

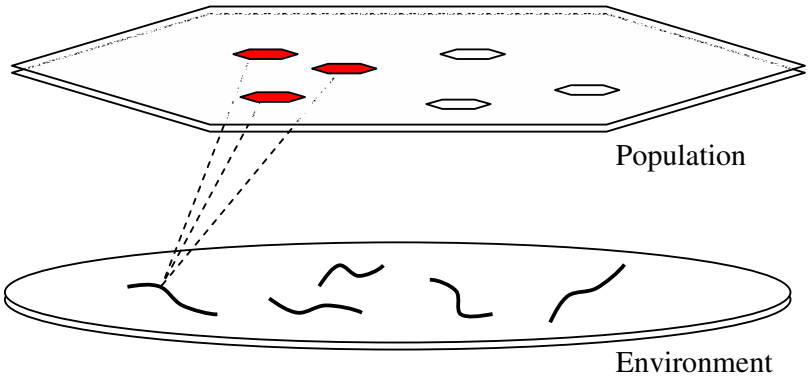


Figura 5: Popolazione che interagisce con l'ambiente. Gli esperti in rosso sono quelli che formano il match-set per un dato input.

Il meccanismo di calcolo della fitness è stato adattato alla particolare applicazione in modo da tener conto anche del fattore di copertura del singolo esperto che diventa una misura indiretta della difficoltà intrinseca di ogni input. Infatti i dataset utilizzati per la predizione di struttura secondaria oltre ad avere uno squilibrio sulle frequenze delle classi (35% alfa, 25% beta, 40% coil) presentano anche una certa variabilità riguardo al rumore sul segnale di omologia. Questo fa si che certi tipi di input (in particolare quelli di classe beta) sono particolarmente difficili da predire. E' stato dimostrato sperimentalmente che in un sistema basato unicamente sulla strength dell'esperto, l'evoluzione tende a favorire esperti con una visibilità bassissima o esperti che si concentrano su input semplici.

Per questo è stato definito un parametro che misura la qualità di un esperto in termini della sua capacità di copertura $C_e = |I_e|/|I|$, dove I_e è l'insieme di input riconosciuti dall'esperto *e* e I è il dataset complessivo.

6.3 Codifica degli input

Il problema della codifica degli input è di fondamentale importanza per facilitare il compito di predizione dei classificatori. Secondo la letteratura corrente l'utilizzo dell'informazione di tipo evolutivo derivante soprattutto da allineamenti multipli è in grado di fornire un notevole miglioramento delle prestazioni rispetto al caso della codifica standard one-shot. L'aumento di prestazioni è valutato a circa il 10-15% dove si è preso come riferimento la differenza di prestazioni di una rete neurale con codifica oneshot (60%) e con codifica del multiallineamento (75%).

In realtà, pur riconoscendo l'effettiva importanza di questo tipo di informazione, si potrebbe studiare questo fenomeno alla luce di due aspetti fondamentali, uno di tipo tecnologico legato al comportamento delle reti neurali e uno di tipo informativo legato all'effettiva informazione che un allineamento multiplo riesce ad fornire in più rispetto alla conoscenza dei singoli aminoacidi.

Da un punto di vista tecnologico le reti neurali presentano difficoltà a generalizzare quando devono lavorare con feature simboliche, soprattutto se, per mancanza di informazioni aggiuntive, si deve adottare la codifica one-shot. Infatti, sebbene la rete neurale abbia un potere espressivo maggiore rispetto ad un semplice classificatore come il naive-bayes, risultati sperimentali dimostrano che utilizzando le sole label dei residui senza ulteriori informazioni di correlazione i due classificatori hanno prestazioni circa uguali (60%) e il naive-bayes presenta una capacità di generalizzazione maggiore.

D'altra parte utilizzando schemi di codifica che tengono conto dell'effettiva relazione esistente fra aminoacidi si ottengono dei risultati significativamente maggiori del 60% pur non utilizzando nessun tipo di informazione evolutiva. Infatti una semplice codifica basata su matrice di sostituzione omogenea (es BLOSUM62) porta a delle prestazioni del 65% mentre accurati sistemi di codifica a 3 feature ottenuti con tecniche di apprendimento automatico si arriva addirittura a prestazioni del 68% [Riis96]. Da queste osservazioni si deduce che l'effettivo miglioramento derivante da informazioni di tipo evolutivo sia quantificabile in circa 7%.

6.3.1 Codifica a due livelli

Gli studi delle codifiche basate su matrici di sostituzione sono stati approfonditi soprattutto in relazione al tipo di informazione che racchiudono. Le matrici di sostituzione, essendo calcolate come valori medi su dataset proteici molto grandi, contengono informazioni di carattere generale sulla sostituibilità degli aminoacidi che potrebbero non essere presenti in un allineamento che invece è strettamente correlato con la proteina in esame. Un problema della codifica basata sull'allineamento multiplo infatti è quello di contenere informazioni che possono essere troppo specifiche per l'allineamento e perciò nascondere delle caratteristiche importanti nella sequenza in esame. Un esempio tipico si ha quando la sequenza presenta un'aminoacido poco frequente nella colonna dell'allineamento. Se questa sostituzione è fondamentale per la funzione della proteina viene introdotto un rumore di fondo nella codifica che difficilmente verrà corretto dalla rete neurale. Questo effetto è inoltre molto amplificato dagli errori che ci possono essere nell'allineamento stesso soprattutto nelle zono a bassa similarità.

In Figura 6 è mostrato un esempio di questo fenomeno in cui la metionina della sequenza target (l'ultima) è poco rappresentata nell'allineamento.

```
...VWGQCGGQ...NWSGPTCCASGSTCVYSNDYY...
...HYGQCGGI...GYSGPTVCASGTTCQVLNPYY...
...HWGQCGGI...GYSGCKTCTSGTTCQYSNDYY...
...LYGQCGGA...GWTGPTTCQAPGTCKVQNQWY...
...QWGQCGGI...GYTGSTTCASPYTCHVLNPYY...
...QWGQCGGQ...NYSGPTTCKSPFTCKKINDFY...
...IWGQCGGN...GWTGATTCASGLKCEKINDWY...
...VWGQCGGN...GWTGPTTCASGSTCVKQNDFY...
...DWAQCGGN...GWTGPTTCVSPYTCTKQNDWY...
...HWAQCGGI...GFSGPTTCPEPYTCAKDHDIY...
...RWQQCGGI...GFTGPTQCEEPYICTKLNDWY...
...VWAQCGGQ...NWSGTPCCTSGNKCVKLNDFY...
...LYEQCGGI...GFDGVTCCSEGLMCMKMGPYY...
...PYGQCGGM...NYSGKTMCSPGFKCVELNEFF...
...AYYQCGGSKSAYPNGNLACATGSKCVKQNEYY...
...RYAQCGGM...GYMGSTMCVGGYKCMAISEGS...
...EYAACGGE...MFMGAKCCKFGLVCYETSGKW...
```

Figura 6: Colonna di un'allineamento multiplo in cui è avvenuta una sostituzione fondamentale

La codifica proposta ha lo scopo di utilizzare contemporaneamente questi due tipi di informazione: quello globale (I livello) riferito alle proprietà generali degli aminoacidi e quello locale (II livello) riferito allo specifico input tenendo conto del contesto proteico in esame.

Definiamo prima alcune notazioni:

- Fissato un ordinamento arbitrario degli aminoacidi, assegnamo un indice da 1 a 20 ad ogni aminoacido (per esempio 1/Alanina, 2/Arginina, 3/Asparagina, …, 20/Valina) e riserviamo l'indice 0 al gap.
- Sia $\mathbf{P} = \langle P_i, i = 0,1,...,n \rangle$ un allineamento multiplo di lunghezza L dato da un insieme di catene proteiche allineate dove P_0 è la proteina di riferimento (senza gap) mentre le altre sono proteine correlate con P_0 che possono contenere anche gap. Indichiamo con $P_i(j)$ il j-esimo elemento della sequenza P_i che può essere un residuo o un gap.
- Sia **B** una matrice di sostituzione 20×20 normalizzata in 0-1 e indichiamo con B_k ($k = 1,2,...,20$) la riga *k* della matrice che codifica per l'aminoacido *k* e con $B_k(r)$ il grado di sostituibilità degli aminoacidi *k* e *r*. La riga e la colonna di indice 0 rappresentano la codifica del gap che sarà tutta nulla eccetto l'elemento $B_0(0)$ che è assegnato a 1.
- Sia **Q** una matrice $21 \times L$ che rappresenta la codifica finale associata alla sequenza in esame P_0. $Q(j)$ indica la j-esima colonna della matrice che rappresenta la codifica del j-esimo aminoacido della sequenza $P_0(j)$, mentre $Q_r(j)$ rappresenta il contributo dell'aminoacido *r* nella codifica.

La normalizzazione della matrice **B** è fatta con il seguente criterio:

1. Indicati con μ e σ la media e la deviazione standard della matrice di sostituzione iniziale **S**, viene calcolata la matrice equalizzata **E** applicando un'opportuna sigmoide centrata in μ e con ampiezza $[-\sigma, +\sigma]$:

$$E_k(j) = \sigma \cdot \tanh\left(B_k(j) - \mu\right) \qquad \forall k, j = 1,2,...,20$$

2. Siano E^m e E^M il valore massimo e minimo della matrice equalizzata **E**; la matrice normalizzata è calcolata nel seguente modo:

$$B_k(j) = \frac{E_k(j) - E^m}{E^M - E^m} \qquad \forall k, j = 1,2,...,20$$

$$B_0 = B(0)^T = [1,0,0,...,0]$$

L'algoritmo usato per calcolare la codifica della proteina P_0 è il seguente

1. Inizializzo **Q** con la codifica basata su matrice di sostituzione calcolata lungo la sequenza P_0:

$$Q(j) = B_{P_0(j)}{}^T \qquad \forall j = 1,2,...,L$$

2. Aggiungo ai valori di **Q** i pattern derivanti dalle stesse codifiche calcolate lungo le altre sequenze dell'allineamento

$$Q(j) = Q(j) + B_{P_i(j)}{}^T \qquad \forall j = 1,2,...,L \;\; \forall i = 1,2,...,n$$

3. Normalizzo gli elementi di **Q** in [0-1]

$$Q_r(j) = \frac{Q(j)}{\sum_s Q_s(j)} \qquad \forall j = 1,2,...,L \;\; \forall r = 0,1,...,20$$

E' importante notare a questo punto che la valutazione della matrice **B** viene fatta una sola volta in quanto è indipendente dalla proteina. La codifica **Q** invece è calcolata per ogni proteina

La codifica mista appena descritta tiene conto sia dell'informazione derivante dall'allineamento multiplo che quella derivante da una matrice di sostituzione (per esempio BLOSUM80). In Figura 7 è vengono confrontate le codifiche oneshot (OS), allineamento multiplo (MA), e allineamento multiplo pesato con la matrice di sostituzione (MAS). La codifica OS prende in considerazione soltanto l'aminoacido della sequenza in esame ed è la meno informativa, la codifica MA considera le frequenze di sostituzione ma presente spesso una forma molto simile alla OS in soprattutto nelle zone altamente conservate. La codifica MAS invece tende a dare un peso notevole anche a quegli aminoacidi che presentano un grado

di sostituibilità indipendentemente dal fatto che nell'allineamento vengano sostituiti

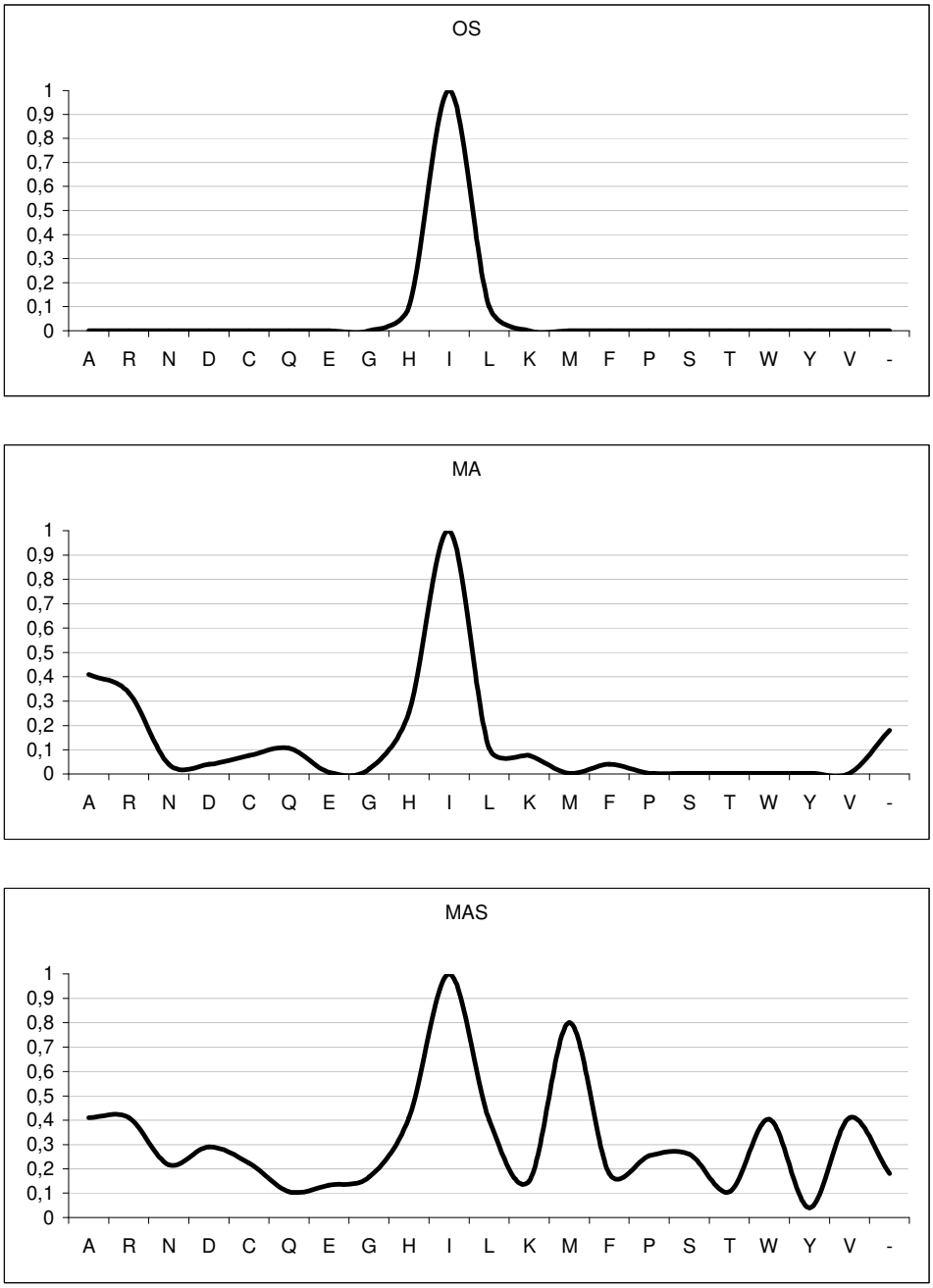
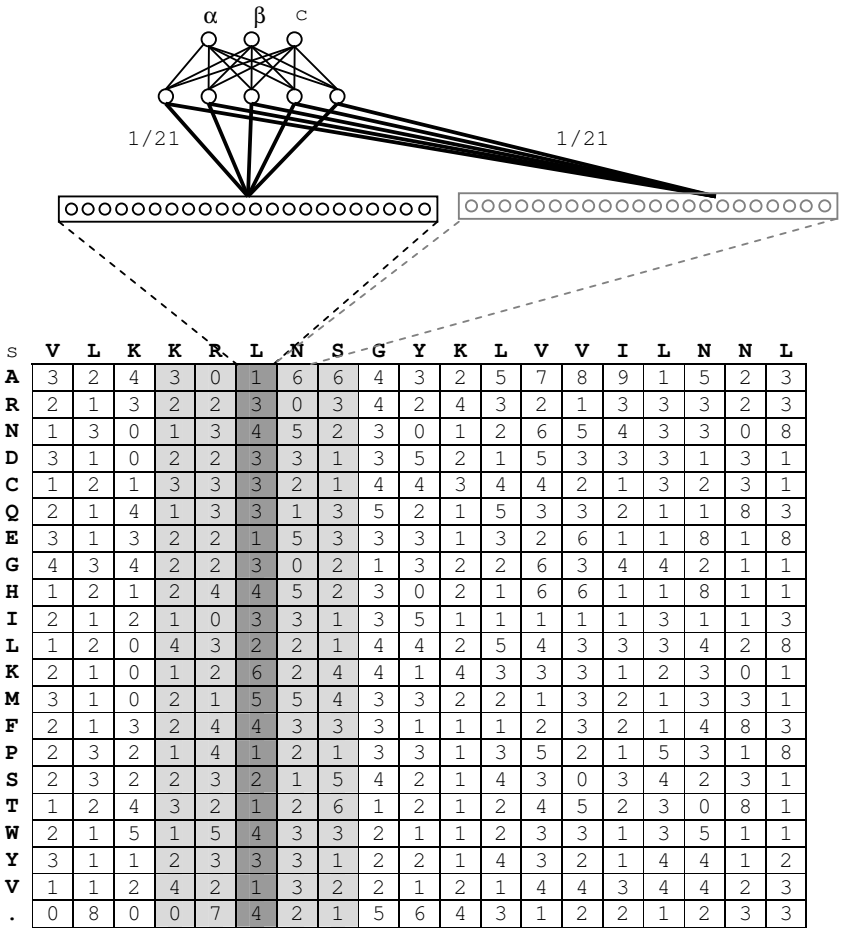


Figura 7: Codifiche oneshot, MA e BLOSUM80+MA

In accordo con i risultati sperimentali, la codifica appena definita riduce notevolmente l'overfitting e produce un miglioramento complessivo delle performance del 2%.

6.3.2 Rete neurale

Come abbiamo già visto il predittore dell'esperto è una rete neurale. Sebbene sia possibile scegliere dei predittori arbitrariamente complessi, è stato scelto di mantenere semplice la struttura del singolo esperto. A parte i motivi di complessità computazionale, la scelta deriva soprattutto dall'assunzione che un funzionamento corretto delle guardie debba fornire agli esperti un sottospazio relativamente semplice da trattare.



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

Figura 8: Rete Neurale dell'esperto

E' stato scelto di implementare i classificatori con delle reti neurali MLP con un solo strato nascosto formato da un numero di neuroni dipendente dalla dimensione di I_g secondo la relazione approssimata: $nh = 15 + \left| I_g \right| / 500$. In pratica la rete neurale base ha 15 neuroni e viene aggiunto un neurone in più ogni 500 input

riconosciuti dalla guardia. Infatti maggiore è il numero di input da trattare nel training set maggiore dovrà essere la memoria della rete, avendo mantenuti costanti il numero di neuroni di ingresso e di uscita.

La rete ha in ingresso una finestra di 15 residui per cui essendo ogni residuo codificato con 21 numeri secondo la codifica spiegata precedentemente, il numero di neuroni di ingresso sarà 315 (= 21×15) .

I neuroni di uscita sono sempre 3 dove ognuno rappresenta la propensione dell'input ad assumere una conformazione alfa/beta/coil secondo la nota codifica one-shot (001/010/100).

In Figura 8 è mostrata la rete neurale dell'esperto in cui è messo in evidenza che ogni residuo è codificato con un pattern di 21 elementi. Il pattern è generato dalla matrice ottenuto combinando le informazioni dell'allineamento multiplo con quelle della matrice di sostituzione come spiegato precedentemente.

6.4 Risultati sperimentali

Per valutare le prestazioni del predittore basato sull'architettura a esperti multipli e per facilitare i paragoni con altri sistemi, sono stati effettuati esperimenti su dei dataset noti in letteratura.

Il dataset di test RS126, derivato da il lavoro di Rost [Rost93], è composto da 126 proteine e 23.363 aminoacidi. Il dataset di training T [Pollastri02] è derivato da una selezione del database di strutture terziarie PDB rimuovendo le proteine con meno di 30 aminoacidi o con una risoluzione di meno di 2.5Å. Il dataset risultante è ridotto ulteriormente eliminando le sequenze con più del 50% di similarità. Inoltre sono state rimosse anche le sequenze con similarità maggiore del 25% rispetto a quelle presenti nel test set. Il risultante dataset contiene 1180 sequenze e 282.303 aminoacidi (input).

Le guardie sono basate su un insieme di feature di tipo chimico-biologico non necessariamente discriminanti per le classi strutturali. Infatti ricordiamo che lo scopo delle metriche non è quello di separare lo spazio in sottoinsiemi omogenei per la classe, ma di selezionare input facilmente classificabili per la rete neurale. Sebbene feature con queste caratteristiche possono essere selezionate anche a

partire da regole arbitrarie, risulta naturale aspettarsi che informazioni di tipo biologico possano essere più adatte in quanto sono in grado di riconoscere dei fenomeni chimici che altrimenti resterebbero nascosti al sistema. La lista delle feature utilizzate è mostrata in Figura 9. Per ogni feature mostrata in figura vengono generate due regole nel seguente modo: indicato con $m(a,b)$ il valore della feature media calcolata sulla finestra di residui (a,b) vengono considerate le seguenti regole:

- $m(-w/2, w/2) > th$
- $m(-w,0) > m(0,w)$

La prima controlla se il valore della feature valutata nella finestra centrata nel residuo è maggiore di una soglia. La seconda controlla se il valore della feature valutata nella finestra a sinistra del punto centrale è maggiore del valore della stessa feature a destra. Di conseguenza le 8 feature mostrate in figura generano guardie di dimensione 16.

Feature
idrofobicità
carica
dimensione
polarità
ionizzabilità
indice di propensione alpha
indice di propensione beta
indice di propensione coil

Figura 9: Feature per le guardie

Gli esperimenti sono stati eseguiti facendo evolvere una popolazione iniziale di 500 esperti generati in maniera casuale con l'unico vincolo che il fattore di copertura minimo di ogni esperto sia il 5% del dataset in modo che la dimensione media del match-set sia di 20 esperti.

Ogni rete neurale è addestrata con un 10 epoche dell'algoritmo di back-propagation utilizzando un learning rate di 0.05.

Lo scopo degli esperimenti è quello di valutare la dipendenza delle performance dalla quantità di conoscenza del dominio inserita nel sistema e l'impatto dell'utilizzo di un sistema ad esperti multipli ottimizzato geneticamente.

Il primo esperimento consiste nel valutare il comportamento di una popolazione non ancora addestrata geneticamente. E' stata generata una popolazione di 500 esperti casuali ed è stato eseguito l'addestramento soli dei classificatori singoli. Senza eseguire nessun altra ottimizzazione il sistema è stato testato nel dataset. Ripetendo l'esperimento per 1000 volte si è ottenuto un risultato medio del 70% di precisione che è di poco superiore a quello di un esperto con visibilità globale (69%). Questo incremento è dovuto alla diversità degli esperti che si viene a creare grazie al fatto che sono tutti addestrati su input differenti.

Un secondo esperimento mette in evidenza il miglioramento ottenuto con la codifica MAS. Vengono create due popolazioni identiche di 500 esperti casuali e vengono fatti evolvere con il genetico (5000 step) in maniera indipendente. La prima popolazione utilizza una codifica MA mentre la seconda una codifica MAS con una matrice BLOSUM62. I risultati mostrano che l'algoritmo genetico riesce a migliorare le performance del 5% (74%) rispetto al caso monolitico e si ha un ulteriore incremento di circa 3% dovuto al sistema di codifica MAS (76.8%).

I risultati ottenuti (Figura 10) sono stati confrontati con quelli di altri sistemi noti in letteratura secondo le linee guida di [cuff99]. In particolare sono stati testati i programmi NNSSP, PHD, DSC, PREDATOR e SSPRO sul dataset RS126. I risultati mostrano che con la codifica MAS le prestazioni (76.8%) sono paragonabili allo stato dell'arte attuale.

Programma	Q3
PREDATOR	70.3
DSC	71.1
NNSSP	72.7
PHD	73.5
MASSP (MA)	74.1
MASSP (MAS)	**76.8**
SSPRO	77.0

Figura 10: Risultati sperimentali. Percentuale di input classificati correttametne (Q3)

Conclusioni

Uno dei problemi principali in bioinformatica è l'analisi e la caratterizzazione di sequenze biologiche sia da un punto di vista strutturale che funzionale. Sono stati presentati e discussi due importanti rami di ricerca, quello della comparazione multipla di sequenze proteiche e quello della predizione di strutture secondarie proteiche.

Per il primo aspetto è stato presentato un algoritmo di comparazione multipla che utilizza informazioni strutturali delle sequenze e una tecnica di astrazione per guidare la ricerca della configurazione ottimale.

Per il secondo sono state studiate architetture ad esperti multipli per la predizione di strutture secondarie. In particolare l'architettura NXCS permette di sfruttare delle regole legate al dominio per partizionare automaticamente lo spazio degli input con un algoritmo XCS e nello stesso tempo di effettuare una classificazione con il meccanismo della combinazione di classificatori multipli.

I sistemi proposti in entrambi i casi hanno permesso di raggiungere risultati paragonabili allo stato dell'arte attuale e in certi casi superiori.

Bibliografia

[Altschul90] Stephen F. Altschul, Warren Gish, Webb Miller,Eugene W. Myers, David J. Lipman, *"Basic Local Alignment Search Tool"*, 1990.

[Altschul97] Altschul, S.F., Madden, TL, Schaffer, AA, Zhang, J, Zhang, Z, Miller, W & Lipman, DJ. *"Gapped BLAST and PSI-BLAST: a new generation of protein database search programs"*. Nucl. Acids Res., 1997

[Anfinsen73] Anfinsen, C.B.: Principles that govern the folding of protein chains. Science. (1973) 181:223–230

[Bairoch00] Bairoch A., Apweiler R.: The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. Nucleic Acids Res. (2000) 28:45–48

[Bairoch98] Kay Hofman, Philipp Bucher, Laurent Falquet, Amos Bairoch, "The PROSITE database, its status in 1999", 1998.

[Berman00] Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, N., Weissig, H., Shindyalov, I.N., Bourne, P.E.: The Protein Data Bank. Nucleic Acids Research. (2000) 28:235–242

[Blundell93] Blundell, T.L., Johnson, M.S.: Catching a common fold. Prot. Sci. (1993) 2(6):877–883

[Boczko96] Boczko, E.M., Brooks, C.L.: First-principles calculation of the folding free energy of a three-helix bundle protein. Science. (1995) 269(5222):393–396

[Bowie91] Bowie, J.U., Clarke, N.D., Pabo, C.O. Sauer, R.T.: Identification of protein folds: matching hydrophobicity patterns of sequence sets with solvent accessibility patterns of known structures. Proteins: Struct., Funct., Genet. (1990) 7:257–264

[Breiman84] Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth, Belmont, CA (1984)

[Breiman96a] Breiman, L.: Stacked Regressions. Machine Learning, 24 (1996) 41-48

[Breiman96b] Breiman, L.: Bias, Variance, and Arcing Classifiers. Technical Report n. 460, Statistics Dept., Univ. of California at Berkeley, CA (1996)

[Carrillo88] H. Carrillo, D. Lipman, The Multiple Sequence Alignment Problem in Biology. SIAM J Appl Math 1988;48(5):1073-1082

[Chothia86] Chothia, C.: Proteins – 1000 families for the molecular biologist. Nature (1992) 357:543–544.

[Chothia92] Chotia, C.: One thousand families for the molecular biologist. Nature (1992) 357:543–544

[Chou74] Chou, P. Y. & Fasman, U. D. (1974). Prediction of protein conformation. Biochem., 13, 211-215.

[Clark89] Clark, P., Niblett, T.: The CN2 Induction Algorithm. Machine Learning 3(4) Kluwer, (1989) 261-283

[Covell92] Covell, D.G.: Folding protein alpha-carbon chains into compact forms by Monte Carlo methods. Proteins (1992) 14:409–420

[Cuff99] Cuff, J.A., Barton, G.J.: Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. PROTEINS: Structure, Function and Genetics (1999) 34:508–519

[Dandekar94] Dandekar, T., Argos., P.: Folding the main chain of small proteins with the genetic algorithm. J. Mol. Biol., (1994) 236:844–861

[Dayhoff78] Dayhoff, M. O. 1978. Survey of new data and computer methods of analysis. In M. O. Dayhoff, ed., Atlas of Protein Sequence and Structure, vol. 5, supp. 3, pp. 29, National Biomedical Research Foundation, Silver Springs, Maryland.

[Domingos00] Domingos, P.: A Unified Bias-Variance Decomposition for Zero-One and Squared Loss. Proc. of the Seventeenth National Conference on Artificial Intelligence, Austin, Texas (2000) 564-569

[Freund97] Freund, Y., Schapire, R. E.: A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. Journal of Computer Science and System Sciences, 55(1) (1997) 119-139

[Frishman96] Frishman D, Argos P (1996). Incorporation of non-local interactions in protein secondary structure prediction from the amino acid sequence. Protein Eng, 9(2):133-142.

[Frishman97] Frishman, D. and Argos, P. (1997) 75% accuracy in protein secondary structure prediction. Proteins, 27, 329-335

[Garnier78] Garnier, J., Osguthorpe, D. J. & Robson, B. (1978). Analysis of the accuracy and Implications of simple methods for predicting the secondary structure of globular proteins. J. Mol. Biol., 120, 97-120.

[Garnier87] Gibrat, J.-F., Garnier, J. & Robson, B. (1987). Further developments of protein secondary structure prediction using information theory. New parameters and consideration of residue pairs. J. Mol. Biol., 198, 425-443.

[Gething92] Gething, M.J., Sambrook, J.: Protein folding in the cell Nature (1992) 355:33–45

[Giunchiglia97] Giunchiglia, F., Villafiorita, A., Walsh, T.: Theories of Abtraction. AI Communications (1997) 10:167-176

[Goldberg89] Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addi-son-Wesley (1989)

[Greer90] Greer, J.: Comparative modelling methods: application to the family of the mammalian serine proteases. Proteins (1990) 7:317–334

[Hartl94] Hartl, F.U.: Secrets of a double-doughnut. Nature (1994) 371:557–559.

[Havel93] Havel, T.F.: Predicting the structure of the avodoxin from Eschericia coli by homology modeling, distance geometry and molecular dynamics. Mol. Simulation, (1993) 10:175–210

[Henikoff92] Steven Henikoff, Jorja G. Henikoff “Amino acid substitution matrices from protein blocks”, 1992

[Heringa99] Heringa, J.: Two strategies for sequence comparison: profile preprocessed and secondary structure-induced multiple alignment. Computers and Chemistry (1999) 23:341-364.

[Hinds94] Hinds, D.A., Levitt, M.: Exploring conformational space with a simple lattice model for protein structure. J. Mol. Biol. (1994) 243:668–682

[Holland75] Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, (1975)

[Holland76] Holland, J.H.: Adaption. In: R.Rosen and F.M. Snell (eds.): Progress in Theoretical Biology 4, New York (1976)

[Holland86] Holland, J.H.: Escaping Brittleness: The possibilities of General-Purpose Learning Algo-rithms Applied to Parallel Rule-Based Systems. In: R.S. Michalski, J. Carbonell, M. Mitchell (eds.): Machine Learning II, Morgan Kaufmann (1986) 593-623

[Holley89] Holley, H. L. & Karplus, M. (1989). Protein secondary structure prediction with a neural network. Proc. Natl. Acad. Sc. U.S.A., 86, 152-156.

[Jacobs91] Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive Mixtures of Local Ex-perts. Neural Computation, 3 (1991) 79-87

[Jacobs92] Jordan, M.I., Jacobs, R.A.: Hierarchies of Adaptive Experts. In Advances in Neural Information Processing Systems 4, J. Moody, S. Hanson, and R. Lippman (eds.) Morgan Kauf-mann (1992) 985-993

[Jones99] Jones, D.T.: Protein secondary structure prediction based on position-specific scoring matrices. J. Mol. Biol. (1999) 292:195–202.

[Kanehisa88] Kanehisa, M. (1988). A multivariate analysis method for discriminating protein secondary structural segments. Prot. Engin., 2, 87-92

[King96] King RD, and Sternberg MJ (1996). Identification and application of the concepts important for accurate and reliable protein secondary structure prediction. Protein Sci, 5(11):2298-310.

[Knoblock91] Knoblock, C.A., Tenenberg, J.D., Yang, Q.: Characterizing Abstraction Hierarchies for Planning. Proc. of the Ninth National Conference on Artificial Intelligence (1991) 2:692-697

[Kovacs99] Kovacs, T.: Strength or Accuracy? A Comparison of Two Approaches to Fitness. Second Int. Workshop on Learning Classifier Systems during GECCO99 (1999)

[Krogh95] Krogh, A., Vedelsby, J.: Neural Network Ensembles, Cross Validation, and Active Learning. In G. Tesauro, D. Touretzky, and T. Leen, (eds.) Advances in Neural Information Processing Systems, MIT Press, 7 (1995)

[Krogh98] Anders Krogh, “An Introduction to Hidden Markov Models for Biological Sequences” (1998)

[Lander93] Yi, T.-M. & Lander, E. S. (1993). Protein Secondary Structure Prediction Using Nearest-neigbor Methods. J. Mol. Biol., 232, 1117-1129.

[Lanzi98] Lanzi, P.L.: Adding memory to XCS. Proc. of the IEEE Conference on Evolutionary Computation (ICEC98) (1998)

[Lanzi99] Lanzi, P.L., Perrucci, A.: Extending the Representation of Classifier Conditions Part II: From Messy Coding to S-Expressions. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '99) I, Morgan Kaufmann (1999) 345-353

[Levin86] Levin JM, Robson B, and Garnier J (1986). An algorithm for secondary structure determination in proteins based on sequence similarity. FEBS Lett, 205(2):303-308.

[Levitt76] Levitt, M.: A simplified representation of protein conformations for rapid simulation of protein folding. J. Mol. Biol. (1976) 104:59–107

[Levitt83] Levitt, M.: Protein folding by constrained energy minimization and molecular dynamics. J. Mol. Biol. (1983) 170:723–764

[Lim74] Lim, V. I. (1974). Structural Principles of the Globular Organization of Protein Chains. A Stereochemical Theory of Globular Protein Secondary Structure. J. Mol. Biol., 88, 857-872

[Mitchell92] Mitchell, E. M., Artymiuk, P. J., Rice, D. W. & Willett, P. (1992). Use of techniques derived from graph theory to compare secondary structure motifs in proteins. J. Mol. Biol., 212, 151-166.

[Morgenstern96] Burkhard Morgenstern, Andreas Dress, Thomas Werner, "Multiple DNA and protein sequence alignment based on segment-to-segment comparison", 1996.

[Needle70] Needleman, SB & Wunsch, CD, "A general method applicable to the search for similarities in the amino acid sequence of two proteins". J. Mol. Biol., 1970.

[Notredame00] Cédric Notredame, Desmond G. Higgins, Jaap Heringa, "T-Coffee: A Novel Method for Fast and Accurate Multiple Sequence Alignment" 2000.

[Notredame96] C. Notredame, D.G. Higgins, SAGA: Sequence Alignment by Genetic Algorithm, Nucleic Acid Research, Vol. 24, 1515-1524, 1996.

[Notredame98] Cédric Notredame, Liisa Holm, Desmond G. Higgins, "COFFEE: an objective function for multiple sequence alignments", 1998.

[Orengo94] Orengo, C.A., Jones, D.T., Thornton, J.M.: Protein superfamilies and domain superfolds. Nature (1994) 372:631–634

[Pearson88] Pearson W.R. and Lipman D.J. (1988): Improved tools for biological sequence comparison. Proc. Natl. Acad. Sci. USA 85, 2444-2448.

[Plaisted81] Plaisted, D.: Theorem Proving with Abstraction. Artificial Intelligence (1981) 16(1):47-108

[Pollastri02] Pollastri, G., Przybylski, D., Rost, B., Baldi, P.: Improving the Prediction of Protein Secondary Structure in Three and Eight Classes Using Neural Networks and Profiles. Proteins (2002) 47:228–235

[Ptitsyn83] Ptitsyn, O. B. & Finkelstein, A. V. (1983). Theory of protein secondary structure and algorithm of its prediction. Biopolymers, 22, 15-25.

[Qian88] Qian, N. & Sejnowski, T. J. (1988). Predicting the secondary structure of globular proteins using neural network models. J. Mol. Biol., 202, 865-884.

[Riis96] Riis, S.K., Krogh, A.: Improving prediction of protein secondary structure using structured neural networks and multiple sequence alignments. J. Comp. Biol. 3 (1996), 163–183.

[Rivest87] Rivest, R.L.: Learning Decision Lists. Machine Learning 2(3) (1987) 229-246

[Robson76] Robson, B. (1976). Conformational properties of amino acid residues in globular proteins. J. Mol. Biol., 107, 327-56

[Rost93] Rost, B., Sander, C.: Prediction of protein secondary structure at better than 70% accuracy. J Mol Biol (1993) 232:584-599.

[Rost99] Rost, B.: Twilight zone of protein sequence alignments. Protein Engineering (1999) 12(2):85-94.

[Roterman89] Roterman, I.K., Lambert, M.H., Gibson, K.D., Scheraga, H.A.: A comparison of the charmm, amber and ecepp potentials for peptides. ii. phi-psi maps for n-acetyl alanine n'-methyl amide: comparisons, contrasts and simple experimental tests. J.Biomol. Struct. Dynamics, (1989) 7:421–453

[Sacerdoti74] E. Sacerdoti, "Planning in a hierarchy of abstraction spaces," Artificial Intelligence, Vol. 5, No. 2, 115 - 135, 1974.

[Saitta98] Saitta, L., Zucker, J.D.: Semantic Abstraction for Concept Representation and Learning. Symposium on Abstraction, Reformulation and Approximation (SARA98), Pacific Grove, California (1998) 103-120

[Salamov94] Solovyev, V. V. & Salamov, A. A. (1994). Predicting a-helix and b-strand segments of globular proteins. CABIOS, 10, 661-669.

[Salamov95] Salamov, A. A. & Solovyev, V. V. (1995). Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignment. J. Mol. Biol., 247, 11-15.

[Sali95] Sali, A.: Modelling mutations and homologous proteins. Curr. Opin. Biotech. (1995) 6:437-451

[Sanchez97] Sanchez R., Sali, A.: Advances in comparative protein-structure modeling. Curr. Opin. Struct. Biol. (1997) 7:206–214

[Schapire97] Schapire, R.E., Freund, Y., Bartlett, P., Lee, W.S.: Boosting the Margin: a New Explanation for the Effectiveness of Voting Methods. Proc. of the Fourteenth Int. Conference on Machine Learning (1997) 322-330

[Shapire99] Schapire, E.: A Brief Introduction to Boosting. Proc. of the Sixteenth Int. Joint Conference on Artificial Intelligence (1999)

[Skolnick90] Skolnick, J., Kolinski, A.: Simulations of the folding of a globular protein. Science, (1990) 250:1121–1125

[Taylor83] Taylor, W. R. & Thornton, J. M. (1983). Prediction of super-secondary structure in proteins. Nature, 301, 540-542.

[Thompson94] J. Thompson, D. G. Higgins, T. J. Gibson, "Clustal W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice", Nucleic Acids Res 1994.

[Thompson99b] Thompson, J.D., Plewniak, F., Poch, O.: BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. Bioinformatics (1999) 15:87-88

[Unger89] Unger, R., Harel, D., Wherland, S., Sussman, J.L.: A 3-D building blocks approach to analyzing and predicting structure of proteins. Proteins (1989) 5:355–373

[Vajda97] Vajda, S., Sippl, M., Novotny, J.: Empirical potentials and functions for protein folding and binding. Curr. Opin. Struct. Biol. (1997) 7:228–228

[Valiant84] Valiant L.: A Theory of the Learnable. Communications of the ACM, 27 (1984) 1134-1142

[Vapnik98] Vapnik, V. N.: Statistical Learning Theory. John Wiley and Sons Inc., New York (1998)

[Vere80] Vere, S.A.: Multilevel Counterfactuals for Generalizations of Relational Concepts and Pro-ductions. Artificial Intelligence 14(2) (1980) 139-164

[Weigend95] Weigend, A.S., Mangeas, M., Srivastava, A. N.: Nonlinear Gated Experts for Time Series: Discovering Regimes and Avoiding Overfitting. Int. Journal of Neural Systems, 6(1995) 373–399

[Wilson95] Wilson, S.W.: Classifier Fitness Based on Accuracy. Evolutionary Computation, 3(2) (1995) 149-175

[Wilson98] Wilson, S.W.: Generalization in the XCS Classifier System. Proc. of the Third Annual Genetic Programming Conference, San Francisco, CA, Morgan Kaufmann, (1998) 665-674.

[Wilson99] Wilson, S.W.: Get Real! XCS with Continuous-Valued Inputs. Festschrift in Honor of John H. Holland, L.Booker, S. Forrest, M. Mitchell, and R. Riolo (eds.). Center of Study of Com plex Systems, The University of Michigan, ANN Arbor, MI, May 15-18 (1999).

[Zvelebil87] Zvelebil, M. J., Barton, G. J., Taylor, W. R. & Sternberg, M. J. E. (1987). Prediction of protein secondary structure and active sites using alignment of homologous sequences. J. Mol. Biol., 195, 957-961.