Università degli Studi di Bologna

FACOLTÀ DI INGEGNERIA

Dottorato di Ricerca in Ingegneria Elettronica,
Informatica e delle Telecomunicazioni

XIX Ciclo

ING-INF/05

Estrazione e rappresentazione della conoscenza nella bioinformatica

Tesi di Dottorato di:

Dott. **Lorenzo Baldacci**

Relatore:

Chiar.mo Prof. Ing. **Dario Maio**

Coordinatore:

Chiar.mo Prof. Ing. **Paolo Bassi**

Anno Accademico 2005-2006

Introduzione

La proteomica è la disciplina che studia il comportamento delle proteine e, in particolare, si propone di comprendere quali delle loro caratteristiche sono coinvolte attivamente nelle funzioni dei processi biologici. Allo stato attuale è infatti noto solo in un numero limitato di casi quali insieme di funzioni biologiche sono associate a una specifica proteina.

A suddetto risultato è possibile lavorare svolgendo esperimenti in-vitro che tuttavia risultano essere molto costosi sia in termini economici che di tempo. Una strada alternativa è quella di sfruttare l'assunto per cui proteine con caratteristiche simili svolgono funzioni simili, ciò permette di inferire le funzionalità di una proteina in base alle similarità che essa presenta rispetto a strutture funzionalmente note. Il problema diventa quindi quello di evidenziare le similarità tra proteine; le tecniche di comparazione più diffuse sono oggi quella *sequenziale* e quella *strutturale* che fondano le proprie funzioni di similarità rispettivamente sulla sequenza di aminoacidi che compongono le proteine e sulla struttura tridimensionale che questi aminoacidi assumono.

Tuttavia l'evoluzione può portare mutazioni nelle sequenze, che si riflettono poi in modifiche topologiche della struttura, mantenendo comunque invariate le regioni di superficie che sono necessarie alla proteina per svolgere la sua funzione. In questi casi una comparazione di sequenza o di struttura potrebbe non essere sufficiente a identificare relazioni di tipo funzionale. Ciò a spinto i ricercatori a valutare una terza modalità di comparazione che si basa direttamente sulle caratteristiche della superficie proteica che, sebbene più difficile da modellare, è in ultima analisi la vera e unica artefice dell'interazione tra composti proteici.

Attualmente gli approcci alla comparazione di superfici proteiche utilizzano solo porzioni di superficie che sono già state annotate in quanto precedentemente riconosciute come *attive* nello svolgimento di specifiche funzioni biologiche. Queste tecniche, pur estremamente utili nella ricerca di proteine che presentano una stessa interfaccia nota, risultano inefficaci quando si tratta di comparare proteine non precedentemente annotate, oppure quando si voglia considerare ai fini della comparazione tutta la superficie e non limitarsi alle aree riconosciute come funzionalmente attive.

In questo lavoro di tesi vengono studiate le problematiche relative all'analisi della superficie proteica per la ricerca di similarità tramite un approccio non supervisionato. Il nostro approccio è composto da tre fasi: inizialmente le superfici

proteiche vengono suddivise in regioni aventi valori omogenei per proprietà chimiche (es. potenziale elettrico) e geometriche (es. curvatura); a partire da un database di superfici proteiche così clusterizzate sono poi estratte, mediante tecniche di data mining, pattern complessi che ricorrono in più proteine. Ogni pattern sarà caratterizzato, oltre che dalle proprietà fisico-geometriche delle regioni che lo compongono, anche dalla disposizione spaziale delle regioni. La similarità tra proteine è infine calcolata in base al numero e alla complessità dei pattern comuni.

Le relazioni di similarità ottenute possono essere utilizzate per eseguire una comparazione diretta di due proteine, oppure per classificarne un intero database. In tale direzione il nostro studio si è occupato di definire due algoritmi di classificazione utilizzati tra l'altro per verificare l'efficacia dell'approccio proposto che è stato applicato sia a database sintetici, sia a database di proteine reali.

Il lavoro è organizzato in nove capitoli, di cui, di seguito, verranno riassunti i contenuti.

I primi due capitoli introducono il dominio applicativo fornendo una panoramica delle problematiche biologiche ed indicando quali settori delle scienze del computer sono coinvolti. Questa discussione partirà da tematiche generali, toccando aree di ricerca molto discusse quali quella del genoma, per dirigersi verso argomenti più specializzati come quelli dell'analisi proteica. Ampio spazio sarà dato alla proteomica, area nella quale il lavoro di ricerca si inquadra, e a una descrizione approfondita dell'oggetto di ricerca, ossia le proteine. Infine saranno riassunte le aree di ricerca dell'analisi proteica inquadrando così definitivamente il lavoro di tesi.

Il terzo capitolo descrive nel suo complesso l'approccio metodologico utilizzato. Inizialmente verrà fornita una descrizione sullo stato dell'arte dell'analisi proteica superficiale e successivamente verrà dato uno schema del metodo proposto definendo così i passi fondamentali che saranno poi esplosi nei capitoli successivi.

Il capitolo quattro tratta il problema del clustering di superficie. In particolare verranno descritti i vincoli biologici che dovranno essere soddisfatti per ottenere regioni significative, per poi passare alla definizione formale della superficie proteica. Il capitolo introduce tre algoritmi di clustering di superficie e definisce un insieme di indici necessari alla comparazione e valutazione dei metodi proposti (omogeneità, regolarità e copertura).

Il quinto capitolo analizza la problematica della ricerca di pattern frequenti. Indica, anche in questo caso, quali siano i vincoli da introdurre per la specificità dell'applicazione. Infine viene introdotto un algoritmo di mining originale per la

ricerca di pattern complessi di superficie, in particolare sarà descritto il metodo utilizzato per valutare l'allineamento di pattern. Il capitolo conclude fornendo esempi di allineamento allo scopo di convalidare l'efficacia della metodologia proposta.

Nel sesto capitolo verrà introdotto il problema della classificazione di proteine e saranno riportati i due algoritmi implementati. Un primo semplice di tipo gerarchico implementato in quanto largamente utilizzato anche nel contesto biologico e ben comprensibile da esperti del dominio. Un secondo basato sulla tecnica del simulated annealing, e tutt'ora in via di valutazione, studiato allo scopo di fornire una classificazione in tempi ragionevoli e aggirare i problemi dati dalle scelte greedy del gerarchico. Sarà data particolare attenzione alla descrizione e formalizzazione delle funzioni obiettivo.

Nel settimo capitolo saranno fornite le descrizioni dei dataset utilizzati, i risultati e i commenti alle prove sperimentali. Questa fase di validazione dell'approccio è stata effettuata in maniera supervisionata da esperti del dominio.

L'ottavo capitolo fornisce la descrizione di un tool utilizzato e attualmente in via di sviluppo per la validazione degli algoritmi: PDBVision. Verranno descritte le funzionalità richieste per facilitare l'analisi di superficie e avere un conforto visuale nella validazione dei risultati.

Nel capitolo nove sono presentate le conclusioni e vengono proposti alcuni sviluppi futuri per le ricerche.

1. Il dominio applicativo

1.1. La biologia computazionale

Nel corso dell'anno 2000 è stato dato l'annuncio del completamento del sequenziamento del genoma umano, i risultati sono stati pubblicati congiuntamente in due numeri speciali di Nature [NAT01] e Science [SCI01]. Si tratta di un fatto particolarmente importante non tanto per l'utilità attuale dei dati prodotti, che pure è molto elevata, quanto in prospettiva per le opportunità che comporta [PAL99].

Questa opinione è motivata dalla constatazione che HGP, Human Genoma Project, e Celera Genomics [w16] altro non hanno fatto se non "limitarsi" a ordinare circa tre miliardi di coppie di lettere estratte da un semplice alfabeto di quattro lettere. Tale alfabeto è infatti composto dai simboli A, T, C e G in rappresentanza delle basi azotate Adenina, Timina, Citosina e Guanina che compongono la molecola chiave del nostro patrimonio genetico, il DNA.

Come detto, tuttavia, tali dati sono importanti in prospettiva per le informazioni che contengono e per le implicazioni che queste hanno con numerose patologie tutt'oggi difficilmente curabili [NOW95]. I due passi logici citati, estrazione delle informazioni e connessione alle patologie, non sono però triviali, tanto è vero che i tempi stimati per il loro completamento sono equiparabili, se non superiori, a quelli utilizzati per raggiungere la "pietra miliare" dell'annuncio sopra menzionato [EMM00].

Un ragionamento basato solo sui tempi è però ingannevole in quanto le competenze richieste sono differenti. Effettuando un confronto utilizzando gli argomenti informatici, si possono individuare due problematiche di livello generale. La prima riguarda la pubblicazione di dati e comprende quindi tutti quegli aspetti che sono connessi con la creazione e la messa a disposizione agli utenti delle informazioni prodotte. La seconda problematica riguarda invece l'utilizzo dei dati al fine di estrarne conoscenza.

1.1.1. La raccolta dei dati

La raccolta di dati genomici è un processo in continua evoluzione. Recentemente sono state introdotte tecniche, per l'analisi e il sequenziamento molecolare, che hanno permesso un notevole incremento della velocità di estrazione di dati biologici [DEL98, w32]. Il loro effetto è stato tale che per i dati raccolti con queste tecniche è

stato coniato un nuovo termine: *HTD, High Throughput Data* [KUE99, BIT99, ADA91, BOG95, KEL99, SIE99]. Tale incremento è particolarmente evidente analizzando i dati relativi alle statistiche di crescita delle basi di dati primarie.

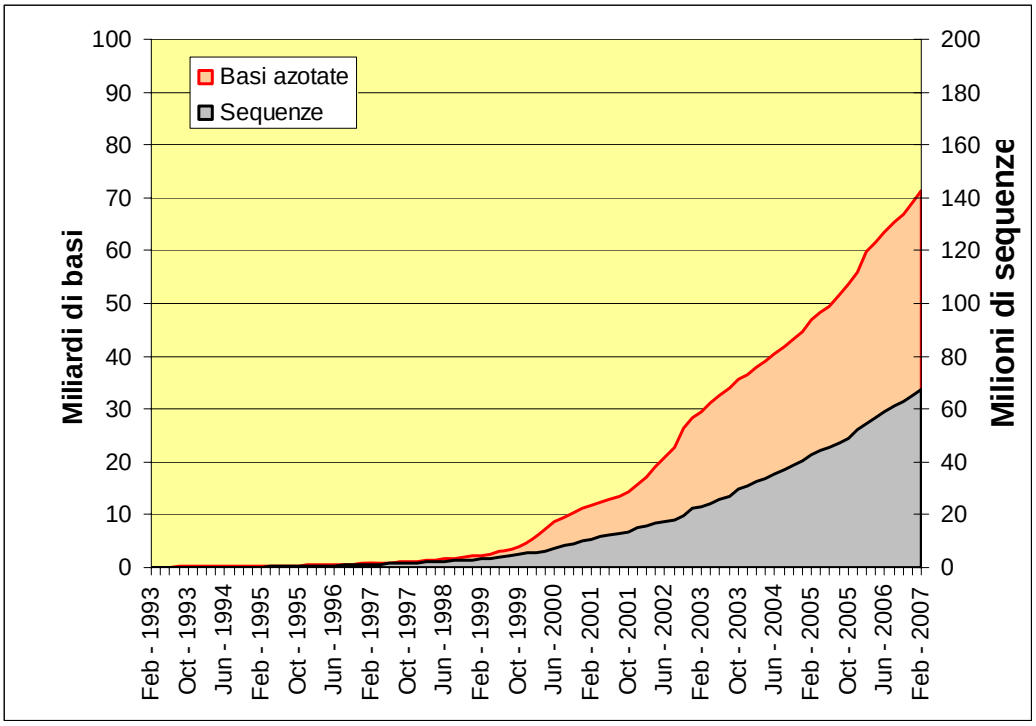

Fig. 1.1: Crescita della base di dati nucleotidica Genbank

Le banche dati di sequenze di acidi nucleici sono spesso definite Banche Dati Primarie in quanto contengono solo informazioni sulla sequenza delle basi azotate che costituiscono le molecole. Esistono anche basi di dati definite specializzate che associano alle sequenze anche informazioni funzionali delle molecole

Prendendo ad esempio la base di dati mantenuta da NCBI, National Center for Biotechnology Information [NUC00, WHE01], denominata Genbank si può notare come dalle 606 sequenze presenti nel 1982 si è passati a quasi 70 milioni nel febbraio del 2007 (fig. 1.1) [GRE00].

È evidente che il mantenimento di una simile mole di dati pone alcuni problemi in termini di efficienza sia per quanto riguarda l'accesso, sia per il modo in cui i dati devono essere organizzati. Non a caso le maggiori basi di dati, come ad esempio Genbank, sono suddivise in modo da fornire all'utente una prima chiave di selezione e da imporre nel contempo uno schema di organizzazione interna logicamente semplice. Le divisioni infatti in generale rappresentano le specie cui appartengono i dati mantenuti.

Per di più il problema del mantenimento dei dati relativamente alla prospettiva di una sola base di dati, non è che una minima parte del problema, infatti la situazione è più complessa. In generale ogni sorgente di dati, ad esempio un laboratorio di sequenziamento di medie dimensioni, mantiene una propria base di dati contente le sequenze prodotte. Queste possono confluire in una base di dati "generica" come la già citata Genbank, ma non vi sono assicurazioni sui tempi in cui questa operazione verrà eseguita e nemmeno sul fatto che ciò avvenga.

Il risultato di questa situazione è che attualmente le basi di dati contenenti informazioni relative al genoma sono circa 500 e crescono del 10% ogni anno. Ovviamente tale cifra comprende anche quelle basi di dati che nascono per specializzazione da un contenitore generico al fine di fornire informazioni più specifiche e più accurate.

Questa situazione è dovuta all'altra grande problematica del settore, l'estrazione di conoscenza, dove la situazione non può dirsi assestata [STA91, EDD96, EDD98, HAU98, PAR98, w20, w42] e dove in particolare ogni nuova informazione prodotta è molto importante. A tal punto che spesso vale la pena rendere disponibili i risultati ottenuti attraverso la creazione di una base di dati apposita. Ne sono esempi UniGene [w7, w8 e w13], COG [NUC00], Blocks [NUC00, w20], ProDom [NUC00] e molti altri [NUC00]. È facile intuire che esiste una notevole eterogeneità nel mantenimento dei dati che ovviamente si traduce in una moltitudine di scelte diverse: dai DBMS utilizzati, agli schemi delle basi di dati, al formato dei dati mantenuti [w14].

Un ulteriore problematica sorge allorquando si analizzino le competenze dei curatori delle basi di dati. Come detto infatti vi è una notevole libertà nella creazione degli stessi, quindi anche negli scopi che tali "contenitori" assumono, ma vi sono anche due ordini di problemi legati all'utilizzazione di un linguaggio non completamente standardizzato [SOB97, SMI98]. Si tratta di conflitti semantici derivanti dall'utilizzazione dei nomi, ossia oggetti diversi denominati ugualmente, oppure medesimi oggetti con più denominazioni in basi di dati diverse [ASH97, PEN99]. Queste situazioni sono possibili principalmente perché non vi è omogeneità nelle competenze dei soggetti coinvolti nella produzione dei dati e perché i progetti per una standardizzazione del linguaggio, come ad esempio Gene Ontology [w40], si trovano a fronteggiare una forte inerzia verso il cambiamento da parte dei curatori delle basi di dati che hanno sempre esitato ad effettuare delle variazioni nell'uso della terminologia, anche perchè tale variazione comporta la modifica dei contenuti delle basi di dati e quindi l'esposizione al rischio di inconsistenze interne.

1.1.2. L'estrazione di conoscenza

I dati raccolti dalle operazioni di sequenziamento, come detto, non sono importanti in quanto tali, ma in prospettiva per un loro utilizzo futuro [AND97, ALT98, JAN99, TAU96, FIS96, SEI96]. I tre miliardi di coppie di basi infatti sono di grande interesse perché in opportune sequenze assumono particolari significati. Tali successioni "dotate di senso" si chiamano geni e codificano ciascuno una proteina [w1]. Le proteine costituiscono il "materiale" basilare per il funzionamento di ogni essere vivente [NAT01, SCI01].

I passi logici finora espressi costituiscono il *dogma centrale della biologia molecolare* che può anche essere rappresentato dal seguente schema:

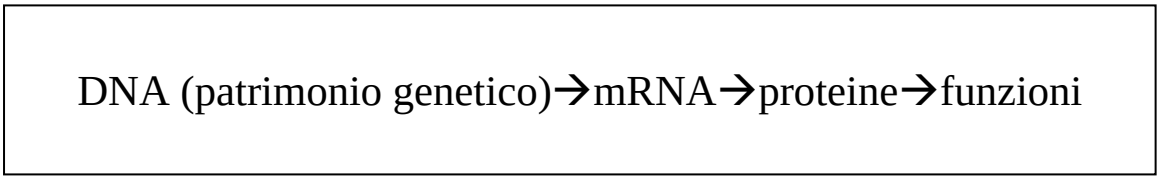

Fig. 1.2: Dogma centrale della biologia molecolare

dove il passaggio da DNA a mRNA è chiamato *trascrizione* mentre quello successivo da mRNA alla proteina è detto traduzione. Ancora una volta tuttavia l'effettiva implementazione dei passi logici comporta alcuni problemi.

Anche se il sequenziamento del genoma ha portato alla conoscenza della sequenza di basi azotate che costituiscono il patrimonio gnomico umano, ciò non significa che si conosca in maniera dettagliata come le informazioni sono portate da questa macromolecola. Infatti il DNA è composto da geni e da lunghe sequenze considerate, allo stato attuale, prive di informazioni. L'insieme dei geni rappresenta solo circa un decimo dell'intero DNA. Recenti studi hanno però riportato prove sperimentali che, contrariamente a quanto pensato finora, anche alcune zone considerate non significative, contribuiscono alla creazione di una proteina [LAB01].

Le problematiche sopra citate suggeriscono di studiare direttamente le proteine in quanto sono le dirette responsabili dello svolgimento delle funzioni cellulari. Da quest'ultima affermazione nasce un nuovo campo di applicazioni: la *proteomica* [EIS00, PAN00, SMI00, NAT01, KAH95, GIB96, KOO98], ovvero la disciplina che studia le proteine ed in particolare i loro effetti [WIS00, WAN00, SAN00], le relazioni tra loro. L'approccio introdotto dalla proteomica quindi mira a studiare

direttamente queste macromolecole [HOL94] operando così su informazioni equivalenti, la sequenza proteica è infatti codificata dai geni.

Un ulteriore vantaggio è dato dal fatto che le proteine prodotte sono differenti in base alla posizione della cellula nell'organismo, mentre il patrimonio genetico è unico per ogni soggetto, se ovviamente si escludono fenomeni di mutazione. Le ricerche in vivo utilizzando le proteine sono in quest'ottica più semplici in quanto lo spazio delle soluzioni è di dimensioni inferiori.

Un campo applicativo che trae enormi vantaggi dalla ricerca nel campo della proteomica è data dalla farmacogenomica [ROS00, SAV00, REE00] che ha come obiettivo la creazione di farmaci tagliati su misura dell'individuo cui sono rivolti. Conoscere infatti il codice permette di capire dove sono situati gli errori e soprattutto permette di agire, attraverso farmaci appositamente studiati, per porvi rimedio. Si tratta di un approccio completamente nuovo che prescinde dall'utilizzo di farmaci sintomatici, ovvero che si dirigono verso l'eliminazione dei sintomi indesiderati, e permette invece di creare farmaci curativi, ovvero che eliminano la causa

1.2. La bioinformatica

L'informatica è presente pesantemente in entrambe le fasi citate, cioè nella pubblicazione dei dati e nell'estrazione di conoscenza, velocizzando o rendendo addirittura possibili alcuni processi. Non si tratta tuttavia di una semplice informatizzazione o di un banale affiancamento [KRE00]. L'interazione tra il ricercatore e lo strumento informatico è infatti molto forte, al punto tale che agli esperimenti in-vitro e in-vivo si sono aggiunti quelli in-silico. A ulteriore testimonianza del forte legame ormai connaturato con la professione del moderno biologo vi è inoltre il fatto che la dicitura stessa è stata cambiata da semplice informatica in bioinformatica [BOG94, BEN96, AND97, GER97, SOB97, ALT98, SPE00, SMI00].

Quest'ultimo termine indica un campo di ricerca molto vasto e dai confini non ben delineati [w23, w28], come è anche dimostrato dal fatto che non esiste una definizione unica per questa disciplina [SAV00, w26]. Ecco infatti una serie ristretta di possibili definizioni:

- *Bioinformatica* è la scienza che sviluppa basi di dati e algoritmi con lo scopo di rendere più veloce e di sviluppare la ricerca in campo biologico [w26].

- *Bioinformatica* è una combinazione di Ingegneria del Software, Information Technology e genetica per estrarre e analizzare informazioni connesse con l'ambito della gnomica [w2].
- *Bioinformatica* è costituita dalla scienza e dalla tecnologia relative all'apprendimento, alla gestione e all'analisi delle informazioni biologiche.
- *Bioinformatica* è la disciplina scientifica che comprende tutti gli aspetti relativi all'acquisizione di informazioni di natura biologica, l'elaborazione, l'analisi, il mantenimento e la pubblicazione delle stesse oltre ovviamente all'interpretazione dei dati. Per far questo combina tecniche e strumenti sviluppati da varie discipline tra le quali la matematica, l'informatica e la biologia [BEN96].
- La *bioinformatica* è una disciplina basata sulla conoscenza. La risorsa chiave è la conoscenza e la tecnologia chiave è la gestione dell'informazione [BAK98].

Si possono tuttavia individuare tre principali aree [REE00] che costituiscono e caratterizzano questa scienza [GRU00]. Esse riguardano:

- Lo sviluppo di applicativi di supporto agli esperimenti di laboratorio [OLS95, SCH96, MAR98, DEL98, w30]. Si tratta di strumenti che permettano di automatizzare procedure di ricostruzione di sequenze, tipiche nella tecnica shotgun, ma anche di strumenti che permettano di gestire dati con un inferiore livello di dettaglio, come ad esempio il posizionamento dei marker sulle mappe genetiche [SCH97, SCH98]. In quest'area sono inoltre inclusi tutti quei software che permettono di gestire le informazioni derivanti da tecniche di acquisizione dei dati diversi dalle sequenze come ad esempio i microarray [LOC00, MAR99] e i radiation hybrid [STE97, NEW98, w34].
- La progettazione, l'implementazione e l'integrazione di basi di dati. È ipotizzabile che anche in futuro il trend di crescita del numero di basi di dati sarà positivo. E questo in ragione principalmente di due motivazioni. La prima riguarda il fatto che i genomi completamente sequenziati sono in numero limitato rispetto a quelli in cui i lavori sono in corso, la seconda proviene dalla necessità di annotare i risultati di analisi effettuate sui database attuali. Ne sono un esempio quelle che contengono sequenze riconosciute come quelle attive in particolari funzioni biologiche come ProDom, Prosite [HOF99], Prints-s [NUC00] e Blocks.

- Tecniche e applicativi per il data-mining [GSU98, CKC99, BOU00, JRF00, GRB99, w24]. Quest'area riguarda la seconda fase del progetto genoma ed è connaturata con l'estrazione di informazione/conoscenza dai dati prodotti [RAS97, w27, w30, w38]. Anche se è stata presentata sotto il singolo nome di data-mining quest'area è molto eterogenea comprendendo divisioni che in comune hanno solo lo scopo di estrarre conoscenza ed il fatto di affrontare problematiche simili.

Discipline coinvolte	***Risorse dati***	***Applicazioni bioinformatiche***
Algoritmi e strutture dati Grafica Elaborazione dei segnali Architetture hardware Intelligenza artificiale DBMS Statistica Simulazioni Teoria dell'informazione Elaborazione d'immagini Robotica Ingegneria del software	Database comunitari Sistemi informativi dei laboratori	Acquisizione dati Determinazione di strutture Modellazione molecolare Simulazione molecolare Progettazione strutturale di farmaci Allineamento strutturale Comparazione strutturale Predizione struttrale Allineamento sequenziale Evoluzione molecolare Annotazione di geni

Fig. 1.3: Bioinformatica: fondamenta e applicazioni

Una visione riassuntiva e schematica alternativa è stata presentata da David Bentos [BEN96] e comprende le fondamenta, in termini di aree disciplinari nell'ambito dell'informatica, su cui si basa la bioinformatica, le possibili sorgenti dei dati e le principali aree applicative presenti (fig. 1.3).

1.2.1. Problematiche generali

Nell'ambito della bioinformatica sono presenti numerose aree di ricerca [BEN96, GER97] ed alcune sono già state in parte evidenziate. Nell'analisi che seguirà non verranno prese tutte in considerazione, ma le si valuteranno tenendo presente l'ambito informatico come prevalente.

Si è già accennato al fatto che esistono un gran numero di fonti di dati caratterizzate da una notevole libertà dei parametri che le contraddistinguono.

Questa situazione ha notevoli conseguenze allorquando il compito principale sia l'estrazione di informazioni in quanto si traduce in una molteplicità di scelte possibili. A questo primo ordine di problemi bisogna però aggiungerne almeno altri due aventi valenze generali dati dal trade-off tra tempi e qualità dei risultati della ricerca e dalla forte eterogeneità del settore. Nonostante la presentazione delle stesse avvenga distinguendole e analizzandole in modo separato, è necessario evidenziare fin d'ora il fatto che tali problematiche interagiscono pesantemente rendendo più complicato il farvi fronte.

1.2.2. Accessibilità dei dati

Come già accennato in 1.2, il numero delle basi di dati mantenenti dati di valenza biologica ed in particolare connessi con l'ambito dei vari progetti legati al genoma è molto elevato. L'ordine di grandezza tuttavia non è il solo problema legato a questa moltitudine di possibili fonti di dati. Esistono, infatti, ulteriori fattori da tenere in considerazione prima di iniziare un qualsiasi percorso di esplorazione alla ricerca dei dati desiderati. È necessario in particolare soffermarsi sui contenuti delle basi di dati stesse, sulle relazioni che intercorrono al loro interno e su quelle invece tra basi di dati diverse.

Una peculiarità del settore che non è assolutamente ovvia è data dal fatto che le basi di dati presenti hanno, in generale, intersezione non nulla tra loro, anche se nessuna è strettamente contenuta nelle altre. Questa situazione è dovuta all'origine stessa dei dati. Spesso le basi di dati hanno avuto origine dalla necessità di mantenere in un unico luogo i risultati di alcuni esperimenti effettuati tramite differenti tecniche. A questo bisogna aggiungere il fatto che i dati di origine biologica sono estremamente complessi e pertanto difficilmente una sola tecnica riesce a catturarne completamente i dettagli. Da tutto ciò si evince che basi di dati differenti sovente mantengono solo una parte dell'intero oggetto e quindi il ricercatore deve operare utilizzando query multi-database al fine di ricostruire quanto necessita trovandosi a fronteggiare tutte quelle problematiche associate ai database distribuiti.

Una seconda peculiarità dei contenuti riguarda il fatto che alcune basi di dati non mantengono semplicemente dei dati prodotti attraverso esperimenti in-vitro e in-vivo, ma anche provenienti da esperimenti in-silico. La conseguenza è che tali basi di dati derivano in parte o totalmente da altre. Ne sono casi esemplari le basi di dati contenenti cluster come UniGene oppure COG, ma anche quelle proteiche che

possono derivare da quelle nucleotidiche in ragione di una semplice traduzione, come TR-EMBL.

1.2.3. Efficienza ed efficacia dei metodi

Nell'analizzare dati complessi come quelli biologici risulta necessario stabilire un giusto compromesso tra il tempo di calcolo e la qualità dei risultati. Il tempo di calcolo risulta essere un punto critico in quanto gli algoritmi sono implementati per risolvere problematiche complesse [PEA90, KBH99, ALT97, STA91, HEN92, KAR93] e la quantità di dati da elaborare risulta essere enorme. Se pensiamo di dover elaborare dati relativi al patrimonio genetico del genere umano, le basi azotate da processare sono oltre i tre miliardi.

L'altro aspetto contrapposto è costituito dalla qualità dei risultati ritrovati con l'operazione di ricerca. Spesso per portare soluzioni ai problemi in tempi accettabili è necessario affidarsi ad euristiche piuttosto che utilizzare algoritmi esatti di ricerca. In generale si tende a lasciare all'utente differenti possibilità di analisi dei dati, in modo che esso possa scegliere se privilegiare i tempi di calcolo alla qualità o viceversa. Non è raro infatti trovare diversi applicativi per analizzare dati provenienti dallo stesso database come avviene per il Blast [ALT90, ALT97, ZHA98] e il FastA [PEA90].

1.2.4. Eterogeneità

Un'ulteriore problematica riguarda un argomento classico dell'informatica allorquando si affronti un'operazione di integrazione dati provenienti da progetti sviluppati in modo indipendente. Nel caso specifico l'eterogeneità è accresciuta dall'elevato numero di progetti che solitamente necessitano di integrazione [ASH97].

Per ciò che riguarda la connessione alle diverse fonti di dati spesso accade che ogni base di dati ha una sua interfaccia specifica di interrogazione che non è riutilizzabile per altre fonti di dati e che dipende dalla versione del prodotto, dal produttore dello stesso e dal tipo di base di dati utilizzata (relazionale [HAG98, SKU99], object oriented oppure di tipo legacy [NUC00]). Inoltre non sempre tali sorgenti sono mantenute utilizzando DBMS, ma spesso il sistema è basato su filesystem accessibili tramite semplici server FTP, in questi il formato utilizzato per mantenere le informazioni, ovvero semplici flat files [BAK98], introduce un ulteriore grado di eterogeneità.

La soluzione di memorizzazione tramite file di testo veniva adottata quando il numero di dati era limitato e la cosa più importante era che questi fossero facilmente capibili dal lettore umano, in pratica che fossero intelligibili.

Si evince quindi che il nome comune assegnato al formato non implica minimamente una struttura sottostante simile. Ogni base di dati ha, infatti, il suo proprio tipo di flat file, ovvero di ordinamento interno al file di testo dei dati [STO01, APW01, w6, w15, w17, w33]. Inoltre, anche all'interno di una medesima base di dati vi possono essere più formati, ad esempio PIR, Protein Information Resource, ne contiene quattro corrispondenti ai maggiori cambiamenti accorsi alla base di dati stessa durante la sua evoluzione.

Infine è indispensabile citare le problematiche di eterogeneità sulla nomenclatura utilizzata negli schemi delle basi di dati e sull'utilizzo improprio dei campi contenenti informazioni biologiche. Questi problemi provengono dal fatto che la bioinformatica è in continua evoluzione e che i progetti di ricerca in quest'ambito sono fortemente pluridisciplinari. La veloce evoluzione e la pluridisciplinarità si riflettono nell'utilizzo di linguaggi estremamente differenti e in continua crescita. Spesso tutto questo porta all'utilizzo di terminologie uguali per identificare oggetti/concetti differenti o viceversa. Allo scopo di unificare i linguaggi in questa disciplina sono nati dei progetti per la definizione di ontologie come il Gene Ontology [w40].

2.Le proteine e l’analisi proteica

Il progetto genoma si è concluso con successo e in anticipo sui tempi previsti. Ma ora che è stata costruita la mappa dei circa 30.000 geni che compongono il DNA umano, risulta necessario comprendere il loro significato. Si sa che i geni, tramite l'acido ribonucleico, servono per codificare le proteine. Ecco dunque che in questi ultimi anni è stato proposto un concetto nuovo: quello di proteoma, ossia l'insieme delle proteine corrispondenti a un genoma [w4]; e una nuova disciplina ha mosso i primi passi: la proteomica appunto, che si affianca alla ricerca genomica nel tentativo di scoprire il funzionamento della cellula e della vita. Si può definire come proteomica l'analisi sistematica di tutte le sequenze proteiche e delle modalità con cui le proteine interagiscono tra loro e nei tessuti. Si tratta di isolare, separare, identificare e determinare la funzione di tutte le proteine di un organismo. Dato che nell'uomo se ne stimano mezzo milione, l'impresa non si presenta tra le più facili.

Quella delle proteine non è una conoscenza teorica: le proteine servono per trasmettere messaggi, costruire i tessuti biologici e ripararli, consentire lo svolgimento di praticamente tutte le funzioni essenziali per la vita della cellula. Quando questo processo si inceppa in qualche punto, si hanno manifestazioni patologiche e per capire come trattarle al meglio è fondamentale identificare le proteine associate al funzionamento anomalo che genera lo stato di malattia.

2.1. Le proteine

L’importanza delle proteine nel processo biologico è molto differenziata. Ad esempio le proteine hanno un ruolo di catalizzatore delle reazioni chimiche che avvengono all’interno della cellula :

- nella funzione di scambio di materia tra i compartimenti cellulari e tra la cellula e l’ambiente esterno (canali, trasportatori di membrana),
- nella comunicazione chimica tra cellule differenti (ormoni, citochine, recettori di membrana),
- nella protezione immunitaria (anticorpi, complessi di istocompatibilità),
- nelle funzioni energetiche (complessi respiratori e fotosintetici),
- nelle funzioni motorie (actina, miosina),
- nella corretta espressione delle informazioni genetiche.

2.1.1. Gli aminoacidi

Le proteine sono macromolecole costituite da una o più catene di elementi più semplici; tali elementi prendono il nome di aminoacidi. In natura vi sono in totale 20 aminoacidi. Una proteina è formata da una sequenza di aminoacidi di lunghezza compresa tra 50 e 1000 unità. Gli aminoacidi sono molecole formate da un atomo di carbonio centrale C^{α} , da un gruppo amminico NH_3^+ e un gruppo acido –OOC (da cui il nome), e da un gruppo di atomi, detto radicale e indicato con R, che differenzia gli amminoacidi tra loro [BUI01].

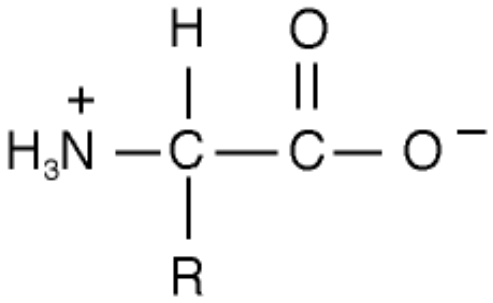

Fig. 2.1 Aminoacido generico

La formazione della proteina avviene attraverso la creazione di un legame tra gli aminoacidi, detto legame peptidico. Da qui il nome di catena polipeptidica con cui vengono spesso indicate le proteine. Gli atomi compresi nel legame peptidico costituiscono la catena principale della proteina, detto anche *backbone*, mentre gli altri costituiscono le *catene laterali*. Le proteine, una volta formatesi, tendono a ripiegarsi su se stesse a causa di alcuni legami deboli (legami a ponte di idrogeno) o di legami più forti (ponti disolfuro). Il legame rigido che si crea tra gli aminoacidi determina la conformazione strutturale della proteina. La sequenza di amminoacidi che compongono una proteina deriva dall'informazione codificata nel DNA, e viene rappresentata semplicemente come una sequenza di lettere (ogni lettera individua un amminoacido, vedi tab. 2.1).

(a) ATTRQDASSYDEQGKRNATWAGRYTDDRWRSAQMD

Fig. 2.2 Esempio di una sequenza di amminoacidi

tab. 2.1 I 20 aminoacidi

Aminoacido	**Sigla**	**Codifica**
Alanina	Ala	A
Citosina	Cys	C
Acido aspartico	Asp	D
Acido glutamico	Glu	E
Fenilalanina	Phe	F
Glicina	Gly	G
Istidina	His	H
Isoleucina	Ile	I
Lisina	Lys	K
Leucina	Len	L
Metionina	Met	M
Asparginina	Asn	N
Prolina	Pro	P
Glutamina	Gln	Q
Arginino	Arg	R
Serina	Ser	S
Treonina	Thr	T
Valina	Val	V
Triptofano	Trp	W
Tiroxina	Tyr	Y

È lecito pensare, come afferma l'ipotesi di Anfinsen (Anfinsen et al., 1961; Anfinsen, 1973), che la conformazione della struttura tridimensionale, ovvero l'occupazione spaziale degli aminoacidi di una proteina, sia interamente determinata dalla sequenza, cioè che nella stringa siano contenute tutte le informazioni necessarie a determinare la struttura tridimensionale della proteina.

2.1.2. la struttura delle proteine

La capacità delle proteine di interagire con l'organismo e più precisamente in particolari funzioni biologiche è strettamente legata alla struttura tridimensionale della proteina. Tale struttura le rende capaci di esplicare specifiche funzioni (come

quella strutturale, quella regolativa o enzimatica, quella energetica, quella ormonale), poiché promuove l'iterazione tra proteine diverse, con il DNA o con sostanze di natura chimica differente. La struttura tridimensionale dipende unicamente dalla catena aminoacidica (che costituisce la cosiddetta struttura primaria o covalente): sono le interazioni tra gli amminoacidi a determinare la conformazione ordinata della proteina. Questa è nota come ipotesi termodinamica di Anfinsen (Anfinsen et al., 1961; Anfinsen, 1973).

La struttura della proteina può essere considerata a diversi livelli di astrazione; si individuano 4 tipi di struttura:

- *Struttura primaria:* cioè la sequenza di amminoacidi delle sue catene peptidiche. Ogni proteina ha una sequenza unica.
- *Struttura secondaria:* l'arrangiamento spaziale della catena polipeptidica ripropone soluzioni strutturali locali ricorrenti per cui esistono delle sottostrutture che si presentano in proteine diverse, ne sono un esempio le α-eliche e i β-sheet (fig. 2.3a, fig. 2.3b). Individuare la struttura secondaria di una proteina significa determinare il tipo e la posizione di queste sottostrutture. Esiste una catalogazione in strutture super secondarie, dette motivi, dove si individuano due o più strutture secondarie. Inoltre esistono organizzazioni di motivi complessi, definite domini.
- *Struttura terziaria:* per struttura terziaria si intende la conformazione stabile e funzionale che una catena di una proteina assume nello spazio. Essa può essere vista come agglomerato di strutture secondarie ed è il risultato di legami instaurati tra i diversi aminoacidi.
- *Struttura quaternaria:* Molte grandi proteine contengono più di una catena polipeptidica, denominata *subunità*; la sistemazione spaziale di queste subunità è la struttura quaternaria della proteina. Le forze che tengono insieme le subunità sono le stesse di quelle che stabilizzano la struttura terziaria di proteine (forze di van der Waals e London , ponti salini e legami ad idrogeno).

2.1.3. La rappresentazione dei dati

I dati raccolti sulle sequenze e strutture delle proteine vengono rappresentati secondo diversi formati e resi disponibili in banche dati attraverso il World Wide Web. Esistono diversi tipi di banche dati, in gran parte specializzate su aspetti particolari della ricerca; ad esempio esistono banche dati contenenti i risultato del sequenziamento delle proteine, quelle contenenti dati sulle strutture, sul

sequenziamento di DNA e RNA, sul genoma umano o su aspetti particolari di esso. La comunità scientifica biologica si affida a specializzate reti bioinformatiche, come la EMBnet (European Molecular Biology Network) [w57] che consentono ai ricercatori l'accesso e la condivisione dei dati.

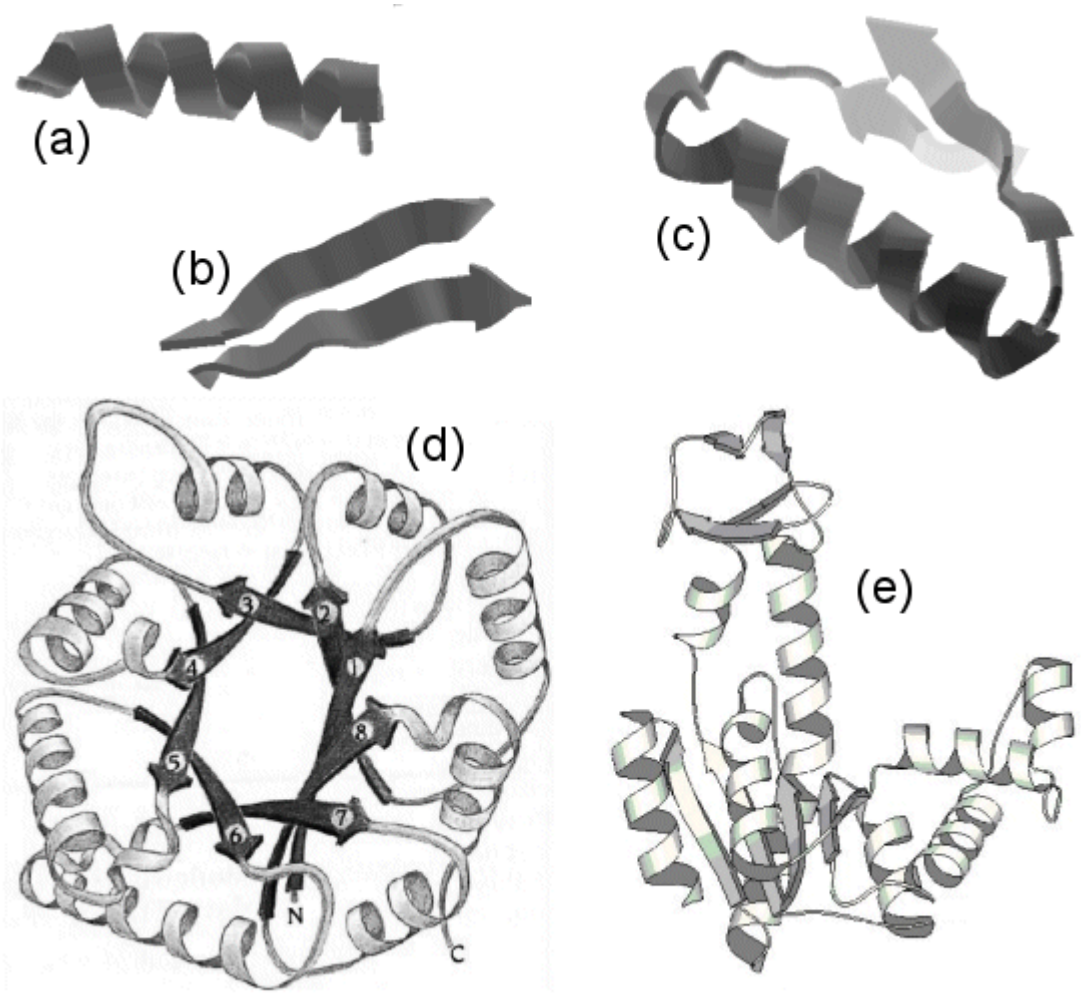

Fig 2.3 Elementi di struttura della proteina: a) α-elica; b) β-sheet antiparalleli; c) struttura supersecondaria; d) dominio, TIM barrel;e) struttura terziaria.

Le sequenze primarie possono derivare dal sequenziamento di proteine isolate e purificate o dalla traduzione di sequenze di acidi nucleici, possono appartenere a proteine pienamente caratterizzate o essere definite come proteine putative. La banca dati contenente i dati più "puliti" e meglio annotati è SWISS-PROT [w43] che raccoglie le sequenze di 260.000 proteine (20-02-2007). La banca dati TrEMBL [w44] contiene invece 3.800.000 sequenze proteiche (20-02-2007) tradotte da acidi nucleici e annotate automaticamente con metodi computazionali. Una banca dati che

somma tutte le sequenze proteiche disponibili è la NR (Non Redundant) [w45] che contiene circa 1.000.000 sequenze, derivanti dall'unione di tutte le banche dati e dall'eliminazione delle sole sequenze identiche.

Il formato maggiormente utilizzato nell'analisi strutturale proteica è il Brookhaven Protein Databank, detto PDB. La banca dati PDB [w46] contiene, allo stato attuale, le coordinate atomiche di 42.000 proteine a struttura nota.

Gli esperimenti che permettono di risolvere la struttura di una proteina sono essenzialmente la diffrazione a raggi X di una proteina cristallizzata e la risonanza magnetica nucleare di (piccole) proteine in soluzione. Il numero di strutture note è relativamente piccolo, se confrontato a quello delle sequenze, a causa delle difficoltà sperimentali della determinazione della struttura tridimensionale. Nel PDB sono disponibili anche strutture proteiche non derivanti da cristallizzazione ma costruite tramite procedure computazionali di predizione strutturale.

2.1.4. Il formato PDB

Il formato PDB è gestito dal consorzio Research Collaboratory for Structural Bioinformatics, composto da: Department of Chemistry and Chemical Biology del Rutgers State University of NewJersey, Biotechnology Division and Informatics Data Center del National Institute of Standards and Technology (NIST), University of California, SanDiego (UCSD) San Diego Supercomputer Center (SDSC). I dati depositati, di pubblico dominio, sono messi a disposizione di tutta la comunità scientifica via http [w46]. Il progetto Protein Data Bank è nato nel 1971 presso Brookhaven National Laboratories (BNL); il numero di strutture depositate ha avuto una crescita esponenziale e al 20-02-2007 conta 42.000 proteine.

Un file Protein Data Bank (PDB) è un archivio di strutture tridimensionali determinate sperimentalmente di macromolecole biologiche, utilizzato dalla comunità scientifica, ricercatori, docenti e studenti. Tale archivio contiene le coordinate atomiche, le citazioni bibliografiche, informazioni sulla struttura primaria e secondaria, e altri generi di informazioni raccolte durante il processo di cristallografia a raggi X.

Lo standard PDB prevede che le informazioni siano registrate in un file di tipo testo le cui righe hanno una larghezza prefissata di 80 caratteri. I primi sei caratteri di ogni riga codificano il tipo di informazione presente nel resto della riga utilizzando dei tag predefiniti. Tra i tag di maggiore importanza vi sono SEQRES che si riferisce alle informazioni sulla sequenza di amminoacidi di ogni catena e

ATOM che serve a specificare i dati sulle coordinate spaziali degli atomi. Ecco un esempio di tali tag:

```
SEQRES   1 A  175  MET ALA PRO LEU GLY PRO THR GLY PRO LEU PRO  1BGE
33
ATOM     47  CA  PHE A  14      17.770   7.846  89.981
1BGE129
```

Fig. 2.4 Esempi di record del formato “pdb”

Nel tag SEQRES compare il numero della riga e la lettera identificativa della catena e di seguito una sequenza di 13 amminoacidi, se ve ne sono di più si continua nella riga successiva incrementando l’indice di riga. Per ogni nuova catena l’indice di riga viene riportato a 1. Nel tag ATOM compaiono nell’ordine da sinistra a destra: indice e nome dell’atomo, tipo di aminoacido, lettera della catena e indice del residuo di appartenenza, coordinate spaziali x, y, z.

Se da un lato l’utilizzo del formalismo PDB consente l’uniformità del trattamento dei dati di struttura esso si rivela piuttosto rigido e non sempre si adatta alle esigenze dei cristallografi. Inoltre la necessità di depositare i dati tempestivamente costringe i ricercatori ad inserire anche dati non completamente raffinati o a volte ambigui. Queste necessità portano alla pubblicazione di archivi di strutture che non rispettano esattamente le direttive, archivi in cui i cristallografi hanno adattato il formalismo alle proprie esigenze. Il risultato è che i metodi automatici che utilizzano questi dati, ad esempio per l’allineamento strutturale delle proteine, non possono utilizzare appieno tutte le informazioni. Il Protein Data Bank mette a disposizione nel proprio sito Internet alcuni software per l’analisi della correttezza dei file PDB; tali programmi oltre a verificare la correttezza del formato tentano di correggere eventuali errori.

Il Protein Data Bank, oltre a consentire l’accesso di pubblico dominio al proprio database fornisce una distribuzione di 7 CD contenenti tutto l’archivio dei dati in formato compresso, ciò consente ai ricercatori l’esecuzione di prove su tutto il database in locale.

I file PDB sono comunemente utilizzati dagli algoritmi di allineamento come base di dati per le coordinate atomiche e le informazioni sulla sequenza delle proteine prese in esame.

Esistono delle applicazioni in grado di visualizzare la struttura della proteina memorizzata in un pdb, rendendo più intuitiva l’informazione contenuta in essi.

Programmi di questo tipo possono essere utilizzati anche per esaminare i risultati di una sovrapposizione strutturale. Una descrizione dettagliata dei principali tool esistenti per la visualizzazione e analisi di dati biologici è data nel capitolo 8.

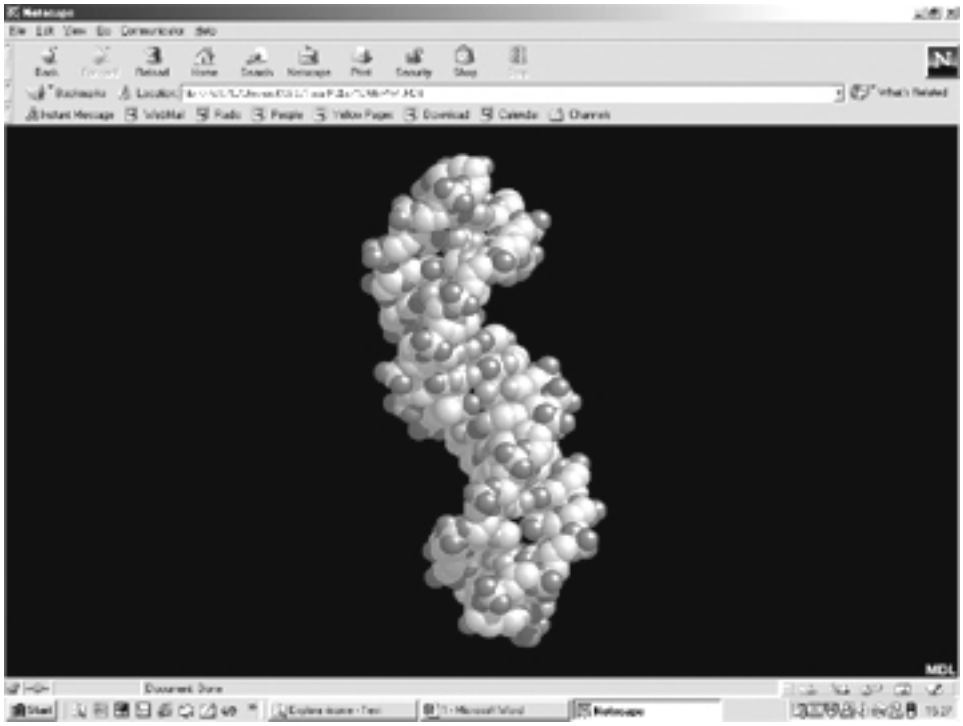

Fig. 2.5 Visualizzazione della struttura di un frammento di DNA, file 1D66.PDB, plugin Chime (www.umass.edu/microbio/chime).

2.2. La proteomica

La recente notizia della completa identificazione della proteina fondamentale per la fisiologia cerebrale ad opera di ricercatori dell'Università di Pavia ha portato all'attenzione anche del pubblico, anche italiano, di quella che sarà una nuova frontiera di ricerca in biologia molecolare: la proteomica.

La proteomica è una disciplina che ha come radice la genomica e che, dopo il raggiungimento del sequenziamento del genoma umano, sta riscontrando sempre più interesse nel mondo della ricerca. Tale disciplina ha come oggetto di studio le proteine, in particolare si propone di comprendere quali delle loro caratteristiche sono coinvolte attivamente nelle funzioni dei processi biologici. Allo stato attuale non esiste, infatti, la conoscenza sufficiente che data una proteina gli associ una funzione biologica.

2.2.1. Linee di ricerca

La proteomica si sta sviluppando secondo tre linee principali [w5]: la prima si occupa della micro-caratterizzazione delle proteine, della loro identificazione su larga scala e delle loro modificazioni; la seconda confronta i livelli proteici in situazioni e tessuti diversi e promette ricadute terapeutiche contro un ampio spettro di patologie; la terza area di ricerca, infine, studia le interazioni delle proteine tra di loro e con i tessuti circostanti. E' molto difficile prevedere il funzionamento di una proteina solo sulla base della struttura, anche se confrontata con altre strutture proteiche analoghe. Per poter effettuare una corretta analisi funzionale, sembra dunque fondamentale comprendere come le proteine si strutturano in complessi proteici e si correlano alla struttura della cellula nel suo complesso. Molti studiosi considerano quest'ultimo come il filone di ricerca più promettente di tutta la proteomica.

2.2.2. Tecnologie

La grande varietà delle proteine e la complessità che caratterizza i sistemi biologici richiedono, come forse in biologia non era mai accaduto prima, il ricorso a tecnologie estremamente sofisticate per supportare la ricerca, e a una conoscenza “di frontiera” per i ricercatori che devono saper combinare esperienze di discipline tra loro molto diverse: chimica, biochimica, informatica e medicina. I dati relativi alle proteine, utilizzati nella proteomica, vengono estratti tramite la separazione di migliaia di polipeptidi per elettroforesi e la loro successiva identificazione. In questa fase gioca un ruolo fondamentale la spettrometria di massa, soprattutto se combinata con strumenti prodotti della tecnologia bioinformatica. Una delle tecnologie di punta è denominata MALDI (acronimo da matrix-assisted laser desorption ionization) e consiste in una tecnica di spettroscopia di massa ideata una decina di anni fa e rivelatasi particolarmente adatta all'analisi proteica.

Anche in questa neonata scienza hanno subito assunto un ruolo di primo piano l’informatica perché oltre ad algoritmi efficienti per l'analisi di questi particolari tipi di informazioni si è dimostrata necessaria un'enorme potenza di calcolo. IBM è già entrata nell'impresa e si è associata tra l'altro a una delle aziende emergenti, l'americana MDS Proteomics, mettendo a disposizione computer e adeguate infrastrutture di rete. La comunità scientifica sta dedicando una grande attenzione alla proteomica e già si comincia a parlare di uno Human Proteomics Project, che dovrebbe raccogliere l'eredità del progetto genoma. L'impresa richiederà senza dubbio investimenti imponenti, ma le promesse, sia scientifiche che economiche sono di grande portata e anche il mondo industriale sembra se ne sia già reso conto.

2.3. Analisi proteiche

2.3.1. Predizione strutturale

La predizione strutturale proteica è una delle più significative aree di ricerca della proteomica. Ha come scopo la determinazione della struttura tridimensionale a partire dalla sequenza aminoacidica della catena polipeptidica. In termini più formali può essere definita come predizione della struttura terziaria tramite le informazioni date dalla struttura primaria.

L'importanza del ruolo della predizione strutturale proteica deriva dal fatto che una massiccia mole di dati derivanti dalle moderne tecniche di sequenziamento sono immagazzinate nei database di sequenze. Il numero di strutture note, invece, è di gran lunga inferire visto che i costi economici e temporali delle tecniche di estrazione strutturale sono molto maggiori rispetto a quelle sequenziali.

Diversi sono gli aspetti che rendono difficoltoso il problema della predizione strutturale, tra questi:

- Esistono più di una struttura possibile data una sequenza proteica.
- Le basi fisiche della stabilità strutturale non sono ancora totalmente note.
- La struttura terziaria di una proteina viene modificata se questa è a contatto di molecole esterne che ne alterano i legami.
- Anche l'ambiente esterno ad una proteina contribuisce al suo ripiegamento, ciò significa che la stessa sequenza può avere diverse strutture in base a fattori esterni.

Malgrado gli ostacoli apportati dagli aspetti sopra elencati, grandi progressi sono stati fatti dai tanti gruppi di ricerca interessati del campo. Un buon numero di approcci sono stati introdotti per risolvere il problema della predizione.- Questi approcci possono essere classificati in due grandi classi: modellazione *ab initio* e modellazione comparativa.

I metodi di modellazione proteica *ab initio* cercano di risolvere il problema della predizione strutturale tramite principi fisici senza utilizzare la conoscenza data dalle strutture risolte precedentemente. In generale si ricerca la miglior struttura attraverso l'uso di metodi stocastici che effettuano una ottimizzazione globale di una appropriata funzione per il calcolo dell'energia. Questi metodi richiedono un'elevata quantità di risorse computazionali e, quindi, tendono ad essere applicati solo a piccole sequenze di proteine. Per poter predire la struttura di proteine di dimensioni maggiori è necessario utilizzare metodi che sfruttano tecnologie di calcolo avanzato

come i supercomputers (BlueGene, MDGRAPE-3) o il calcolo distribuito (Human protome Folding Project).

La modellazione proteica comparativa utilizza l'informazione ottenuta dalle strutture risolte precedentemente come punto di partenza, sottoforma di template, per la predizione. Questo metodo risulta efficace in quanto il numero di strutture terziarie è stimato nell'ordine delle 2000 anche se le proteine in natura stimate nell'ordine di decine di milioni. I metodi che utilizzano la modellazione comparativa possono essere suddivisi in due gruppi principali: l'*homology modelling* e il *protein threading*.

L'*homology modelling* è basato sull'assunzione che due proteine omologhe condividono strutture simili. Questo deriva dal fatto che il ripiegamento di una proteina è in generale maggiormente conservato rispetto alla sua sequenza proteica. Una sequenza target può essere modellata con un buon livello di accuratezza su di un template associato ad una proteina omologa. È stato provato che il collo di bottiglia di questo approccio risulta essere la difficoltà di ottenere un buon allineamento sequenziale piuttosto che errori nella predizione della struttura data un buon allineamento [Zha05].

Nel *protein threading* [Bow91] viene effettuata una scansione della sequenza degli aminoacidi contro un database di strutture risolte. In ogni caso viene utilizzata una funzione di valutazione della compatibilità della sequenza alla struttura, questo permette di ottenere diversi modelli tridimensionali possibili. Questi tipi di modellazione sono anche conosciuti col nome di *3D-1D fold recognition* visto l'utilizzo contemporaneo di informazioni strutturali e sequenziali.

2.3.2. Allineamento sequenziale

Nella bioinformatica l'allineamento sequenziale consiste nel confrontare sequenze di DNA, RNA o proteine allo scopo di identificare sottosequenze simili che possono indicare relazioni evolutive, strutturali o funzionali tra le sequenze. Sequenze allineate di nucleotidi o residui di aminoacidi sono generalmente rappresentate come righe all'interno di una matrice. Tra i residui possono essere inseriti gap allo scopo di migliorare l'allineamento ottenuto. Se due sequenze in un allineamento condividono un antenato comune, è possibile ottenere delle informazioni relative ai percorsi evolutivi; infatti gli aminoacidi che non corrispondono possono rappresentare delle mutazioni genetiche mentre i gap possono rappresentare dei percorsi evolutivi differenti.

Le caratteristiche biochimiche degli aminoacidi sono determinate dalla catena laterale, infatti come introdotto nel capitolo 3. l'aminoacido è costituito da una parte comune capace di formare un legame peptidico e un'altra diversa per ogni aminoacido. Le catene laterali, anche se tutte diverse tra loro, possono presentare caratteristiche biochimiche simili, questo permette di determinare un grado di similarità tra i diversi aminoacidi. In una mutazione il grado di similarità tra i due aminoacidi coinvolti indica quanto questa è importante dal punto di vista biochimico. Si parla di mutazione conservativa quando gli aminoacidi sono molto simili, mentre, in caso contrario si parla di mutazione non conservativa.

Nell'allineamento sequenziale il grado di similarità tra due aminoacidi che occupano una particolare posizione nella sequenza può essere interpretato come una misura grossolana del grado di conservazione di una particolare regione. Tale misura può evidenziare zone ad alta similarità, ossia dove gli aminoacidi sono completamente conservati o le eventuali mutazioni sono di tipo conservativo. Le zone maggiormente conservate potrebbero individuare strutture importanti per il funzionamento biologico.

Gli approcci per l'allineamento sequenziale possono essere suddivisi in due categorie: allineamento di coppie di sequenze e allineamento multiplo. L'allineamento di coppie viene semplicemente utilizzato nella ricerca di similarità tra due sequenze, le regioni simili possono indicare motivi strutturali condivisi tra le due sequenze. L'allineamento multiplo permette di trovare similarità su un insieme più ampio di sequenze. Questo viene generalmente utilizzato per identificare regioni conservate tra gruppi di sequenze che ipoteticamente sono correlate dal punto di vista evolutivo.

2.3.3. Allineamento strutturale

L'allineamento strutturale ha come scopo la comparazione di due o più strutture polipeptidiche per cercare similitudini nella loro conformazione spaziale. Questo tipo di analisi è tipicamente applicato a strutture terziarie proteiche, ma può anche essere utilizzata su grandi molecole come l'RNA. Contrariamente alla semplice superimposizione strutturale, dove un sottoinsieme dei residui corrispondenti sono noti, l'allineamento strutturale non utilizza una conoscenza a priori delle corrispondenze.

La comparazione strutturale è un metodo di comparazione prezioso per analizzare proteine aventi una bassa similarità di sequenza, in questi casi infatti è difficoltoso

ottenere informazioni su relazioni evolutive tramite le sole tecniche di allineamento sequenziale. Tuttavia è indispensabile trattare i risultati con cautela in quanto potrebbero essere alterati dagli effetti dell'evoluzione convergente (vedi sez. 2.4.1) dove sequenze di aminoacidi non correlate potrebbero convergere in una struttura terziaria comune.

L'allineamento strutturale può essere utilizzato per comparare due o più catene contemporaneamente. Naturalmente per poter effettuare un allineamento strutturale è indispensabile che la proteina abbia la struttura primaria e terziaria note. Queste informazioni sono ottenute tramite le tecniche di estrazione dati o possono essere predette utilizzando i metodi descritti in questo capitolo. L'allineamento strutturale può anche risultare uno strumento utile per la valutazione degli allineamenti da metodi sequenziali [Zha05].

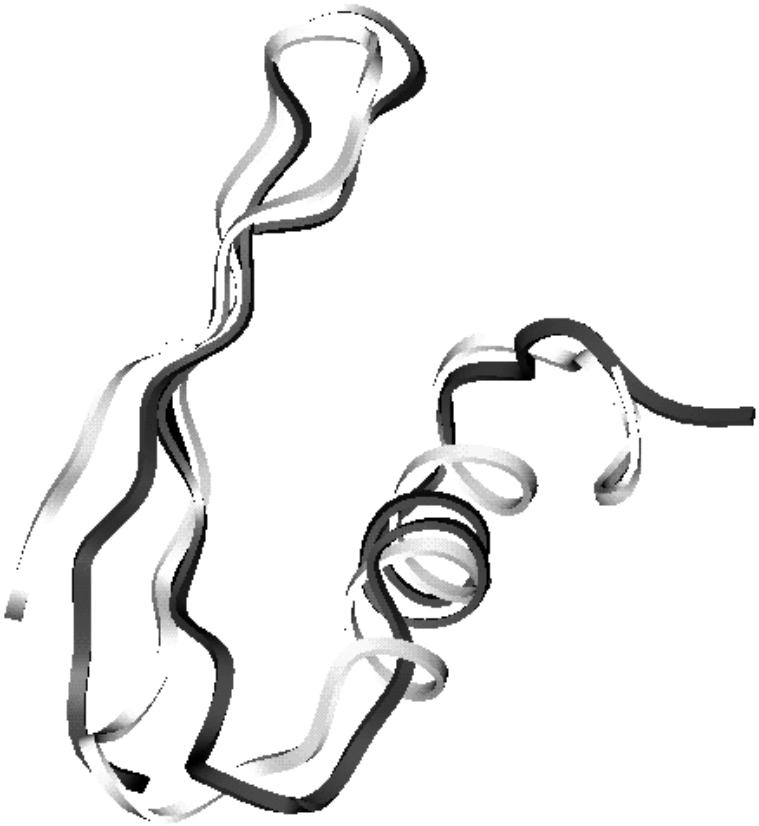

Fig. 2.6 Rappresentazione grafica tridimensionale di un allineamento strutturale tra due catene di proteine

Il risultato di un allineamento strutturale è composto da una matrice di rotazione e un vettore di traslazione che, applicati ad una delle due strutture, individuano la soluzione ottenuta. Un ulteriore risultato che quantifica la bontà dell'allineamento è la distanza RMSD calcolata tra le due strutture. In fig. 2.6 è rappresentato un allineamento effettuato su due catene polipeptidiche.

I metodi maggiormente utilizzati per allineare strutture proteiche sono:

- il DALI (Distance ALIgnment matrix) [Hol96], che effettua una suddivisione della catena polipeptidica della proteina e calcola una matrice delle distanze tra i frammenti. Le matrici sono poi utilizzate per ricercare pattern di contatto tra frammenti successivi;
- l'SSAP (Sequential Structure Alignment Program) [Tay94], che si basa sulla Dynamic Programming per produrre un allineamento strutturale basato sui contatti polipeptidici. I risultati di questo metodo di allineamento sono utilizzati per effettuare una classificazione gerarchica delle proteine conosciuta col termine CATH (Class, Architetture, Topology, Homology);
- il CE (Combinatorial Extension) [Shi98], risulta essere simile al DALI in quanto suddivide le strutture in frammenti, questi sono poi utilizzati per cercare il miglior allineamento su coppie di frammenti. Queste coppie sono poi utilizzate per generare un'unica matrice di similarità e identificare l'allineamento strutturale finale.

2.3.4. Docking molecolare

Il problema del docking molecolare consiste nel determinare la struttura di un complesso proteico formato da due o più proteine senza l'ausilio di misure sperimentali. L'interesse sul docking molecolare è stato alimentato dal rapido incremento delle strutture proteiche. Per alcune proteine, che non subiscono importanti variazioni a livello strutturale nel formare complessi, è possibile calcolare con buoni risultati l'interfaccia di docking. Metodi risolutivi del docking molecolare non specificano la sequenza dei legami formati tra le diverse proteine ma restituiscono la struttura del complesso risultante.

Le interazioni, che caratterizzano i processi biologici, della maggior parte delle proteine sono interamente sconosciute. Persino le proteine che partecipano ad un processo biologico ben conosciuto come il ciclo di Krebs, possono avere interazioni o funzioni in processi differenti. In caso di interazioni note tra proteine, sorgono altri

quesiti. Le malattie genetiche conosciute sono causate da proteine mutate o dispiegate come la fibrosi cistica, in questi casi è fondamentale conoscere come, data una mutazione genetica, questa possa causare anomalie nelle interazioni proteiche. Una volta conosciuta la risposta a tali quesiti, sarebbe possibile costruire proteine sintetiche allo scopo di interagire nelle funzioni biologiche e sarà quindi essenziale conoscere precisamente le interazioni di tali proteine.

Dato un insieme di proteine sarebbe interessante sapere:

- se interagiscono tra loro formando un complesso proteico;
- in caso di interazione quale struttura il complesso assume;
- in caso di formazione di un complesso quanto è stabile il legame instaurato;
- in caso di mancata interazione, quale mutazione si potrebbe applicare per portarle alla formazione di un complesso.

In futuro, quando il problema del docking molecolare sarà conosciuto in tutti i suoi aspetti, sarà possibile rispondere in maniera esauriente ai quesiti sopra citati. Inoltre quando i principi fisici che regolano le interazioni saranno completamente noti, sarà possibile calcolare il docking di tutte le proteine per inferire le loro funzioni. L'unico prerequisito necessario sarebbe la conoscenza della struttura proteica, sia essa determinata in maniera sperimentale.

2.4. La superficie molecolare

Agli inizi degli anni '60 Kauzmann [Kau59] studiò la distribuzione delle cariche idrofobiche interne ad una proteina in base alla conformazione della propria catena. In questo lavoro mostrò un modello del folding di una proteina evidenziando le catene laterali idrofobiche degli aminoacidi racchiusi nella parte interiore della struttura proteica. Ipotizzò così che le proteine, come le gocce d'olio, tentano di ripiegarsi in modo da disporre preferibilmente la parte idrofobica all'interno della struttura evitando così il contatto con il solvente esterno. Questo fu uno dei primi lavori che fece una separazione degli atomi in base alla loro posizione nella molecola.

tab. 2.2: raggi atomici di Van der Vaals

Elemento	Raggio di Van der Vaals (Å)
H	1.2
N	1.5

O	1.4
F	1.35
P	1.9
S	1.85
Cl	1.8

La prima definizione di superficie molecolare fu introdotta da Van der Waals. Questa corrisponde alle superfici esterne degli atomi aventi raggio uguale a quelli riportati in tab. 2.2. La superficie di Van der Waals è raffigurata in fig. 2.7.

La proprietà di idropatia (idrofobicità e idrofilicità) è una delle caratteristiche principali degli aminoacidi. Questa caratteristica viene naturalmente ereditata dalla proteina in quanto essendo composta da aminoacidi presenta zone idrofobiche e idrofiliche lungo la catena polipeptidica. Una definizione di questa proprietà è data in seguito in sezione 2.5.2.

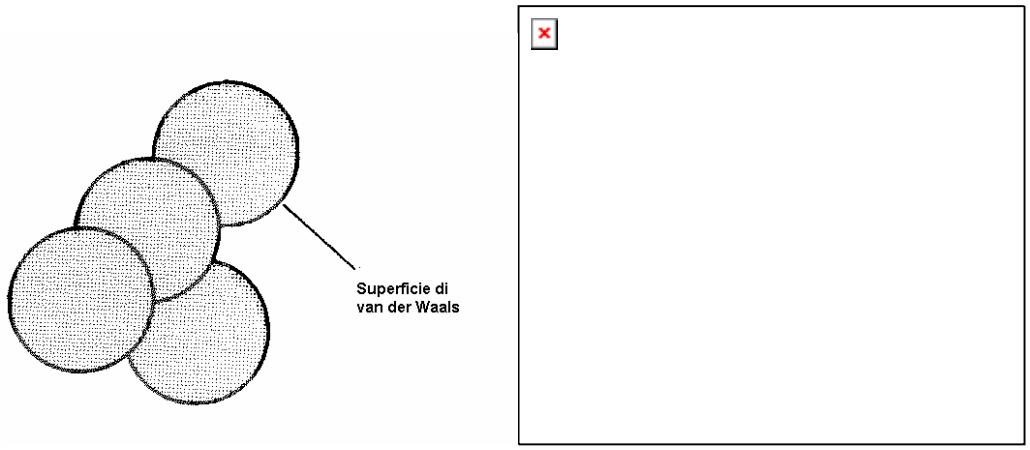

Fig. 2.7: Tipi di superfici proteiche

Allo scopo di quantificare le parti idrofobiche non racchiuse internamente alla struttura B.K. Lee e F. Richards [Lee71] introdussero nel 1971 la *superficie accessibile al solvente*. Questa superficie è composta da tutti i punti toccati dal centro di una sfera-sonda che rotola sulla proteina (fig. 2.7). Ne risulta una specie di superficie di Van der Waals con raggi atomici aumentati del raggio della sfera sonda. Il volume incluso dalla superficie accessibile al solvente è chiamata *volume escluso dal solvente*. Lee e Richards calcolarono l'area accessibile di ogni atomo della proteina nella conformazione ripiegata e trovarono che la diminuzione

dell'area esposta nel ripiegamento è maggiore per gli atomi idrofobici rispetto a quelli idrofilici, rafforzando così la tesi di Kauzmann.

Queste idee furono raffinate da Richmond e Richards nel 1978 [Ric78] quando introdussero la *superficie di contatto*. Questa superficie è definita dalla parte della superficie di Van der Waals che può essere toccata da una sfera-sonda con raggio pari a quello di una molecola d'acqua. Non molto più tardi Richards concluse il raffinamento della definizione della superficie intrapresa con Richmond definendo la *superficie non accessibile al solvente*. Questa superficie è il risultato dell'unione della superficie di contatto con quella definita dalla parte di superficie nascosta della sfera di probe quando è in contatto con più di un atomo.

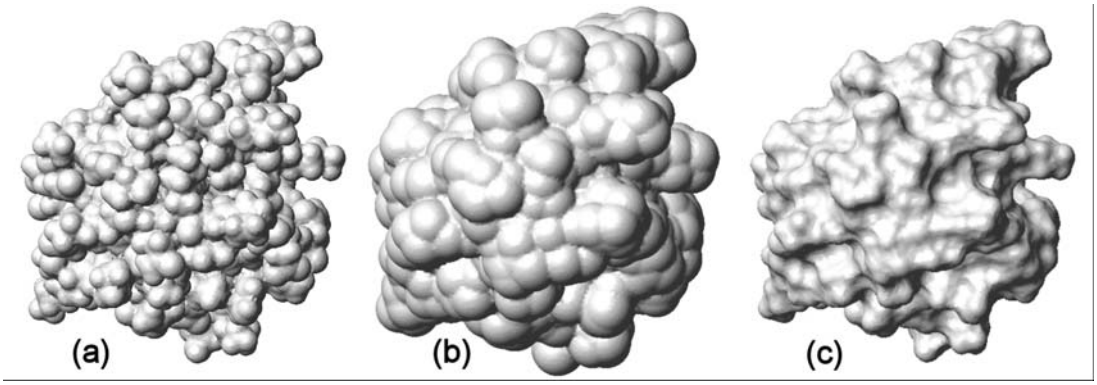

Fig. 2.8: Esempi di superfici proteiche per una proteina: (a) Van der Waals, (b) accessibile al solvente, (c) non accessibile al solvente.

Da allora sono stati pubblicati diversi metodi per il calcolo della superficie accessibile e non accessibile al solvente. Per quanto riguarda la superficie accessibile al solvente il metodo di Lee e Richards [Lee78] calcolava questa area moltiplicando la lunghezza degli archi accessibili con lo spazio compreso tra i piani. Il metodo proposto da Shrake e Rupley [Shr73], a differenza del precedente, piazzava 92 punti sulla superficie di ogni atomo definita dalla sfera con raggio aumentato e determinava quali punti erano accessibili al solvente, ovvero quelli non interni a nessuna altra sfera.

J. Greer e B. Bush proposero un metodo per il calcolo della superficie non accessibile al solvente [Gre78] e lo applicarono per visualizzare l'interfaccia di superficie utilizzata dalle emoglobine e per quantificare il volume dello spazio rimasto vuoto tra le superfici. Il loro lavoro consiste nel far cadere delle sfere-sonda sugli atomi della proteina e calcolare il punto di collisione con la superficie di Van der Waals. Nella collisione il punto più basso della sfera rappresenta un punto della superficie non accessibile al solvente. Questo metodo risulta corretto con superfici

convesse o piane mentre non riesce a gestire superfici che presentano una spinta concavità. Connolly et al. [Con81] proposero nel 1981 un metodo alternativo a quello introdotto da Greer per ovviare alle carenze che presentava. Definirono tre differenti tipi di patch di superficie: concava, convessa e a sella. La sfera-sonda che rotola sulla superficie si può trovare in tre stati diversi di contatto con la proteina. In un primo caso tocca un solo atomo di superficie, la parte di superficie di Van der Waals toccata dalla sfera quando si trova in questo stato rappresenta la patch convessa di superficie non accessibile al solvente; quando la sfera tocca due atomi definisce una patch a forma di sella; quando la sfera tocca tre atomi la parte della sua superficie compresa tra i tre punti di contatto rappresenta la patch concava

2.4.1. Motivazioni

Se per una proteina non è conosciuto il suo fenotipo ma è nota la sua struttura, è possibile presumere le sue funzioni comparando la sua struttura ad altre con funzionalità note . La similarità tra due strutture può implicare un antenato comune e quindi suggerire dettagli sul ruolo funzionale della proteina . Comunque è possibile che tali strutture omologhe abbiano funzioni differenti , le proteine, infatti, possono adottare strutture terziarie simili sviluppando però funzioni e siti attivi differenti .

Nei casi sopra esposti, e descritti con maggior livello di dettaglio nelle pubblicazioni citate, la comparazione strutturale non sarebbe efficace per presumere le funzionalità delle proteine analizzate. Questi casi non sono affatto rari, infatti è stato stimato che solo il 59% dei casi, dove viene riscontrata una similarità di struttura non supportata da similarità di sequenza, le proteine condividono un sito attivo funzionale. Le funzioni associate per similarità strutturale devono essere quindi trattate con un certo livello di indecisione. Inoltre, solo il 20% delle nuove proteine mostrano motivi strutturali sconosciuti rendendo ancor più problematica l'associazione funzionale tramite analisi strutturale.

Le strutture molecolari, sottoposte a processo evolutivo, possono subire delle variazioni causate da mutazioni genetiche. L'evoluzione può portare strutture simili a divergere in termini funzionali oppure strutture differenti a convergere verso lo stesso ruolo. In questi casi si può parlare di *evoluzione divergente* ed *evoluzione convergente*. In fig. 2.8 viene mostrato come il processo evolutivo può mutare le funzioni proteiche mantenendo la similarità strutturale. In figura abbiamo due proteine, P_1 e P_2, che presentano una struttura molto simile (circolare) e mostrano sulla loro superficie lo stesso sito attivo, rappresentato dal quadratino più scuro.

Queste due proteine sottoposte allo stesso periodo evolutivo (in verticale è rappresentato il tempo) possono essere leggermente modificate da mutazioni genetiche, in particolare P_1 viene modificata perdendo il sito attivo. In questo modo, al tempo t_2, otteniamo P_1' e P_2' strutturalmente simili ma funzionalmente differenti. Per queste due proteine possiamo parlare di evoluzione divergente. Al contrario l'evoluzione porta le due proteine P_2 e P_3, strutturalmente diverse, a svolgere la stessa funzione. Infatti per evoluzione convergente P_3 acquisisce il sito attivo presente su P_2 pur mantenendo la gran parte della sua struttura invariata.

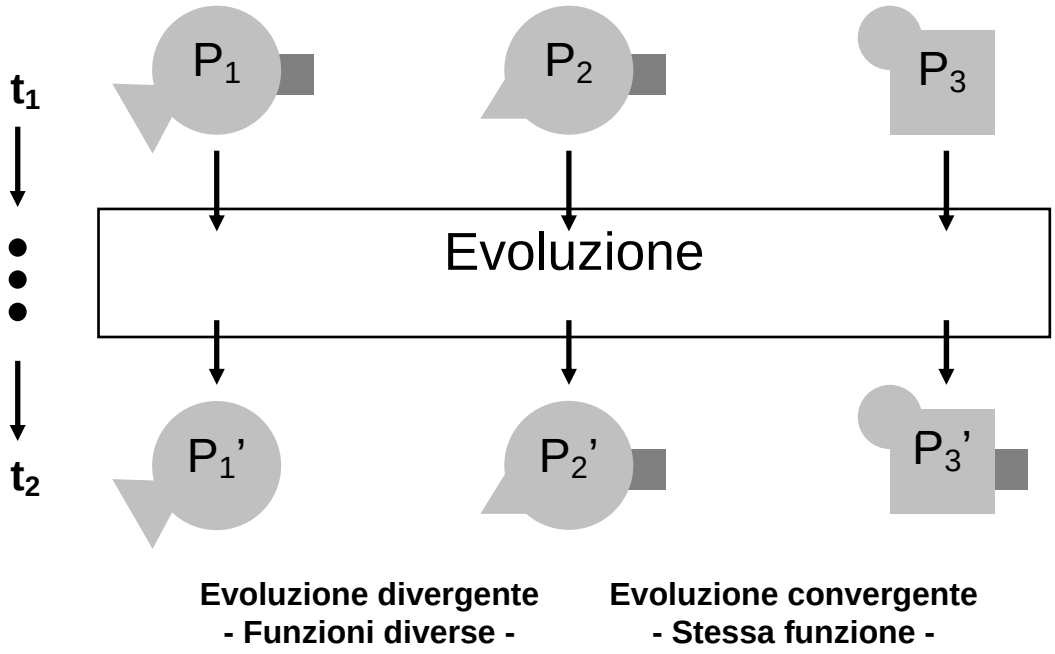

Fig. 2.8: Evoluzione convergente e divergente.

2.5. Le proprietà fisico-chimiche di superficie

Molti studi sono stati effettuati allo scopo di aumentare le conoscenze sulla natura delle aree di superficie coinvolte nelle interazioni tra proteine diverse. Queste aree di superficie vengono comunemente chiamate interfacce. In generale questi lavori analizzano la parte di superficie molecolare individuata dalle interfacce conosciute per portare alla luce le caratteristiche che favoriscono tali interazioni. Le proprietà determinanti di superficie possono essere classificate in due categorie: (a) proprietà fisico-chimiche e (b) proprietà geometriche. In seguito verranno riportati gli aspetti individuali dei lavori più rappresentativi.

(a) Come ci si aspettava sin dalle origini dei lavori sulla accessibilità del solvente, le diverse descrizioni della superficie proteica sono state applicate allo studio della solvatazione, della idrofobicità e dell'idrofilicità . Roe e Teeter hanno sviluppato

un metodo per predire la posizione delle molecole d'acqua vicino alle catene laterali degli aminoacidi polari. Cesari e Sander hanno studiato i contatti tra proteine e atomi di solventi e hanno evidenziato che le molecole di solvente nei solchi di superficie presentano una stabilità maggiore di contatto. Anche nel problema del docking molecolare l'idrofobicità è stata inserita come caratteristica fondamentale nel calcolo dell'energia di interazione . Nel considerare poi il calcolo del potenziale elettrostatico sulla superficie proteica è indispensabile citare gli studi effettuati da Lavery e Pullman che analizzarono questa proprietà sulla superficie del DNA. Altri innumerevoli lavori sono stati effettuati per definire il potenziale elettrostatico sulla superficie molecolare e per calcolare l'energia di interazione proteina-proteina o proteina-solvente.

(b) Da studi statistici effettuati da Janin e Chothia all'inizio degli anni '90 è stato ipotizzato che la superficie proteica appartenente a una interfaccia di interazione consiste di una o più aree contigue e che la superficie di interazione misura circa $1600 \pm 400\ \text{Å}^2$, l'equivalente di circa 170 ± 39 atomi di superficie. Spesso le interfacce sono costituite da una singola patch ma esistono casi in cui l'interazione coinvolge più zone non contigue sulla superficie proteica. Infatti circa 10 anni più tardi Chakrabarti e Janin evidenziarono che le interfacce di interazione possono essere sensibilmente maggiori in dimensioni e possono essere composte da più zone non necessariamente contigue. While, Jones e Thornton analizzarono ed evidenziarono come anche l'indice di curvatura gioca un ruolo importante nel favorire l'intrazione tra proteine, seguirono poi altri lavori che evidenziarono interfacce piane, concave e convesse.

Nelle seguenti sezioni verranno descritte in maniera dettagliata le proprietà considerate in questo lavoro di tesi ossia il potenziale elettrostatico e l'idrofobicità. L'indice di curvatura non è introdotto in questa sezione in quanto è una proprietà che si calcola in maniera puntuale sulla superficie e non proviene da caratteristiche biochimiche dgli aminoacidi. Il calcolo della curvatura verrà formulato in maniera formale in sez 4.2.3.

2.5.1. Il potenziale

Il potenziale elettrostatico viene regolato sulla superficie molecolare tramite la legge di Poisson-Boltzmann la quale sostiene che il valore del potenziale elettrostatico in un punto **r** è pari al lavoro fatto per portare una carica positiva

unitaria dall'infinito a quel punto. Il potenziale elettrostatico molecolare (in seguito MEP) deriva dalla sovrapposizione degli effetti dati dalle cariche positive nei nuclei degli atomi e dalle cariche negative negli elettroni. Il contributo positivo è calcolato tramite la seguente formula:

$$V_{+}(\mathbf{r}) = \sum_{i=1}^{N} \frac{Z_i}{|\,\mathbf{r} - \mathbf{R}_i\,|}$$

dove *N* rappresenta il numero dei nuclei considerati nel calcolo, Z_i rappresenta la carica dell'atomo e $\mathbf{R}_i$ rappresenta il vettore posizione del nucleo nello spazio. Non è possibile applicare la stessa formula agli elettroni nel calcolo del contributo negativo in quanto la loro distribuzione è continua nello spazio, occorre quindi sostituire la sommatoria con un integrale e introdurre la densità elettronica ρ.

$$V_{-}(\mathbf{r}) = -\int \frac{\rho(\mathbf{r})d\mathbf{r}'}{|\,\mathbf{r}'-\mathbf{r}\,|}$$

Per i sistemi di interesse biologico è impensabile calcolare il potenziale dalla funzione d'onda quantomeccanica per il numero notevole di atomi di questi sistemi, il che rende il calcolo computazionalmente impossibile. Si fa perciò ricorso a metodi approssimati, derivati e controllati nel corso degli anni .

Il potenziale può essere derivato con buona approssimazione in modo additivo da contributi rappresentabili mediante opportuni sistemi di cariche puntiformi q_i poste nelle posizioni occupate dagli atomi, così facendo si stimano sia gli effetti positivi che negativi delle cariche.

$$V(\mathbf{r}) = \sum_{i}^{N} \frac{q_i}{|\,\mathbf{r} - \mathbf{R}_i\,|}$$

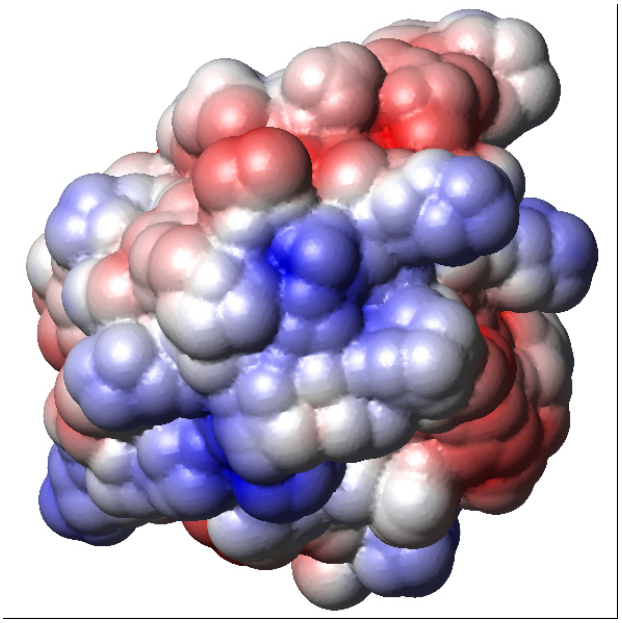

Fig. 2.9: Potenziale elettrostatico visualizzato sulla superficie proteica.

Il MEP è una proprietà locale e generalmente viene calcolata utilizzando un reticolo tridimensionale di punti intorno alla molecola. Il valore di questa proprietà sui punti di superficie viene stimato in base a quelli più vicini del reticolo. Nella rappresentazione grafica il potenziale elettrostatico viene codificato mediante i colori rosso per indicare MEP negativo e blu per positivo (fig 2.9).

2.5.2. L'idropatia

L'idropatia è una delle proprietà principali degli aminoacidi ma, purtroppo, allo stato attuale è ancora una delle caratteristiche meno conosciute. La parola idrofobia e idrofilia significano letteralmente "paura dell'acqua" e "amicizia con l'acqua". È chiaro che i residui idrofobici preferiscono stare in un ambiente privo d'acqua mentre quelli idrofilici cercano di mantenersi in contatto con essa.

La mancanza di omogeneità nella definizione di questa proprietà ha portato alla formulazione di misure di idropatia basate su diversi test sperimentali tra cui ricordiamo:

- Comportamento alla solubilità in ottanolo e in acqua
- Polarità della catena laterale calcolata tramite tecniche chimico-quantistiche
- Calcolo della distribuzione degli aminoacidi polari in superficie e nel core della proteina
- Valutazione della costituzione atomica delle catene laterali

Questa disomogeneità nel calcolo dell'idropatia ha portato alla pubblicazione in letteratura di innumerevoli scale che la quantificano. La scelta della scala da utilizzare risulta quindi molto importante e deve essere effettuata in base all'utilizzo che ne verrà fatto. Attualmente molti lavori pubblicati indicano nella scala proposta da Kyte-Doolittle quella di riferimento (tab. 2.3)

tab. 2.3: scala di idropatia secondo kyte-doolittle

Alanine	1.8
Arginine	-4.5
Asparagine	-3.5
Aspartic acid	-3.5
Cysteine	2.5
Glutamine	-3.5
Glutamic acid	-3.5
Glycine	-0.4
Histidine	-3.2
Isoleucine	4.5
Leucine	3.8
Lysine	-3.9
Methionine	1.9
Phenylalanine	2.8
Proline	-1.6
Serine	-0.8
Threonine	-0.7
Tryptophan	-0.9
Tyrosine	-1.3
Valine	4.2

La mancanza di un riferimento per la misura di idropatia è ovviamente il risultato dell'assenza di una definizione univocamente riconosciuta per questa proprietà. Molte persone tendono a definire l'idrofobicità come una carica che attrae aminoacidi non polari in modo da formare legami idrofobici. Questo modo di definire tale proprietà è tuttavia forviante in quanto descrive correttamente il comportamento di oggetti di questo tipo ma non ne specifica il perché.

Non esiste nessuna forza idrofobica, esiste invece una forza di attrazione tra atomi chiamata forza di Van der Waals. Questa forza risulta essere del tutto trascurabile a grandi distanze interatomiche e comunque molto debole a distanze dell'ordine di qualche Å.

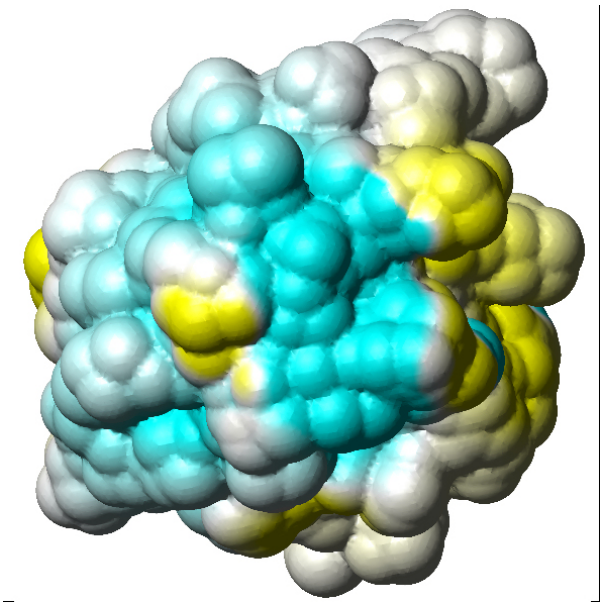

Fig. 2.10: Idropatia visualizzata sulla superficie proteica.

La differenza sostanziale tra aminoacidi idrofilici e idrofobici consiste nel fatto che i primi presentano la possibilità di formare legami a ponte di idrogeno mentre i secondi non ne hanno la capacità. Questa differenza emerge quando molecole aventi diverse idropatie si trovano immerse in un solvente quale l'acqua. La molecola d'acqua ha infatti la capacità di formare dei ponti di idrogeno e quando si trova circondata da altre dello stesso tipo tende a legarsi e organizzarsi con le vicine formando un reticolo ben ordinato.

Quando una molecola idrofilica si trova immersa in questa struttura reticolare di ponti ad idrogeno tende a spezzare alcuni legami rimpiazzandoli però con i propri. Ne deriva che la molecola idrofilica si integra bene in questo reticolo e riesce a convivere vicino a molecole d'acqua. Una molecola principalmente idrofobica, nelle stesse condizioni, non ha la capacità di sopperire ai legami spezzati coi propri e le molecole d'acqua vicine tendono a mantenere i legami con altre molecole d'acqua, spingendo così quella idrofobica al di fuori del reticolo. Si pensi a come le gocce d'olio, per la loro natura idrofobica, quando immerse in acqua, tendono ad avvicinarsi tra loro in quanto incapaci ad integrarsi nell'organizzazione a reticolo presentata dall'acqua.

Ecco perché atomi idrofobici cercano di mantenersi vicini ad altri dello stesso tipo e di mantenersi interni alla struttura della proteina mentre quelli idrofilici tendono a portarsi all'esterno della struttura proteica per formare legami a ponte d'idrogeno.

Anche diversi studi hanno evidenziato che in una proteina ripiegata su se stessa la concentrazione degli aminoacidi idrofobici è maggiore all'interno della proteina mentre i residui idrofilici sono maggiormente presenti sulla sua superficie.

3. Un approccio all'analisi di superfici proteiche

Un nuovo sentiero di sviluppo nella ricerca proteomica, focalizza l'interesse, soprattutto, sulle caratteristiche della superficie delle proteine e sul loro modo di interagire. Di notevole importanza risulta, inoltre, la classificazione delle proteine basata proprio sulle loro peculiarità di superficie [18]. Ne consegue una nuova area di ricerca e di sviluppo per la proteomica, qui approfondita evidenziando, in particolare, la definizione di *pattern di superficie*. Uno dei campi di ricerca in cui, dati questi presupposti, si sono raggiunti egregi risultati è quello del *docking biomolecolare* [19], nel quale ci si chiede se due proteine potranno interagire tra loro formando un contatto energetico stabile. La capacità di interagire dipende dall'esistenza sulle due proteine di aree di sufficiente ampiezza che presentino peculiarità compatibili, ad esempio carica elettrica positiva e forma convessa da una parte, carica elettrica negativa e forma concava dall'altra. L'utilizzo di una singola area per arrivare alla classificazione non è, nonostante ciò, sempre utile: le proprietà dell'intera proteina potrebbero essere determinate da un insieme di regioni non necessariamente limitrofe.

Nel progetto intrapreso e descritto in questa tesi lo scopo è determinare, dato un database di proteine di cui non sono note le funzioni, l'insieme dei pattern di superficie complessi che si presentano frequentemente, tali pattern possono poi essere utilizzati per classificare le proteine.

Per ottenere ciò, occorre usufruire, come sintetizzato in fig. 3.1.

(a) di tecniche di clustering, le quali permettono, iniziando da una rappresentazione particolareggiata della superficie proteica, di determinare un set di regioni peculiari;

(b) di definire una rappresentazione essenziale e al tempo stesso valida della superficie proteica;

(c) di applicare a questa tecniche di data mining per determinare i pattern frequenti.

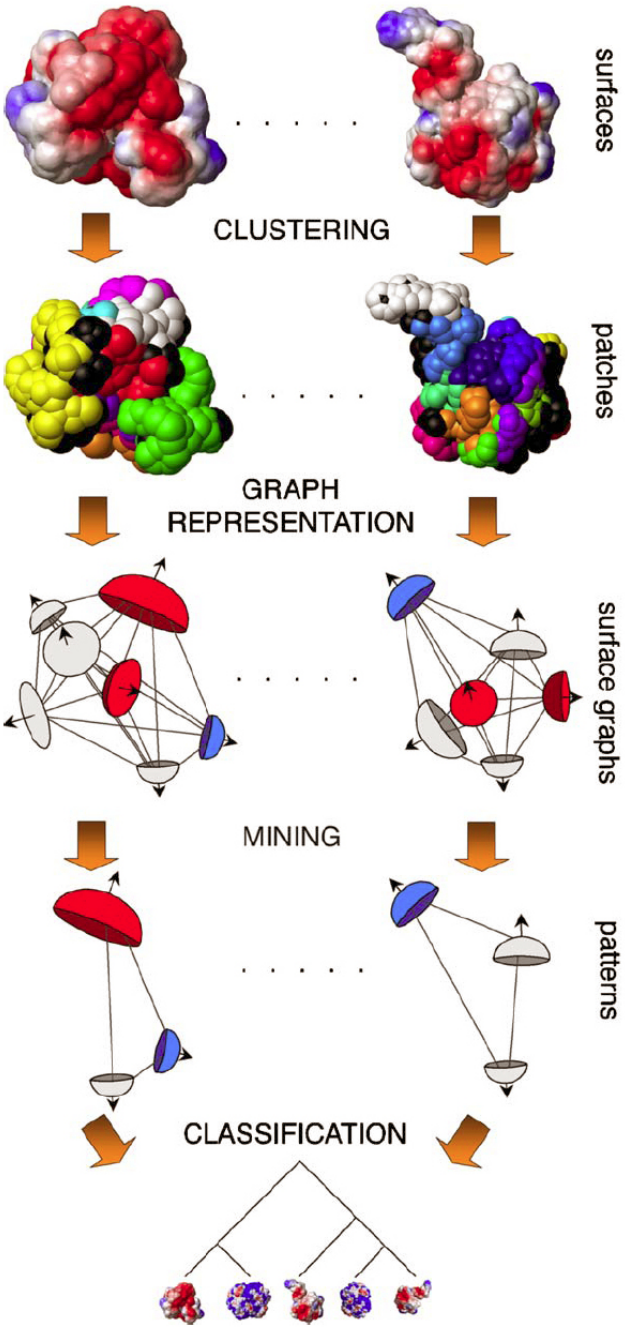

Fig 3.1*:* Il metodo proposto

3.1. Stato dell'arte

Nella bioinformatica la comparazione tra superfici molecolari ha guadagnato un ruolo rilevante nell'ultimo decennio. Questa consiste nell'evidenziare porzioni di superfici simili condivise da più molecole e potrebbe indicare strutture o sequenze differenti con siti attivi simili aventi la stessa funzione.

La scelta del livello di dettaglio nella rappresentazione della superficie è senza dubbio un elemento fondamentale in quanto deve essere raggiunto un compromesso

soddisfacente tra la precisione della rappresentazione e le prestazioni dell'algoritmo di comparazione. I metodi proposti in letteratura possono essere suddivisi in base a tre livelli di dettaglio utilizzati per descrivere la superficie: mesh-based, atom-based e patch-based.

Molti metodi di tipo mesh-based utilizzano il così detto algoritmo di Connolly . Attraverso questo metodo la superficie è definita da una sfera-sonda, di dimensioni pari alla molecola d'acqua, "rotolata" sugli atomi della proteina. La superficie accessibile al solvente è definita dall'insieme dei punti toccati dal centro della sonda. Due metodi di comparazione basati sulla superficie di Connolly sono proposti in, in questi casi la similarità è determinata comparando la disposizione spaziale dei vettori normali e i valori delle proprietà locali sui nodi di superficie.

I metodi di tipo atom-based utilizzano una descrizione della superficie definita dal sottoinsieme degli atomi esposti al solvente. Per esempio in gli autori danno una definizione della α-surface come l'insieme degli atomi toccati da una sonda di un dato raggio. Questa informazione è il punto di partenza per la ricerca di pattern di superficie su un database di proteine allo scopo di classificarle.

Nei metodi di tipo patch-based il livello di granularità della superficie viene aumentato considerando un insieme di atomi esposti, aminoacidi esposti o una rappresentazione più compatta ottenuta segmentando la superfici in regioni omogenee. In gli autori descrivono un metodo per l'annotazione funzionale di strutture proteiche attraverso la comparazione di patch locali di superficie con siti attivi funzionali annotati e disponibili in rete . Altri metodi utilizzano una rappresentazione a grafo della proteina, ottenuto dall'insieme delle patch di superficie, per cercare similarità all'interno di un DB di superficie e conseguentemente classificarlo. In particolare il metodo proposto in cerca il massimo sottografo comune tra due grafi dati, i nodi rappresentano le patch di superficie e sono etichettati come concavi, convessi e a sella, gli archi rappresentano le distanze tra patch.

In tanti lavori presenti in letteratura sul problema del docking molecolare la superficie viene mantenuta al maggior livello di dettaglio e vengono studiati dei metodi per cercare pattern di superficie compatibili con una molecola-sonda data in input. Infatti non molti lavori inquadrati in questo campo della proteomica affrontano il problema di rappresentare la superficie in maniera compatta. Tra la minoranza che utilizzano patch risulta interessante il metodo proposto in dove i

punti di superficie sono raggruppati tramite tecniche di growing in base alle loro proprietà geometriche e bio-chimiche.

3.2. Il metodo proposto

Il lavoro di ricerca presentato in questa tesi ha come scopo la classificazione di proteine basata su proprietà di superficie. In questo capitolo viene sintetizzato l'approccio proposto alla classificazione e verrà poi valutato nel capitolo 7 tramite l'utilizzo di diversi set di proteine, alcuni di essi creati ad hoc per testare gli algoritmi proposti, altri presi da database reali per validare la classificazione in termini biologici.

Come rappresentato in fig. 3.1 l'approccio consiste in tre macro passi: (1) determinare per ogni proteina un insieme di regioni omogenee di superficie attraverso l'utilizzo di algoritmi di clustering; (2) trovare i pattern frequenti di regioni attraverso tecniche di mining; (3) classificare le proteine in base ai pattern di superficie ottenuti al passo precedente.

La ricerca di pattern frequenti invece di singole regioni permette di trovare ampie zone simili su diverse superfici. Infatti la parte di superficie coinvolta nell'interazione tra diverse molecole raramente a valori omogenei di proprietà, per questo è richiesta la ricerca di ampie zone di superficie con, al loro interno, possibili variazioni delle proprietà. In più la ricerca di pattern di superficie tramite tecniche di mining non vincola la ricerca ai soli pattern conosciuti e annotati in database biologici e rende la classificazione finale indipendente dalla conoscenza attuale.

3.2.1. Clustering

L'approccio prende in input un set di proteine descritte secondo il formalismo del PDB e viene calcolata la superficie accessibile al solvente tramite il software MOLMOL . Potenziale, curvatura e idrofobicità sono poi calcolati per ogni punto di superficie secondo i metodi descritti. Il passo 1 utilizza questa descrizione puntuale della superficie e, tramite tecniche di clustering descritte in dettaglio nel capitolo 4, restituisce una rappresentazione compatta che consiste in un insieme di regioni omogenee e connesse di superficie. Le regioni sono poi etichettate con i valori medi delle proprietà, col valore dell'area, viene infine definito il punto medio rappresentante il baricentro e calcolato il versore della regione.

3.2.2. Mining

Le proprietà della proteina non dipendono solo dall'insieme delle regioni ottenute al passo 1 ma anche dalla loro posizione spaziale relativa. Per questo ogni proteina viene rappresentata in maniera compatta tramite un grafo di superficie dove i nodi rappresentano le regioni ottenute dal clustering e gli archi individuati dai segmenti che uniscono i baricentri delle regioni rappresentano la loro posizione relativa come descritto nel capitolo 5. Il passo successivo consiste nell'estrazione dei pattern frequenti di superficie all'interno del set di proteine. Un pattern di superficie è un sottografo del grafo di superficie che descrive la proteina, in questa maniera un pattern modella un insieme di regioni omogenee della proteina e la loro disposizione spaziale. L'estrazione di pattern frequenti da superfici molecolari rappresenta un problema poco descritto in letteratura in quanto due proteine non mostrano mai un pattern identico, per questo è necessario definire una funzione di similarità che consideri le proprietà di superficie di tutte le regioni e le loro disposizioni spaziali relative all'interno del proprio pattern. L'algoritmo di mining implementato, descritto in maniera dettagliata nel capitolo 5.4, è di tipo level-wise e permette di determinare in maniera iterativa i pattern frequenti costituiti da un numero crescente di regioni.

3.2.3. Classificazione

Nell'ultimo passo vengono utilizzate le informazioni ottenute al passo precedente allo scopo di evidenziare una classificazione delle proteine basato sulla superficie. Nel capitolo 6 verranno descritti i due approcci studiati per la classificazione: il primo di tipo gerarchico, il secondo basato sulla tecnica del simulated annealing. Nell'approccio di tipo gerarchico viene dapprima definita una funzione di similarità che valuta i pattern comuni tra coppie di proteine e viene di seguito utilizzato una tecnica gerarchica di classificazione che, a partire da cluster composti da una singola proteina, fonde progressivamente cluster simili secondo le tecniche del *complete-link, average-link, minimum-link* .

4. Clustering di superfici proteiche

Come introdotto nel capitolo 3 il clustering di superfici molecolari deve soddisfare alcuni vincoli, indicati da esperti del dominio applicativo, in modo da ottenere regioni significative dal punto di vista biologico:

1. Una regione non deve essere di dimensioni esigue per essere significativa nell'interazione tra diverse superfici. La dimensione deve essere quindi un parametro dell'algoritmo di clustering esprimibile in termini di superficie assoluta ($Å^2$) oppure in funzione del numero minimo di atomi coinvolti nella regione. Gli esperti del dominio applicativo che hanno partecipato alla ricerca hanno indicato a questo proposito che patch significative si estendono per circa 10 atomi.
2. Una regione deve presentare valori omogenei delle proprietà di superficie, questi valori potrebbero anche essere il risultato di una discretizzazione delle proprietà. In altre parole tutti i punti appartenenti ad una regione dovrebbero idealmente cadere nella stessa categoria per ogni proprietà. Per esempio una regione di potenziale elettrostatico positivo dovrà raggruppare punti etichettati come positivi, secondo una soglia parametrica. Ovviamente il potenziale di questi punti potrà variare liberamente nella fascia dei positivi.
3. Una regione non deve avere una forma estremamente irregolare. Questo vincolo, che potrebbe portare ad una minore copertura della superficie proteica, favorisce la loro rappresentazione spaziale. In più regioni di forma regolare permettono di caratterizzare meglio zone significative delle proteine aumentando la robustezza del metodo. Regioni di forma complessa saranno ottenute per composizione di regioni regolari.
4. Anche se non è richiesto di coprire l'intera superficie della proteina con regioni è indispensabile, per garantire una rappresentazione significativa, che l'unione di queste raggiunga una elevata percentuale di copertura.

Mentre il requisito (1) è considerato essere necessario nella definizione di regioni, non esistono indicazioni precise sui requisiti (2)-(4). Gli algoritmi di clustering superficiale descritti nel capitolo 4 sono stati studiati e implementati allo scopo di valutare quale dei requisiti sopra citati sono da considerare effettivamente vincoli nella definizione di regioni per gli scopi preposti dal lavoro di ricerca.

4.1. Stato dell'arte

In sez. 3.1 sono stati introdotti diversi metodi di comparazione basati su una rappresentazione compatta della superficie. Tutti questi sono finalizzati alla comparazione di proteine ma non trattano in maniera approfondita problematiche di clustering di superficie. Esiste però un numero considerevole di lavori di ricerca su algoritmi di clustering per superfici 3D, molti di questi sono conosciuti come "mesh partitioning approaches" e sono pubblicati nel campo della computer graphics. Uno di questi è la *surface simplification* che si pone l'obiettivo di ridurre la complessità della rappresentazione a mesh della superficie senza perdita considerevole di dettaglio. Molti altri approcci nel campo della mesh partitioning sono focalizzati alla soluzione del problema della scomposizione di modelli CAD allo scopo di velocizzare la ricerca di modelli all'interno di database .

Molti approcci proposti nei diversi campi della computer graphics sono di tipo *boundary-based*. Questo tipo di approcci ha come scopo la scomposizione della superficie lungo bordi delineati da un alto o basso indice di curvatura. In viene proposto il *morphological watersheed approach*, un metodo generale per la scomposizione di superfici 3D, dove la segmentazione avviene lungo i bordi definiti da nette differenze della superficie normale, in questa maniera non è necessario definire a priori dei bordi attraverso la derivata seconda della curvatura. I metodi di tipo *watersheed* fanno affidamento ad un approccio di tipo top-down e, scelto un punto iniziale, hanno la caratteristica di scendere di punto in punto secondo la maggior pendenza per la proprietà scelta, unire tutti i punti toccati all'interno della stesso cluster e terminare la discesa non appena si raggiunge un punto di minimo. Un fase di post-processing ha il compito di fondere tutti cluster che raggiungono lo stesso minimo. Wu e Levine proposero un altro interessante metodo di tipo boundary-based basato sulla simulazione della distribuzione di carica elettrostatica sulla superficie. Il loro approccio è fondato sul presupposto che, secondo la teoria della visione umana, l'uomo percepisce i punti di bordo lungo linee di curvatura negativa massima.

In generale la peculiarità degli approcci di tipo boundary-based è la scarsa possibilità di segmentazione nelle aree di bassa definizione dei bordi, infatti questi sono normalmente posizionati in regioni che presentano un massino o minimo valore di curvatura. Per questo motivo gli approcci di tipo *boundary-based* sono difficilmente applicabili in problematiche di scomposizione di superfici provenienti dal dominio biologico, infatti nelle superfici molecolari i punti di bordo che

separano regioni differenti non sono necessariamente definiti come punti di massima o minima curvatura.

Un'altra importante classe di metodi di clustering di superficie è basata sull'idea del *region growing* . Nel metodo presentato in i nodi sono prima classificati in base al loro valore discreto di curvatura, sono poi considerati semi per far crescere delle patch tramite il metodo del *region growing* e infine patch simili e adiacenti sono fuse insieme per ottenere le regioni finali. Nel metodo presentato in la suddivisione della superficie è ottenuta in tre fasi: definizione della segmentazione iniziale tramite il metodo del region growing, calcolo dei baricentri, assegnamento dei nodi ai baricentri e opzionalmente fusione dei segmenti. Un ulteriore algoritmo per la segmentazione della superficie basato sulla proprietà di curvatura fonde entrambe le tecniche del boundary-based e region-growing per ottenere la definizione finale delle regioni .

Gli algoritmi che utilizzano metodologie basate sul region growing risultano essere molto efficienti e sono largamente utilizzati nel campo della bioinformatica per risolvere problematiche relative al docking molecolare tuttavia non possone essere applicati al nostro contesto per profonde differenze derivanti dal dominio applicativo.

4.2. Rappresentazione formale della superficie proteica

Solitamente per rappresentare una superficie proteica 3D, al maggior livello di dettaglio, vengono utilizzate delle strutture dati conosciute col termine di mesh. Il loro utilizzo, ai fini del nostro approccio, è troppo dettagliato in quanto non interessa catturare piccole variazioni locali delle proprietà. Inoltre, come indicato da esperti del dominio, non si ritiene necessario suddividere le mesh superficiali associate allo stesso atomo in regioni separate, anzi è stato indicato che l'estensione delle regioni deve essere di almeno una decina di atomi per risultare significativa nelle interazioni stabili tra superfici.

In generale una rappresentazione meno dettagliata basata su atomi dà la possibilità di descrivere la superficie proteica senza eccessiva perdita di informazione e porta ad un netto miglioramento delle prestazioni nell'elaborazione al calcolatore.

Data una proteina D_i, rappresentiamo la sua superficie con un grafo connesso non-orientato:

$G^i = (V^i, E^i)$

dove ogni $v_j^i \in V^i$ rappresenta un atomo di superficie di D_i e E^i include tutti gli archi (v_j^i, v_k^i) tali che gli atomi v_j^i e v_k^i sono vicini sulla superficie di D_i. Ad ogni nodo $v_j^i \in V^i$ è associato il vettore $\mathbf{x}(v_j^i)$ delle coordinate 3D del centro dell'atomo corrispondente e con valori locali delle tre proprietà: curvatura ($cur(v_j^i)$), potenziale elettrostatico ($elp(v_j^i)$) e idrofobicità ($hyd(v_j^i)$).

4.2.1. Potenziale elettrostatico

È stato dimostrato che il potenziale elettrostatico guida fortemente i legami tra superfici nel docking molecolare , infatti le interazioni biomolecolari sono spesso risultato di legami instaurati tra superfici con potenziale complementare.

Il potenziale in v_j^i, $elp(v_j^i)$, è ottenuto mediando i valori locali di potenziale delle mesh di superficie che appartengono all'atomo v_j^i. Il potenziale elettrostatico di ogni singola mesh è stato definito, come descritto in sez 2.5.1, tramite la legge di Poisson-Boltzmann .

4.2.2. Idrofobicità

Come introdotto in sez 2.5.2 l'idrofobicità è la caratteristica chimica delle molecole che non hanno la possibilità di formare legami a ponte d'idrogeno con molecole d'acqua. In maniera opposta le molecole idrofiliche hanno questa capacità riuscendo così ad integrarsi nella struttura reticolare formata dall'acqua.

Dal punto di vista analitico non esiste ancora un accordo ben preciso di come calcolare questa caratteristica delle molecole, in letteratura sono così definite diverse scale di idrofobicità che possono essere utilizzate a seconda dei contesti di analisi.

Nel nostro approccio l'idrofobicità viene calcolata secondo la scala di Kyte-Doolittle in quanto è maggiormente utilizzata nei lavori aventi come oggetto di ricerca la superficie molecolare. Il potenziale idrofobico $hyd(v_j^i)$ di un atomo v_j^i è calcolato in base alla formula:

$$hyd(v_j^i) = h_{aa} \frac{SA_j^i}{TA_{aa}}$$

dove *aa* è l'aminoacido di v_j^i, h_{aa} è la sua idrofobicità secondo la scala di Kyte-Doolittle, SA_j^i è la sua area sulla superficie accessibile al solvente e TA_{aa} è l'area totale dell'aminoacido *aa* sulla superficie.

Nell'utilizzare le proprietà di superficie bisogna considerare che potenziale elettrostatico e idrofobicità sono caratteristiche non indipendenti. Infatti mentre le regioni con potenziale neutro possono essere idrofobiche o idrofiliche, non ha significato esprimere la carica idropatica per quelle positive o negative. Ci sono due motivi che supportano questa considerazione: (1) la forza di attrazione/repulsione data dal potenziale elettrostatico è tanto maggiore di quella fittizia di attrazione data dal potenziale idrofobico da renderla trascurabile; (2) tutti gli aminoacidi che presentano una carica positiva/negativa sono di tipo idrofilici, questo rende le regioni di superficie elettrostaticamente cariche quasi completamente idrofiliche e, di conseguenza, senza carica idrofobica.

4.2.3. Curvatura

Il valore locale di curvatura del nodo v_j^i è calcolato sulla base della superficie discreta formata dall'insieme dei nodi che hanno distanza 1 sul grafo. In è definita la *curvatura media normale* come:

$$K(v_j^i) = \frac{1}{2A}\sum_{k=1}^{|N|}\left(\left(\cot\alpha_k + \cot\beta_k\right)\left(\mathbf{x}(v_j^i)\,\mathbf{x}(nv_k)\right)\right)$$

dove:

- $N = \{nv_1,...nv_n\}$ è l'insieme degli atomi vicini di v_j^i;
- A è l'area di superficie definita dai vicini di v_j^i come indicato in
- α_k e β_k sono i due angoli opposti all'arco $(v_j^i, nv_k) \in E^i$ nei due triangoli che condividono questo arco come in fig. 4.1.

La *curvatura media* in v_j^i, $\mathbf{K}_m(v_j^i)$, è definita come la metà della norma di $\mathbf{K}(v_j^i)$. Infine il valore di $\mathbf{K}_m(v_j^i)$ viene mediato con quelli dei vicini ai fini di attenuare gli effetti del calcolo locale:

$$cur(v_j^i) = \frac{2\mathbf{K}_m(v_j^i) + \sum_{k=1}^{|N|}\mathbf{K}_m(nv_k)}{|N|+2}$$

dove $|N|$ è la cardinalità di N.

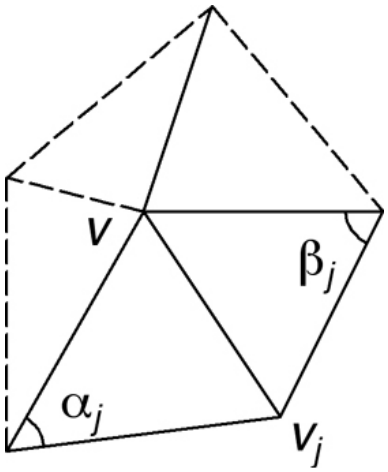

Fig 4.1*:* Nodi vicini a V e angoli opposti all'arco (V,V_j)

4.2.4. Discretizzazione delle proprietà

Tra i vincoli descritti in sez. 3, il secondo indica che l'omogeneità di una regione deve essere valutata tramite valori discreti delle proprietà di superficie. Il modo di definire valori discreti per le categorie è stato introdotto in , questo lavoro indica che tre categorie sono rilevanti per la curvatura: convesso, piano e concavo; tre categorie sono rilevanti per il potenziale: positivo, negativo e neutro; ed infine due categorie sono rilevanti per l'idropatia: idrofilico e idrofobico. Dal momento che il dominio applicativo non evidenzia una netta divisione tra le categorie, sono stati definite delle zone di indecisione (chiamate zone grigie) tra categorie contigue delle proprietà. In fig. 4.2 è mostrata la discretizzazione delle tre proprietà. La discretizzazione è guidata da 5 parametri: α_{cur} e β_{cur} sono, rispettivamente, il centro e la larghezza delle zone grigie tra "piani" e "non piani" per la proprietà di curvatura; α_{elp} e β_{elp} sono, rispettivamente, il centro e la larghezza delle zone grigie tra "neutro" e "carico" per la proprietà di potenziale elettrostatico; mentre β_{hyd} rappresenta la larghezza della zona grigia tra "idrofobico" e "idrofilico" per la proprietà di idropatia. Il significato delle zone grigie verrà chiarito con più dettaglio nelle prossime sezioni.

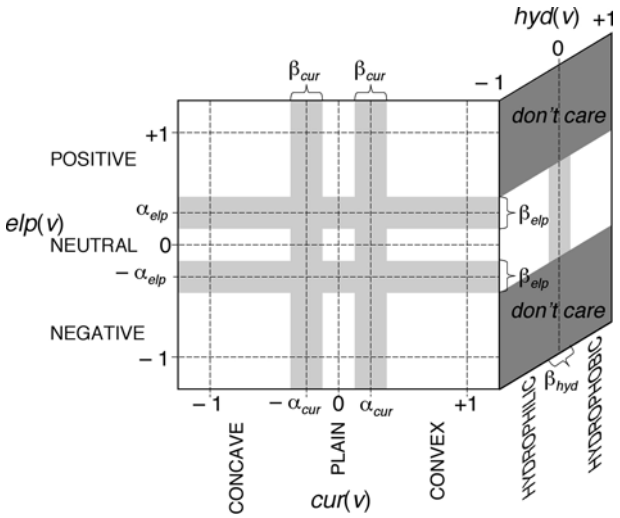

Fig 4.2: Discretizzazione per le proprietà di superficie.

In base alla discretizzazione descritta, la categoria del nodo v_j^i viene definita come segue:

$$\overline{cur}(v_j^i)=\begin{cases} concavo \text{ se } cur(v_j^i) < \alpha_{cur}+\dfrac{\beta_{cur}}{2}, \\ piano \text{ se } \alpha_{cur}-\dfrac{\beta_{cur}}{2} < cur(v_j^i) < \alpha_{cur}+\dfrac{\beta_{cur}}{2}, \\ convesso \text{ se } cur(v_j^i) > \alpha_{cur}-\dfrac{\beta_{cur}}{2}, \end{cases}$$

$$\overline{elp}(v_j^i)=\begin{cases} negativo \text{ se } elp(v_j^i) < \alpha_{elp}+\dfrac{\beta_{elp}}{2}, \\ neutro \text{ se } \alpha_{elp}-\dfrac{\beta_{elp}}{2} < elp(v_j^i) < \alpha_{elp}+\dfrac{\beta_{elp}}{2}, \\ positivo \text{ se } elp(v_j^i) > \alpha_{elp}-\dfrac{\beta_{elp}}{2}, \end{cases}$$

$$\overline{hyd}(v_j^i)=\begin{cases} idrofilico \text{ se } hyd(v_j^i) < \dfrac{\beta_{hyd}}{2}, \\ idrofobico \text{ se } hyd(v_j^i) > \dfrac{\beta_{hyd}}{2}. \end{cases}$$

I nodi che presentano una categoria definita per tutte e tre le proprietà saranno chiamati nodi bianchi, quelli che per almeno una proprietà non hanno una categoria ben definita vengono chiamati nodi grigi.

4.2.5. Funzioni obiettivo

Come già introdotto in sezione 4, nel campo biologico, non è stata delineata una singola funzione target per valutare la bontà di un clustering di superficie. D'altra

parte c'è una convergenza di idee che indicano come l'omogeneità delle proprietà, la regolarità della forma e l'alta copertura della superficie della proteina siano buone caratteristiche di valutazione delle regioni definite dal clustering. Di seguito verranno introdotti gli indici di valutazione di queste caratteristiche, che saranno poi usate nella sezione sperimentale per quantificare la qualità dei clustering ottenuti tramite gli algoritmi implementati:

- *Omogeneità:* ricordando che, come indicato dal contesto biologico, l'omogeneità delle proprietà deve essere valutata sui valori discreti, ossia sulle categorie, e non sui valori continui. Data una regione u_k^i che include m atomi, la sua categoria, presa una proprietà *f*, $\overline{f}(u_k^i)$, è definita come la categoria del valore medio di f per i nodi di u_k^i. Notare che gli algoritmi implementati generano solo regioni bianche, ovvero con valore medio per tutte le proprietà interno ad una zona bianca secondo la fig. 4.2. Il nodo v_j^i è detto compatibile con la regione u_k^i per la proprietà *f* se $\overline{f}(u_k^i) \subseteq \overline{f}(v_j^i)$, ovvero se entrambi sono bianchi e la categoria è la stessa oppure u_k^i è bianco e v_j^i cade dentro una zona grigia adiacente alla categoria di u_k^i. L'omogeneità della regione u_k^i è definita con valori nell'intervallo [0, 1] e calcolata come segue

$$hom(u_k^i) = \begin{cases} \dfrac{\#cur + \#elp + \#hyd}{3m} \text{ se } \overline{elp}(u_k^i) = neutro, \\ \dfrac{\#cur + \#elp}{2m} \text{ altrimenti.} \end{cases}$$

  dove #*f* è il numero dei nodi v_j^i di u_k^i che sono compatibili con u_k^i per la proprietà *f*. Il motivo per cui l'idrofobicità è considerata solo per regioni neutre è spiegato in sezione 2.5.2. Dato un clustering U^i, la sua omogeneità $hom(U^i)$ è calcolato come la media delle omogeneità delle sue regioni.

- *Regolarità:* la regolarità di una regione u_k^i è calcolata come la radice quadrata del rapporto tra il minimo e il massimo autovalore della matrice di covarianza della proiezione dei nodi di u_k^i su di un piano ortogonale al vettore normale della regione u_k^i. Il valore di regolarità varia nell'intervallo]0, 1] (0 indica forma irregolare mentre 1 indica forma circolare) e risulta essere inversamente proporzionale al valore di eccentricità descritto in . La regolarità del clustering U^i, $reg(U^i)$, è la media delle regolarità delle sue regioni.

- *Copertura:* la copertura del clustering U^i, *cov*(U^i), è la percentuale dei nodi della superficie della proteina che sono assegnati ad almeno una regione

4.3. Gli algoritmi proposti

Dallo studio effettuato sugli algoritmi di clustering di superficie descritti in sez 4.1 non è emerso alcun indice di riferimento per quantificarne la bontà in quanto ogni lavoro ha la necessità di ottimizzare aspetti differenti del clustering. Lo studio ha comunque fatto emergere tre indici (omogeneità, regolarità e copertura) perché, come già descritto, forniscono un buon metodo di valutazione. È chiaro che questi indici non sono ottimizzabili tutti allo stesso momento; infatti un algoritmo finalizzato a creare regioni omogenee tenderebbe a formare cluster irregolari per adattarli all'andamento dei valori di proprietà. È stato quindi necessario studiare ed implementare alcuni algoritmi focalizzati su diversi indici e studiare così quale di questi risultasse ottimale nell'analisi di superfici proteiche.

4.3.1. Variante del Region Growing

Come già descritto in sez. 4.1 gli approcci di tipo region growing sono largamente usati in letteratura per trattare il problema del clustering di superficie per le sue proprietà di velocità e semplicità di implementazione . Per i suddetti motivi, il primo approccio al clustering descritto in questo lavoro di tesi si basa su questa tecnica. Il classico algoritmo basato sul region growing non è direttamente adattabile al contesto in quanto utilizza un'unica proprietà di superficie con valori continui per definire regioni omogenee. Nel contesto descritto in sez. 4 è indispensabile che l'algoritmo di clustering sia modificato in modo da rispondere a due ulteriori richieste: (1) garantire l'omogeneità di più proprietà allo stesso tempo; (2) calcolare l'omogeneità delle regioni, utilizzando valori discreti ossia le categorie delle proprietà.

Data una proteina D_i e il suo grafo degli atomi di superficie G^i, la tecnica di clustering proposta prevede tre passi:

(1) *Inizializzazione delle regioni:* l'insieme iniziale delle regioni, $U^i = \{u_1^i,...,u_o^i\}$, è definito inserendo tutte le regioni costituite da nodi bianchi adiacenti nel grafo $G^i = (V^i, E^i)$, che mostrano le stesse categorie di curvatura, potenziale elettrostatico e idrofobicità. I nodi grigi, ovvero quelli che cadono in una zona grigia per almeno una proprietà, non vengono assegnati ad alcuna regione durante questo passo dell'algoritmo.

(2) *Region growing:* l'insieme dei nodi grigi del grafo che non sono stati assegnati ad alcuna regione esistente vengono assegnati secondo l'algoritmo. La procedura di assegnamento controlla e assegna i nodi cercando di migliorare la regolarità della forma delle regioni.

(3) *Eliminazione di regioni piccole:* tutte le regioni costituite da un numero di atomi inferiore a 10 vengono cancellate dall'insieme U^i e i suoi nodi sono etichettati come "non assegnati".

In fig. 4.3 viene mostrato il clustering ottenuto tramite questo algoritmo per la proteina 1CLL.

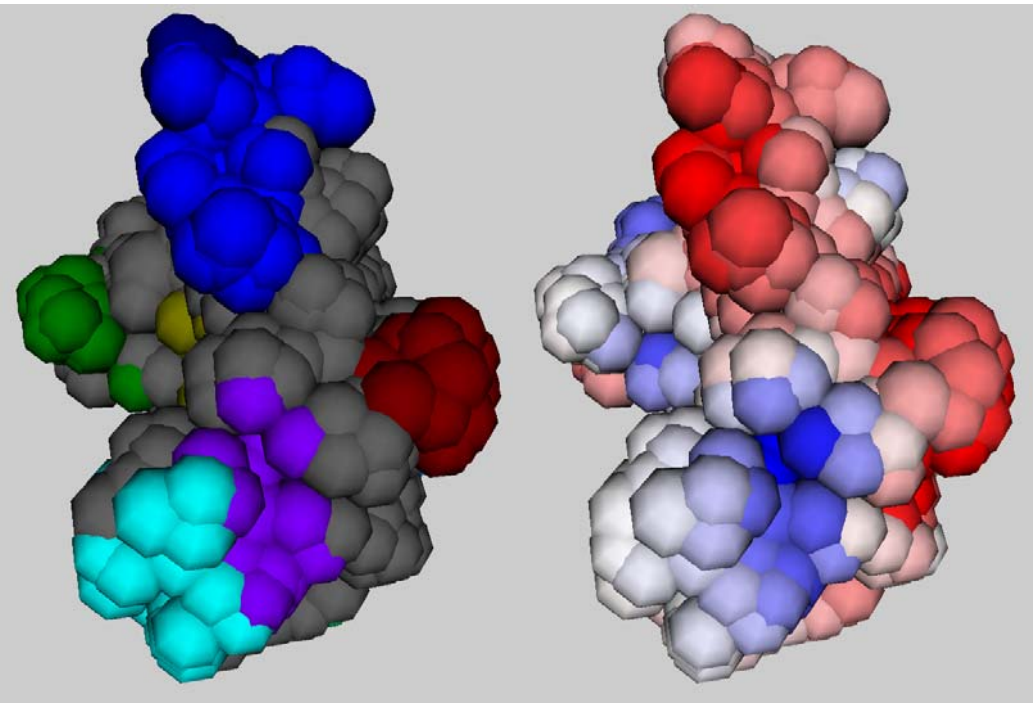

Fig 4.3: Clustering ottenuto tramite l'algoritmo region growing.

```
AssignGreyNodes(G^i, U^i) :
ripeti
{     anyAssigned =FALSE;
      per ogni v_j^i ∈ V^i non ancora assegnato esegui
      {     u_k^i = getPatch(v_j^i);
            //restituisce la regione del nodo più vicino
            //connesso a v_j^i e compatibile con le sue
proprietà
            se u_k^i !=NULL allora
            {     patch[j] = u_k^i;
                  anyAssigned =TRUE;
            }
      }
      //l'assegnamento dei nodi alle regioni viene
ritardato
      //alla fine del ciclo in modo da ridurre la
dipendenza
      //all'ordinamento dei nodi nella lista
      per ogni v_j^i ∈ V^i non ancora assegnato
            se patch[j]!=NULL allora
                  assegna v_j^i alla regione patch[v_j^i];
}     finché ! anyAssigned
```

Fig 4.4: Algoritmo di growing.

4.3.2. Template Matching

Considerando solo le proprietà di regolarità e omogeneità nella definizione di un algoritmo di clustering si otterrebbero come risultato delle regioni estremamente piccole, regioni composte da un solo atomo sarebbero ottimali per entrambe le proprietà. Questo, però, sarebbe in contrasto col vincolo (1) descritto in 4 e quindi è possibile concludere che non è possibile ottimizzare entrambe le proprietà contemporaneamente.

L'algoritmo di clustering basato sul region growing ottimizza infatti l'omogeneità delle regioni, rilassando la loro regolarità. Per studiare quale di queste proprietà è più importante nell'analisi di similarità di superfici è indispensabile definire un secondo algoritmo di clustering che privilegi la regolarità all'omogeneità. Questo secondo algoritmo, chiamato template matching, assegna i nodi del grafo a regioni aventi una forma descritta da un dato template che, in questo contesto, è di forma circolare in quanto ottimizza il vincolo di regolarità descritto in sez. 4.2.5.

La tecnica di clustering consiste in due passi:

(1) *Definizione delle regioni candidate:* una patch circolare u_k^i con centro sul nodo v_j^i e di raggio *r* è una regione che comprende tutti i nodi che hanno una distanza da v_j^i calcolata sulla superficie inferiore del raggio *r*. L'insieme delle regioni candidate U_{cand}^i comprende tutte le regioni circolari centrate su tutti i nodi del grafo che hanno una percentuale di nodi compatibili per ogni proprietà maggiore di una soglia γ. Naturalmente più regioni, di dimensioni diverse, centrate sullo stesso nodo possono essere incluse in U_{cand}^i.

(2) *Ottimizzazione:* in questa fase dell'algoritmo è necessario scegliere un sottoinsieme di regioni non sovrapposte in modo da ottimizzare la copertura della proteina. Sulla base di questa ipotesi la scelta favorirebbe la formazione di regioni piccole in quanto tendono a massimizzare l'omogeneità, la copertura e la regolarità. Per questo è stato inserito un ulteriore vincolo che porta a favorire la scelta di regioni aventi un'ampia estensione. Il problema di ottimizzazione è stato affrontato con un algoritmo esatto di programmazione binaria pura . Dato $U_{cand}^i = (uc_1^i,...,uc_t^i)$ l'insieme delle regioni candidate e m_j il numero di nodi nella regione uc_j^i; la formulazione dei vincoli per la programmazione binaria è la seguente:

$$Maximize \sum_{j=1}^{t} m_j^2 x_j$$

con: $x_j \in \{0,1\} \qquad j = 1,...,t$

$x_j + x_k \leq 1 \qquad \forall c_j, c_k; c_j \cap c_k \neq 0$

ogni variabile binaria x_j ha un valore 1 se la regione uc_j^i è nella soluzione ottimale mentre ha valore 0 nel caso contrario. Il terzo vincolo è espresso per evitare la formazione di regioni sovrapposte. Nella fase di scelta delle regioni

non viene espresso un vincolo di omogeneità, per il fatto che è già stato considerato al passo precedente nella selezione dei candidati.

Una valutazione qualitativa dell'approccio può essere effettuata tramite la fig. 4.5 che mostra il clustering ottenuto tramite questa tecnica sulla proteina 1CLL.

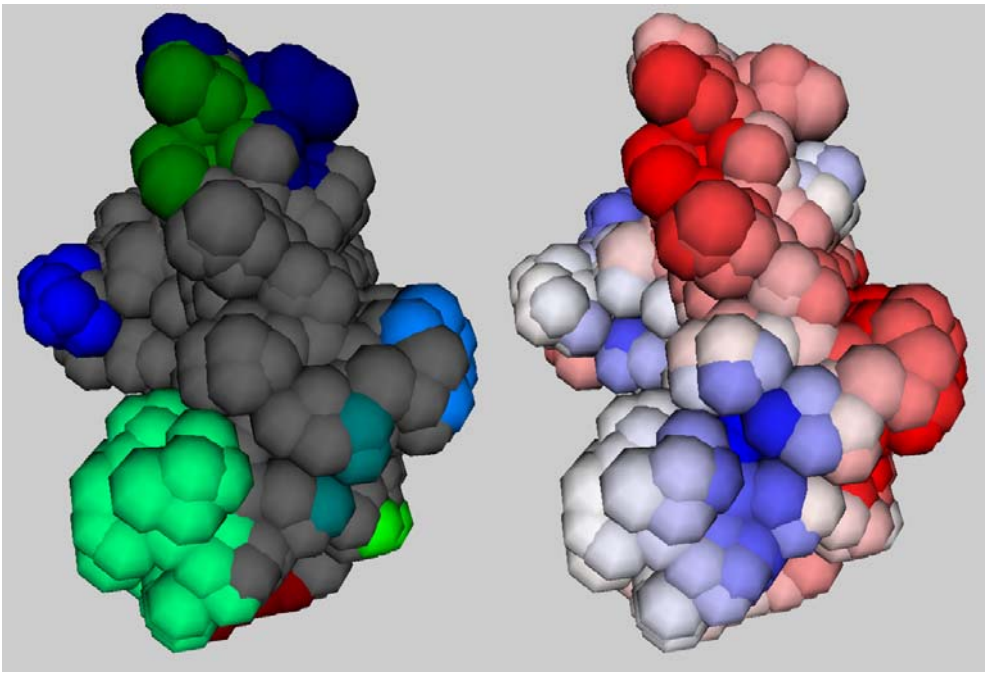

Fig 4.5: Clustering ottenuto tramite l'algoritmo template matching.

4.3.3. Overlapped Region Growing

I precedenti algoritmi introdotti nelle sez. 4.3.1 e 4.3.2 sono stati studiati per ottenere cluster ottimi per omogeneità o regolarità. Come sarà evidenziato in sez. 7.2.1, i risultati sperimentali mostrano in entrambi una difficoltà nell'ottenere una buona copertura. Risulta quindi indispensabile definire un algoritmo di clustering che conduca a coperture maggiori pur mantenendo le proprietà di omogeneità e regolarità su livelli accettabili. L'intuizione che ha portato alla definizione del terzo algoritmo è la seguente: un clustering caratterizzante della superficie proteica include tutte le zone dove gli atomi presentano valori importanti (nodi bianchi) per le proprietà superficiali considerate. Risulta indispensabile che in ognuna di queste zone cresca una regione in maniera indipendente dalle altre, ossia tutti quegli atomi grigi che si trovano ai bordi della regione bianca non siano contesi da diversi cluster. Per questo, nell'assegnazione dei nodi grigi, viene data la possibilità ad ogni atomo di appartenere a più di una regione sempre mantenendo i valori di omogeneità e regolarità all'interno delle soglie parametriche. In questo modo la regolarità e

l’omogeneità non sono proprietà in contrasto tra loro e il livello di copertura della superficie viene aumentato in quanto le regioni sono libere di crescere in maniera indipendente.

La tecnica di clustering è costituita da tre passi:

(1) *Inizializzazione delle regioni:* l’insieme iniziale delle regioni, $U^i = \{u_1^i,...,u_o^i\}$, è definito inserendo tutte le regioni costituite da nodi bianchi adiacenti nel grafo $G^i = (V^i, E^i)$, che mostrano le stesse categorie di curvatura, potenziale elettrostatico e idrofobicità. I nodi grigi, ovvero quelli che cadono in una zona grigia per almeno una proprietà, non vengono assegnati ad alcuna regione durante questo passo dell’algoritmo.

(2) *Growing delle regioni:* una regione alla volta è sottoposta alla fase di growing. Questa fase consiste nell’assegnare in maniera iterativa gli atomi circostanti alla regione fino a quando i valori medi di omogeneità e regolarità rimangono interni a delle soglie di ammissibilità.

(3) *Fusione:* per ogni coppia di regioni viene valutato la percentuale di sovrapposizione. Per ogni coppia di regioni che presenta un valore di sovrapposizione maggiore di 0 viene valutata la fusione e se questa rimane all’interno delle soglie di omogeneità e regolarità allora la nuova regione viene inserita nel clustering e la due fuse vengono cancellate. È evidente che per bassi valori di sovrapposizione la potenziale regione fusa difficilmente rispetterà il vincolo di regolarità, nell’algoritmo è presente un controllo che esclude questi casi per aumentare l’efficienza dell’algoritmo.

In fig. 4.6 è mostrato il clustering ottenuto sulla proteina 1CLL ottenuto con l’overlapped region growing.

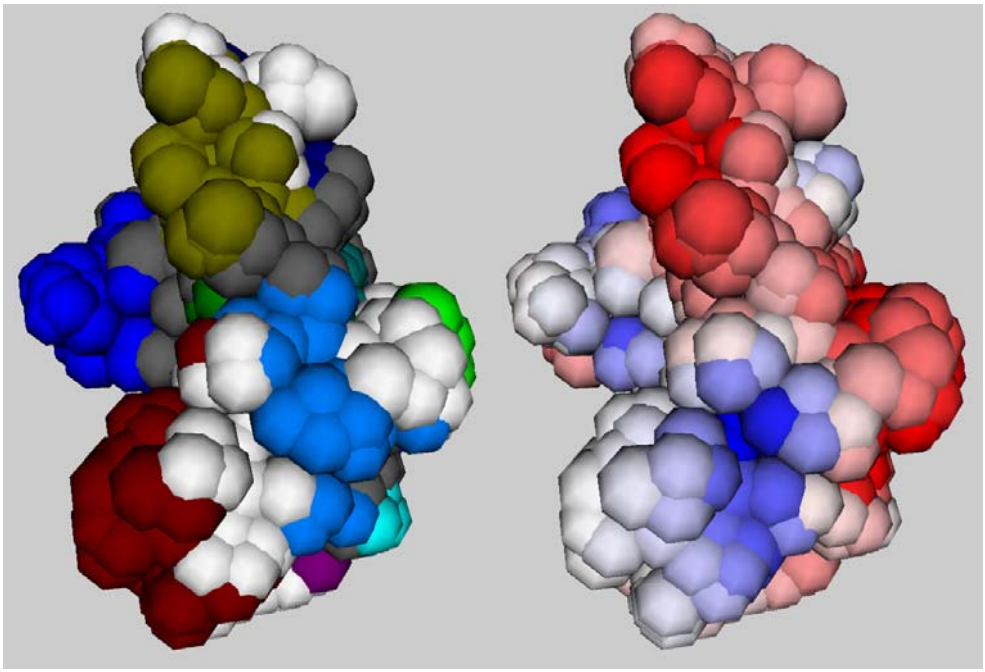

Fig 4.6: Clustering ottenuto tramite l'algoritmo overlapped region growing.

5. Mining di pattern complessi da superfici proteiche

5.1. Stato dell'arte

Il recente aumento di volume dati disponibili in ambito biologico ha determinato la necessità di estrarre in maniera automatica la conoscenza ivi contenuta. Risulta pertanto naturale applicare tecniche di Knowledge Discovery in Databases (KDD) e di data mining. Il termine *data mining* è utilizzato in alternativa a KDD ed è definito come il processo di ricerca di pattern su grandi volumi di dati utilizzando diverse tecniche di estrazione quali: *statistica, information retrieval, machine learning* e *pattern recognition*. Per pattern si intende una qualsiasi espressione, in un opportuno linguaggio, capace di descrivere in modo sintetico informazioni generali relative ad un insieme di dati interessanti. Queste informazioni sono strutturate in modo da mettere in evidenza caratteristiche di regolarità presenti nei database, oppure più semplicemente caratteristiche di alto livello non presenti tra i dati nella forma in cui vengono forniti. Per pattern frequente si intende un sottoinsieme degli item del database il cui numero di presenze nelle transazioni è superiore ad una determinata soglia.

Le tecniche di data mining possono essere classificate in base al tipo di pattern estratto, esistono infatti tecniche per la ricerca di pattern sequenziali , regole associative, correlazioni , casualità , episodi , pattern multi-dimensionali e pattern massimali .

Uno specifico ambito di ricerca del data mining è conosciuto in letteratura come *constrained pattern mining* ossia l'estrazione di pattern che rispettino dei vincoli sui dati.

5.2. Ricerca di pattern frequenti

Sia $I = \{i_1, i_2, ..., i_m\}$ un insieme item e $D = \{T_1, ...T_n\}$ un insieme di transazioni, dove ogni transazione T è un insieme di item tale che $T \subseteq I$. Si dice che una transazione T contiene X, con $X \subseteq I$, se $X \subseteq T$. Un insieme X di $k-$item, con $k = (1,...,k)$, $k \leq m$, è chiamato $k-$itemset. La restrizione $D|_X$ rappresenta un

sottoinsieme di D tale che le transazioni T contenute nella restrizione contengono l'insieme X, ovvero

$$D|_X = \{T_i \in D \mid i \le n \wedge X \subseteq T_i\}$$

Una *regola associativa* è un'implicazione nella forma $X \Rightarrow Y$, dove $X \subset I, Y \subset I,$ e $X \cap Y = \varnothing$. Il supporto s di una regola associativa $X \Rightarrow Y$ indica il rapporto tra il numero di transazioni che contengono l'insieme $X \cup Y$ e il numero totale di transazioni, ovvero

$$s = \frac{|D|_{X \cup Y}|}{|D|}$$

il supporto verrà indicato in seguito come $s = \sigma(X \cup Y)$. La confidenza c di una regola associativa $X \Rightarrow Y$ indica il rapporto tra il supporto di $X \cup Y$ e il supporto di X, ovvero

$$c = \frac{\sigma(X \cup Y)}{\sigma(X)} = \varphi(X \mid Y)$$

Una regola associativa $X \Rightarrow Y$ si dice frequente se il proprio supporto $\sigma(X \cup Y)$ risulta essere maggiore o uguale di una data soglia `minSupp`. Una regola associativa $X \Rightarrow Y$ si dice forte se la sua confidenza $\varphi(X \mid Y)$ risulta essere maggiore o uguale di una data soglia `minConf`.

5.2.1. Ricerca levelwise di pattern frequenti

La ricerca di tipo levelwise di pattern frequenti viene utilizzata da molti lavori proposti in letteratura, fu introdotta da Mannila e Toivonen [40] ed è basata su una ricerca in ampiezza dello spazio di ricerca. In questo lavoro viene introdotta una relazione d'ordine tra pattern per indicarne la generalità, dati due pattern P_1 e P_2, indichiamo $P_1 \le P_2$ per denotare che P_1 è più generale di P_2 o, equivalentemente, che P_2 è più specifico di P_1. In fig. 5.1 viene mostrato lo spazio di ricerca dei pattern dato l'insieme di item $I = \{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$, risulta chiaro che il pattern più generico è $\varnothing$ mentre quello più specifico è I stesso.

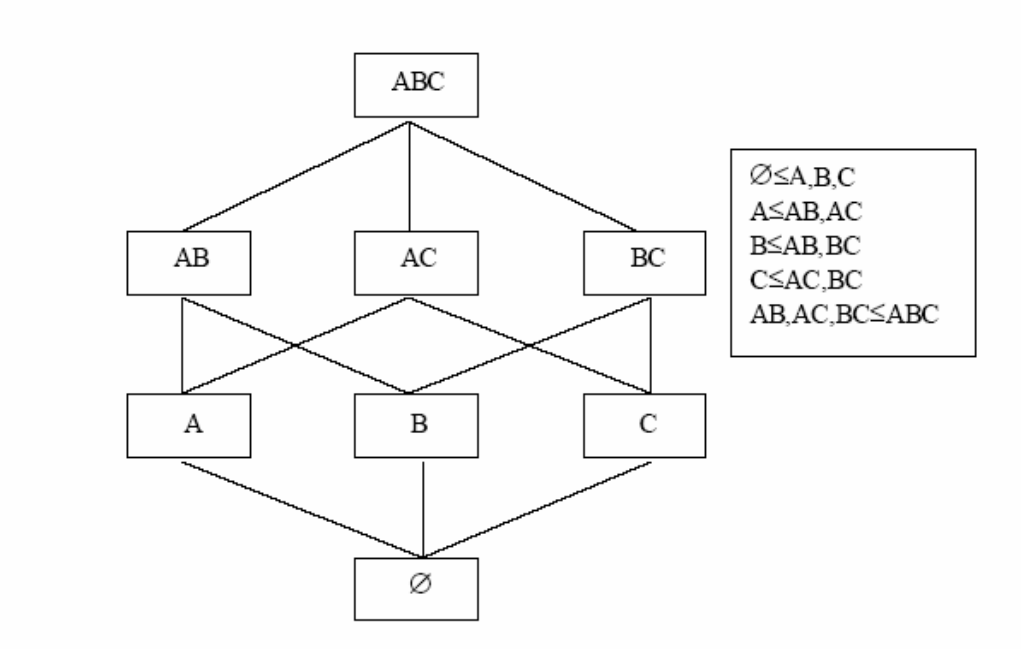

Fig 5.1: Struttura del reticolo costruito a partire dagli item A,B e C

Mannila e Toivonen [40] hanno anche enunciato e dimostrato la proprietà di monotonicità dei pattern frequenti, ovvero:

Definizione: Sia $P \subseteq I$ frequente $\Rightarrow \forall X \subseteq P$ è anch'esso frequente.

Ciò significa che, secondo la fig. 5.1, se il pattern $\{A,C\}$ è frequente allora lo sono anche $\{A\}$ e $\{C\}$ o, equivalentemente, se il pattern $\{B\}$ non è frequente, non lo è nessuno tra le sue specializzazioni $\{A,B\}$, $\{B,C\}$ e naturalmente $\{A,B,C\}$.

La proprietà di monotonicità permette di diminuire lo spazio di ricerca del problema, infatti nella ricerca levelwise viene esplorato un livello alla volta, partendo dai pattern più generali per arrivare man mano a quelli più specifici. L'estrazione di pattern ad un certo livello k consiste, inizialmente, nella generazione dei pattern candidati partendo da quelli di livello $k-1$, poi si passa alla verifica del vincolo di frequenza per ogni candidato. La generazione dei candidati esclude direttamente i pattern più specifici composti da sottopattern non frequenti, effettuando così la potatura delle zone del reticolo che non presentano soluzioni al problema.

La ricerca levelwise presenta inoltre altri pregi:

- Il numero di iterazioni sul database è al più pari a $k+1$, dove k è il massimo livello dei pattern frequenti;
- La complessità dell'algoritmo di estrazione è proporzionale al numero di pattern frequenti.

L’approccio levelwise restituisce un insieme di pattern considerato frequente secondo una soglia data in input. Tali pattern sono poi utilizzati per ottenere l’insieme delle regole associative come descritto nella prossima sezione.

5.2.2. Calcolo orizzontale e verticale del supporto

Un approccio comune alla determinazione del valore del supporto di un itemset candidato è quello di contare direttamente le sue occorrenze nella base di dati transazionale. A questo proposito, viene definito e inizializzato a zero un contatore per ogni candidato. Avviene dunque una scansione di tutte le transazioni del DB ed ogni volta che uno dei candidati viene riconosciuto come sottoinsieme di una qualche transazione, il suo contatore viene incrementato. Tipicamente non vengono generati tutti i sottoinsiemi di ogni transazione, ma solo quelli che sono contenuti in un qualche candidato o che comunque hanno il prefisso in comune con almeno due itemset frequenti del passo precedente. Questa strategia, conta su una rappresentazione orizzontale del database, ovvero vede ogni transazione come l’insieme degli item che contiene .

Un *approccio ortogonale*, è quello di determinare il valore del supporto di ogni candidato, per mezzo dell’intersezione insiemistica. Ogni transazione è identificata univocamente dal proprio tId . Definiamo la $tIdList$ di ogni singolo item come l’insieme degli identificatori corrispondenti alle transazioni contenenti tale item. In accordo con quanto detto, una $tIdList$ esiste anche per un generico itemset X e viene denotata come $X.tIdList$. La $tIdList$ di un generico candidato $s = X \cup Y$ altro non è che l’intersezione $X.tIdList \cap Y.tIdList$ ed il valore del supporto si ottiene banalmente determinando la cardinalità dell’insieme $|s.tIdList|$. Contrapponendosi alla precedente, questa strategia fa uso di una rappresentazione verticale del database, ovvero memorizza ogni item associandolo alla lista delle transazioni in cui esso è contenuto.

Il primo orientamento è più adatto a domini applicativi sparsi, ossia con dimensione dei pattern ridotta, poiché in questi il numero di accessi al DB è comunque limitato, mentre il secondo è più adatto per domini densi.

5.2.3. L’algoritmo Apriori

L’algoritmo Apriori [20] è un algoritmo di data mining di tipo levelwise ed è stato introdotto nel 1994 da Agrawal, tutt’oggi risulta comunque essere un punto di

riferimento per chi affronta il problema della ricerca di regole associative all'interno di grandi database. Apriori è stato studiato allo scopo di analizzare le vendite di prodotti ed estrarre le abitudini dei clienti nascoste implicitamente nelle transazioni di vendita.

Tab. 5.1*:* Notazioni

$k-$itemset	Insieme di item contenente k elementi.
L_k	Insieme di $k-$itemset frequenti. Ognuno è etichettato con il conteggio del supporto.
C_k	Insieme di $k-$itemset candidati. Ognuno è etichettato con il conteggio del supporto.

In fig. 5.2 viene riportato l'algoritmo Apriori. In riga 1 viene creato L_1 semplicemente contando le occorrenze all'interno del database per determinare l'insieme dei 1-itemset, ovvero i singoli item con una presenza all'interno delle transazioni maggiore di `minsupp`. Al generico passo *k*, viene creato l'insieme C_k utilizzando L_{k-1}, l'algoritmo di generazione dei pattern candidati `apriori-gen` è riportato in fig. 5.3. L'insieme L_k è composto di tutti gli $k-$itemset $\in C_k$ che risultano frequenti.

```
1)   L1 = {large 1-itemsets};
2)   for ( k = 2; Lk-1 ≠ ∅; k++ ) do begin
3)           Ck = apriori-gen(Lk-1); // New candidates
4)           forall transactions t ∈ D do begin
5)                   Ct = subset (Ck, t); // Candidates Contained in t
6)                   forall candidates c ∈ Ct do
7)                           c.count++;
8)           end
9)           Lk = {C ∈ Ck | C.count ≥ minsup}
10)  end
11)  Answer = Uk Lk;
```

Fig. 5.2: L'algoritmo Apriori

La funzione `apriori-gen` prende come argomento L_{k-1} e restituisce C_k, è costituita da due passi: (1) nel primo viene effettuato un join tra l'insieme L_{k-1} con L_{k-1};

```
apriori-gen(Lk-1){
  insert into Ck
  select p.item1, p.item2, ..., p.itemk-1, q.itemk-1,
  from Lk-1 p, Lk-1 q
  where p.item1 = q.item1, ..., p.itemk-2 = q.item k-2, p.itemk-1 <
q.itemk-1;
  return Ck
}
```

Fig. 5.3*:* Inserimento nei candidati

(2) mentre nel secondo vengono cancellati tutti i $k-$itemset generati che presentano dei sottoinsiemi di dimensione $k-1$ non appartenenti a L_{k-1}.

```
1.  forall itemsets c ∈ Ck do
2.          forall (k-1)-subset s of c do
3.                  if ( s ∉ Lk-1) then
4.                      delete c from Ck
```

Fig. 5.4*:* Eliminazione candidati

5.3. Mining di superfici proteiche

5.3.1. Rappresentazione a grafo della superficie proteica

Le caratteristiche delle proteine non dipendono solo dall'insieme delle ULR ottenute dal clustering di superficie ma anche dalla loro posizione relativa e orientazione. Viene proposta così una rappresentazione sintetica della proteina che consiste in un grafo 3D completamente connesso. Una proteina D_i è rappresentata in maniera compatta come segue:

Definizione: Data una proteina $D_i \in D$, il suo grafo di superficie è definito come il grafo non diretto e completamente connesso $S^i = (U^i, F^i)$ dove ogni nodo $u_j^i \in U^i$ rappresenta una ULR ottenuta tramite il clustering. Ogni ULR $u_j^i \in U^i$ è etichettata con i valori delle tre proprietà di superficie, chiamate $\overline{cur}(u_j^i)$, $\overline{elp}(u_j^i)$, $\overline{hyd}(u_j^i)$ e la sua area $sur(u_j^i)$.L'arco $(u_j^i, u_k^i) \in F^i$ è etichettato con la lunghezza γ del vettore che connette i baricentri di u_j^i e u_k^i e con i tre angoli α_1, α_2 e β che esprimono la posizione relativa e l'orientazione di u_j^i e u_k^i nello spazio 3D.

In figura 5.5 (a) è mostrato il grafo delle ULR che permette di rappresentare la proteina in maniera sintetica e le informazioni sulle etichette di nodi e archi. In figura 5.5 (b) viene mostrato come è descritta la posizione relativa di una coppia di ULR tramite:

(1) la lunghezza γ dell'arco (u_j^i, u_k^i) che unisce i baricentri delle due ULR;

(2) gli angoli α_1 e α_2 che l'arco (u_j^i, u_k^i) forma con i versori delle due regioni;

(3) l'angolo β formato dalla proiezione dei due versori su di un piano ortogonale all' arco (u_j^i, u_k^i).

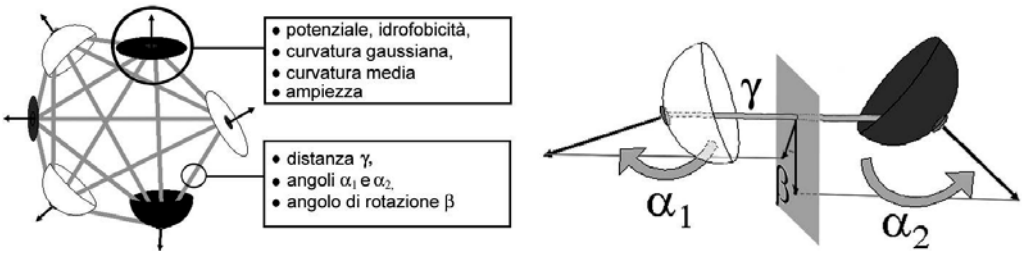

Fig. 5.5: (a) Grafo delle regioni di una proteina; (b) Informazioni necessarie a descrivere la posizione relativa di due regioni

Un pattern P_j^i appartenente alla proteina D_i è definito come un sottografo completamente connesso del grafo che ne rappresenta sinteticamente la superficie, viene chiamato livello di un pattern il numero di regioni che lo costituiscono.

5.3.2. La soluzione proposta

Il problema del mining di pattern complessi da superfici proteiche non può essere affrontato direttamente con le soluzioni proposte in letteratura in quanto il dominio applicativo richiede la definizione di soluzioni originali per fronteggiare le seguenti specificità:

- *relazione di similarità tra pattern*: è indispensabile definire una funzione di similarità per decretare l'appartenenza di un pattern al supporto di un altro in quanto non esistono pattern di riferimento ma, contrariamente, ogni pattern è unico all'interno del DB. La relazione di similarità tra pattern è per definizione non transitiva, ovvero $P \sim Q \wedge Q \sim S \not\Rightarrow P \sim S$;
- *presenza di vincoli spaziali tra le regioni di un pattern*: parte delle informazioni che caratterizzano i pattern non sta sulle singole regioni bensì sugli archi, ossia i pattern sono caratterizzati anche dalla disposizione spaziale delle regioni.

Da queste considerazioni si evince che l'approccio al mining di tipo orizzontale non può essere direttamente applicato in questo contesto dal momento che non è sufficiente un contatore per mantenere il supporto di un pattern. Equivalentemente anche memorizzando l'intero supporto di un pattern, come nell'approccio verticale, non sarebbe sufficiente a ridurre il numero di accessi al DB per il problema introdotto dalla similarità. Infine si noti che le informazioni sugli archi sottendono una metrica Euclidea e, pertanto, la rappresentazione tramite grafo completamente connesso risulta ridondante. Di conseguenza, la ricerca di pattern frequenti non verrà affrontata come mining di sottografi [44] che, richiedendo l'uso di tecniche di determinazione di isomorfismi, renderebbe il problema intrattabile all'aumentare delle dimensioni degli oggetti [18, 45].

In fig. 5.6 viene riportato il ciclo principale dell'algoritmo proposto: la procedura `GetFrequentRegions`(D_i) restituisce l'insieme $\mathcal{L}_1^i$ delle singole regioni frequenti, 1-itemset, della proteina D_i, mentre `MineFrequentPatterns`($\mathcal{L}_1^i$) effettua il processo di mining vero e proprio.

```
1.  for each D_i ∈ D
2.  {
3.      L_1^i =GetFrequentRegions( D_i );
4.      MineFrequentPatterns( L_1^i );
5.  }
```

Fig 5.6*:* Ciclo principale di Mining

La non transitività della similarità tra pattern rende il problema del mining computazionalmente più complesso in quanto il risultato dipende dal pattern preso come riferimento, questo, però, consente di utilizzare in `MineFrequentPatterns`($\mathcal{L}_1^i$) una maggior quantità di strutture dati accessorie dato che il numero di pattern frequenti per ogni proteina sarà ovviamente molto limitato rispetto a quello dell'intero DB.

```
1.  MineFrequentPatterns( L_1^i )
2.  {   for  (k = 2, L_{k-1}^i ≠ ∅ , k++) do
3.      {   C  = CandidatePatterns( L_{k-1}^i );
```

```
4.          for each D_j ∈ D do
5.                  for each P^i ∈ C ; D_j ∈ Prot(P^i) do
6.                  {    for each Q^j ∈ Supp_j(P^i) do
7.                           if    Simil(Q^j, P^i) < σ
8.                                 Supp(P^i)\ = {Q^j}
9.                  }
10.           L^i_k = {P^i ∈ C;| Supp(P^i) |≥ min Supp}
11.       }
12. }
```

Figura 5.7*:* Algoritmo di Mining

In fig 5.7 è riportata la procedura `MineFrequentPatterns`$(\mathcal{L}_1^i)$, mentre in tabella 5.1 è riassunta la legenda dei simboli per facilitarne l'interpretazione.

Tab. 5.1: Leggenda dei simboli utilizzati nell'algoritmo.

$D = \{D_1, ..., D_n\}$	Database delle proteine
$\mathcal{L}_1^i$	Pattern frequenti di livello *k* della proteina D_i
C	Insieme dei pattern candidati
$P^i = (P_1^i, \ldots, P_k^i)$	Pattern di livello *k* della proteina D_i
$Supp(P^i)$	Pattern che formano il supporto del pattern P^i
$Supp_j(P^i)$	Sottoinsieme dei pattern in $Supp(P^i)$ che appartengono a D_j
$prot(P^i)$	Insieme di proteine cui mostrano almeno un pattern in $Supp(P^i)$
σ	Soglia di similarità

Negli algoritmi sono state usate le lettere maiuscole da P a T per denotare i pattern e le corrispondenti minuscole per denotare le loro regioni; per evidenziare il fatto che un pattern/regione appartiene alla proteina D_i, viene aggiunto l'apice i. L'algoritmo procede iterativamente generando a ogni passo l'insieme $\mathcal{L}_i^k$ dei pattern frequenti di D_i di livello k via via crescente (*riga 2*); l'output di un passo rappresenta l'input del passo successivo. Per motivi prestazionali, l'algoritmo lavora su una rappresentazione semplificata dei pattern che consiste in una sequenza di puntatori alle regioni che ne fanno parte. Per questo motivo, dopo aver generato l'insieme C dei pattern potenzialmente frequenti, è necessario accedere al DB per

verificare se effettivamente il vincolo di similarità è soddisfatto (*riga 7*): la funzione `Simil`(P,Q) carica P e Q dal DB e ne calcola la similarità, tenendo conto delle caratteristiche di superficie e relazioni spaziali tra le regioni. Si noti a questo proposito come l'accesso al DB sia ottimizzato recuperando le sole proteine che sono presenti nel supporto dei pattern candidati (*righe 4-6*). Vengono inseriti tra i pattern frequenti quelli che presentano un supporto di cardinalità superiore alla soglia minSupp (*riga 10*).

1. **CandidatePatterns(** $\mathcal{L}^{i}_{k-1}$ **)**
2. { $C = \varnothing$
3. **for each** $P^i, Q^i \in \mathcal{L}^i_{k-1}$; **Mergeable(** P^i, Q^i **)** $\wedge$ | Prot(P^i) $\cap$ Prot(Q^i) | $\geq$ *minsupp*
4. {
5. $T^i = (p^i_1, \ldots, p^i_{k-1}, q^i_{k-1})$;
6. $Supp(T^i) = \{(r^1_1, \ldots, r^l_{k-1}, s^l_{k-1})$;
7. con $R^l \in Supp(P^i) \wedge S^l \in Supp(P^i) \wedge$ **Mergeable(** R^l, S^l **)**}
8. **if (** $| Supp(T^i) | \geq$ *minsupp* **)**
9. $C \cup = \{T^i\}$;
10. }
11. return *C*;
12. }

Fig. 5.8*:* Algoritmo di generazione dei pattern candidati

La parte cruciale dell'algoritmo è rappresentata dalla procedura `CandidatePatterns`$(\mathcal{L}^i_{k-1})$, presentato in fig. 5.8, che genera i pattern candidati di livello k fondendo coppie di pattern compatibili di livello $k-1$.

L'algoritmo considera tutte le possibili coppie di pattern in input (*riga 2*) ed effettua una prima scrematura dei candidati sulla base dei seguenti vincoli:

- *Corrispondenza delle regioni*: per ottenere pattern di livello k è necessario fondere due pattern che condividano $k-2$ regioni. Questo vincolo viene verificato dalla procedura `Mergeable`(P^i, Q^i), presentata in fig. 5.9. Questa procedura garantisce inoltre la generazione non ridondante dei pattern imponendo un ordinamento lessicografico tra le regioni.
- *Controllo della cardinalità del supporto*: se il numero di proteine che presentano

pattern nel supporto dei genitori del candidato è inferiore a minSupp allora il candidato non potrà essere frequente (*riga 3*).

```
1.      Mergeable( P^i , Q^i )
2.      {   if   ( p_1^i = q_1^i ∧ … ∧ p_{k-2}^i = q_{k-2}^i ∧ p_{k-1}^i < q_{k-1}^i )
3.                   return TRUE;
4.          else     return FALSE;
5.      }
```

Fig 5.9*:* Algoritmo di verifica della compatibilità tra pattern

Le coppie che soddisfano le precedenti proprietà vengono fuse (*riga 5*) per generare un pattern di livello k e viene calcolato il relativo supporto (*riga 6*). Si utilizza la rappresentazione esplicita del supporto, tipica degli algoritmi di mining verticali, per evitare di accedere al DB anche in fase di generazione dei candidati e per evitare il calcolo di isomorfismi tra pattern. In questa rappresentazione non ci si limita a indicare l'insieme di proteine in cui compare un pattern simile a quello in esame, ma si specifica direttamente l'insieme di sequenze di regioni che li compone rendendo possibile la verifica della corrispondenza sulla base degli identificatori. Si noti tuttavia che l'accesso al DB è comunque necessario in `MineFrequentPatterns`$(\mathcal{L}_k^i)$ poiché la similarità tra i pattern candidati e il loro supporto, garantita a livello $k-1$, non implica quella a livello k [19].

La tecnica di mining presentata in questo capitolo permette effettivamente l'individuazione di pattern ricorrenti su superfici proteiche, il suo risultato rappresenta il punto di partenza per poter effettuare una classificazione significativa delle proteine sulla base delle caratteristiche superficiali. Si noti inoltre che l'algoritmo implementato può essere applicato a tutte quelle problematiche che richiedono evidenziare similarità su superfici con valori puntuali di proprietà.

5.3.3. La funzione di similarità tra pattern

Data una proteina D^i e il suo insieme di regioni $U^i = \{u_1^i, ..., u_o^i\}$, l'algoritmo di mining rileva i tutti pattern frequenti sulle altre superfici proteiche del database. Per verificare la frequenza di un pattern P^i è indispensabile calcolare il suo supporto. Questo calcolo consiste nel confrontare P^i con tutti i potenziali pattern simili che si trovano sulle altre superfici proteiche. Questo confronto viene effettuato tramite una

funzione di similarità che valuta le qualità delle regioni che costituiscono il pattern e la loro posizione relativa. Nell'algoritmo di mining descritto in fig. 5.7 questa valutazione viene effettuata in riga 7.

Presi $P^i \subseteq D^i$ e $P^j \subseteq D^j$ due pattern aventi la stessa dimensione dove $P^i = \{rp_1^i,...,rp_d^i\}$ e $P^j = \{rp_1^j,...,rp_d^j\}$ rappresentano le regioni che li costituiscono, si dice che P^i e P^j sono simili se entrambe le seguenti condizioni sono verificate:

- per ogni coppia di regioni rp_k^i e rp_k^j la funzione $simRegion(rp_k^i, rp_k^j) = \text{true}$,
- la funzione $bestMatch(P^i, P^j) > \text{thrMatch}$.

dove:

$$simRegion(rp_k^i, rp_k^j) = \begin{cases} \text{true se } \left(\overline{cur}(rp_k^i) = \overline{cur}(rp_k^j)\right) \wedge \left(\overline{elp}(rp_k^i) = \overline{elp}(rp_k^j)\right) \wedge \\ \quad \wedge \left(\overline{hyd}(rp_k^i) = \overline{hyd}(rp_k^j)\right) \wedge \left(\dfrac{\min(sur(rp_k^i), sur(rp_k^j))}{\max(sur(rp_k^i), sur(rp_k^j))} > \text{thrArea}\right) \\ \text{false atrimenti} \end{cases}$$

in altre parole $simRegion(rp_k^i, rp_k^j)$ restituisce true se tutte le proprietà discrete delle regioni sono uguali e il rapporto tra le aree è superiore ad una certa soglia. Mentre La funzione $bestMatch(P^i, P^j)$ restituisce l'errore quadratico medio del miglior allineamento dei baricentri delle regioni che costituiscono i due pattern. L'allineamento dei baricentri è calcolato risolvendo il problema dei minimi quadrati come descritto in sez. 5.3.4.

La funzione di similarità definita in questa sez. permette di valutare l'allineamento di due pattern verificando anche la compatibilità delle proprietà chimicho-fisiche. Questa funzione necessita di due soglie: thrArea che pone un vincolo sulla differenza delle estensioni delle regioni valutate; thrMatch che pone un vincolo sulla differenza della superimposizione dei baricentri.

5.3.4. Allineamento di pattern

Dati due insiemi di punti $P = \{p_1,..., p_n\}$ e $Q = \{q_1,..., q_n\}$ il problema dell'allineamento di corpi rigidi consiste nel trovare un vettore di traslazione t e una matrice di rotazione R tali che:

$\sum_{i=1}^{n} \| p_i - (Rq_i + t) \|^2$ è minimizzata.

Teorema: se (R,t) è la trasformazione ottimale allora i punti $\{p_i\}$ e $\{Rq_i + t\}$ hanno lo stesso baricentro.

Dal teorema ne deriva che il vettore di traslazione t risulta essere uguale al vettore che unisce i baricentri dei due insiemi e il problema indicato dalla precedente formula si riduce alla seguente:

$\sum_{i=1}^{n} \| p_i' - Rq_i' \|^2$ è minimizzata.

dove p_i' e q_i' sono i punti degli insiemi P e Q con i relativi baricentri sono stati traslati nell'origine.

La matrice R può essere trovata risolvendo il problema dei minimi quadrati, qui di seguito verrà riportata la soluzione utilizzando la scomposizione a valori singolari (SVD) della matrice di covarianza H. È possibile ottenere la matrice R tramite i seguenti tre passi:

- calcolare la matrice di covarianza

  $H = \sum_{i=i}^{n} q_i' \, p_i'^{T}$

- effettuare la SVD della matrice H

  $H = U \Sigma V^T$

- la matrice R si ottiene

  $R = VU^T$

6. Classificazione

Il problema della classificazione è stato studiato in molti contesti e da molti ricercatori in ogni disciplina scientifica, questo ha portato alla definizione di innumerevoli approcci. In fig. 6.1 è rappresentata una possibile tassonomia, introdotta da Jain in , delle tecniche di classificazione. A livello più alto c'è una distinzione tra approcci gerarchici e partizionali, i primi restituiscono una serie di classificazioni mentre i secondi ne restituiscono solo una. Altri importanti aspetti dei metodi di classificazione sono elencati di seguito:

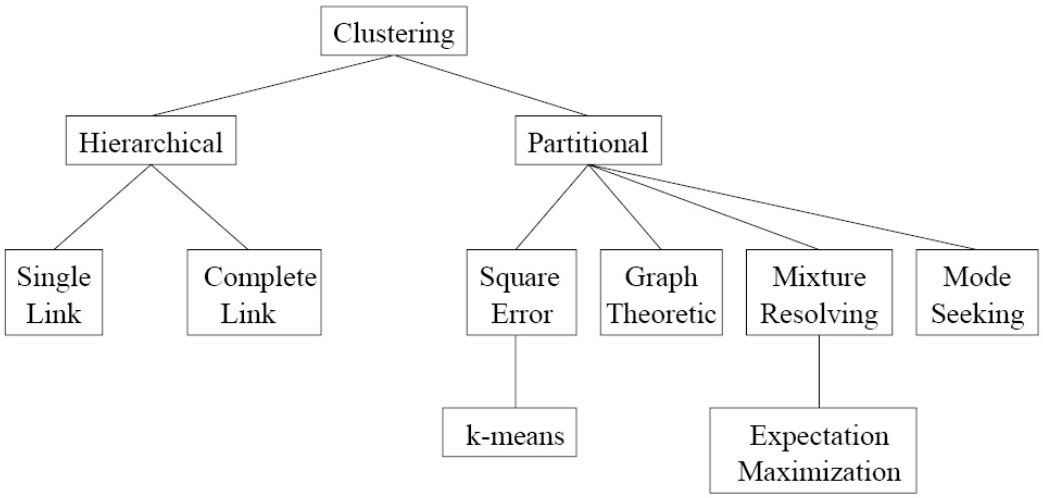

Fig 6.1: Tassonomia degli approcci di classificazione

- *Agglolmerativi* vs. *divisivi*: questo aspetto riguarda il tipo di operazione effettuata dall'algoritmo di clustering per definire il dendrogramma gerarchico. Il punto di partenza di un algoritmo agglomerativo è dato dalla classificazione in cui ogni singolo oggetto risiede in un cluster (*singleton*), successivamente due o più di cluster vengono fuse insieme in maniera iterativa fino a quando il criterio di stop viene soddisfatto. D'altra parte un metodo di tipo divisivo inizia con tutti gli oggetti contenuti in un singolo cluster e vengono compiute divisioni in maniera iterativa fino a quando, anche in questo caso, viene soddisfatto il criterio di stop.
- *Monoproprietà* vs *multiproprietà*: questo aspetto riguarda l'uso sequenziale o simultaneo di proprietà nel processo di clustering. Negli algoritmi di tipo multiproprietà tutte le proprietà vengono considerate contemporaneamente nella

funzione di valutazione delle distanze tra cluster. Al contrario negli algoritmi di tipo monoproprietà le proprietà vengono considerate in sequenza al fine di dividere l'insieme degli oggetti. In fig 6.2 viene illustrato il comportamento di un algoritmo di tipo monoproprietà, si noti come, inizialmente, l'insieme sia suddiviso in due gruppi tramite la proprietà x_1, la linea verticale V rappresenta la prima divisione. In seguito ciascuno dei cluster è suddiviso utilizzando la proprietà x_2, questo significa che sono necessarie due linee di divisione, H_1 per il primo cluster e H_2 per il secondo. Il problema principale di questo approccio è che esso genera 2^d cluster dove d è la dimensionalità degli oggetti. Nelle applicazioni di *information retrieval* non è raro trovare valori di $d > 100$, questo porterebbe ad una eccessiva frammentazione dell'insieme degli oggetti ottenendo cluster piccoli e poco rilevanti.

- *Hard* vs. *Fuzzy*: un algoritmo di clustering di tipo hard colloca ciascun oggetto in un singolo cluster, mentre un approccio di tipo fuzzy gli può assegnare un grado di partecipazione di un oggetto a più di un cluster. Questo approccio può essere convertito nel primo assegnando ogni oggetto al cluster con il più alto grado di partecipazione.
- *Deterministico* vs *stocastico*: questo aspetto è più rilevante nei metodi partizionali dove la classificazione viene ottimizzata tramite una funzione obiettivo. La caratteristica degli algoritmi deterministici è che, dato un input, restituiscono sempre lo stesso risultato, mentre quelli stocastici introducono nella generazione delle soluzioni scelte casuali che possono portare a risultati diversi per lo stesso input.

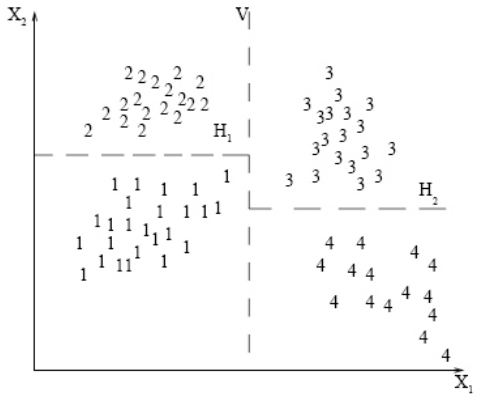

Fig 6.2: Clustering partizionale monoproprietà

In questo capitolo verranno approfondite solo due categorie di algoritmi di classificazione: approcci di classificazione gerarchico di tipo agglomerativo; approcci stocastici di tipo search-based. Gli algoritmi implementati sono basati infatti su queste due metodologie, le motivazioni di queste scelte sono rimandate in sez. 6.5.

6.1. Stato dell'arte

Il problema della classificazione è talmente importante e interdisciplinare che risulterebbe impossibile estrarre dalla letteratura un insieme di lavori di riferimento senza riempire intere pagine di metodologie. In questa sezione saranno riportati principalmente quei lavori che hanno come scopo revisioni generali e analisi comparative di metodi di classificazione applicati nelle diverse aree di ricerca.

Anche se nel campo del pattern recognition , dell'elaborazione di immagini e dell'information retrieval il problema della classificazione riscontra un interesse che mai scemerà, esiste una letteratura per ogni disciplina: la biologia, la psichiatria, l'archeologia, la geologia, la geografia e il marketing. Altri termini che possono essere considerati sinonimi di classificazione sono stati coniati come: unsupervised learning , numerical taxonomy , vector quantization e learning by observation . Anche il campo dell'analisi spaziale di punti è fortemente correlato al problema della classificazione.

È indispensabile citare anche le seguenti pubblicazioni di grande rilievo: libri sul problema della classificazione; visioni generali sulla classificazione pubblicati da Jain et al.; in è riportata l'analisi comparata di diversi algoritmi di classificazione basati sullo spanning-tree; la comparazione di vari schemi di ottimizzazione combinatoriale sono stati riportati in e in .

6.2. Clustering gerarchico

Molti algoritmi di clustering gerarchico sono varianti dell'approccio *single-link* introdotto in , *complete-link* (fig. 6.3) e *minimum-variance* .

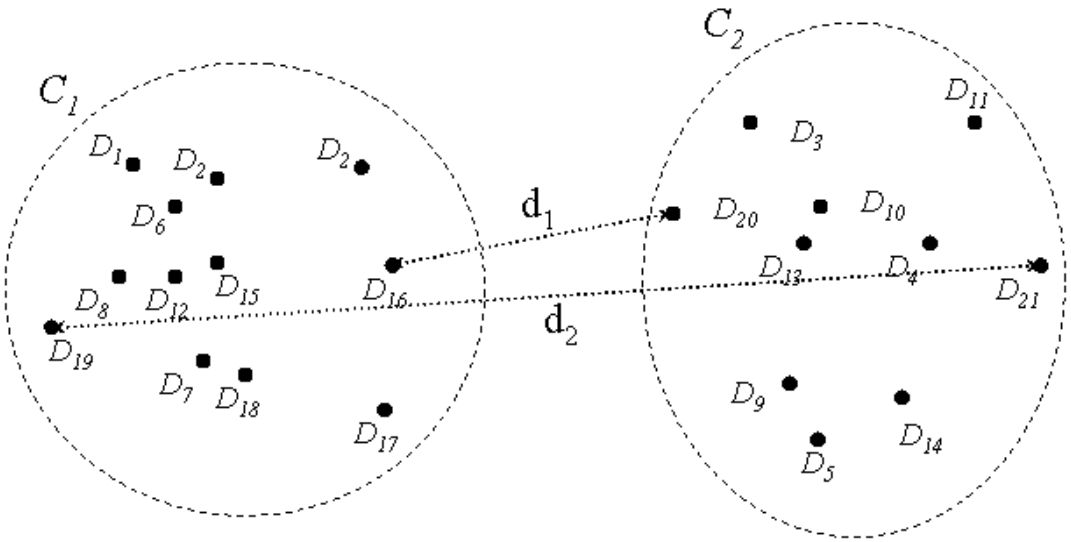

Fig 6.3: Calcolo distanza tramite single-link e complete-link

I primi due risultano essere quelli maggiormente utilizzati, essi differiscono nel modo in cui caratterizzano la similarità tra due cluster. Nel metodo *single-link* la distanza tra due cluster è uguale alla minima distanza di tutte le coppie di oggetti tali che un elemento è appartenente al primo cluster, l'altro al secondo. Diversamente, nel *complete-link*, la distanza è data dalla massima distanza dell'insieme di tutte le coppie sopra descritte. In entrambi i casi due cluster sono fusi insieme per formarne uno solo. La differenza principale tra i due approcci è che l'approccio complete-link tende a formare cluster maggiormente compatti e con una maggiore similarità intra-cluster, il single-link soffre del cosiddetto effetto a catena, ossia tende a formare cluster stretti e allungati.

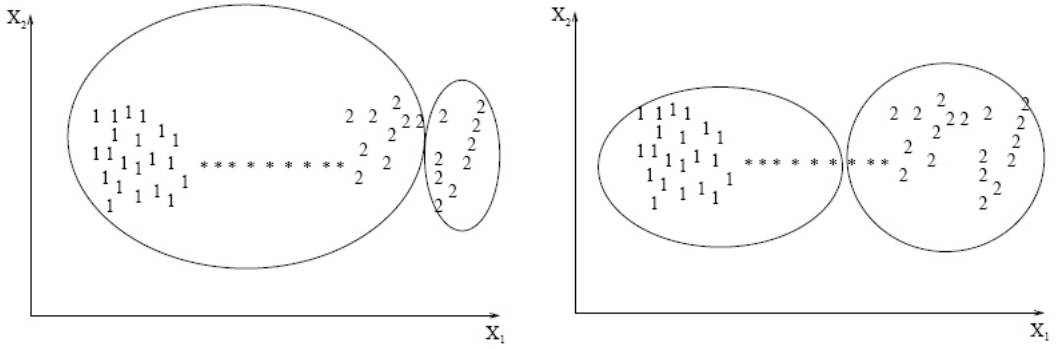

Fig 6.4: (a)Classificazione errata ottenuta col single-link, classificazione corretta ottenuta col complete-link.

In fig 6.4a e 6.4b sono mostrate le classificazioni ottenute tramite i due approcci sullo stesso insieme di dati. In queste figure sono rappresentati due cluster distinti con un "ponte" di oggetti rumorosi. Un algoritmo di clustering basato sul single-link produce la classificazione visualizzata in fig. 6.4a mentre uno basato sul complete-link produce la classificazione mostrata in fig. 6.4b. Nel primo caso una parte degli oggetti etichettati con "2" sono classificati insieme al primo cluster a causa dell'effetto a catena che si è verificato sugli oggetti rumorosi. Nel secondo caso l'algoritmo ha restituito cluster più compatti. In uno studio approfondito, descritto in , è stato osservato che l'approccio complete-link crea gerarchie maggiormente vicine alla realtà dell'approccio single-link. In fig. 6.5 è riportato lo pseudo-codice di un approccio gerarchico agglomerativo.

(1) Creare un cluster per ogni oggetto.
(2) Calcolare la matrice delle distanze tra i diversi cluster secondo la funzione obiettivo.
(3) Trovare i cluster più simili secondo la matrice delle distanze.
(4) Fondere i due cluster per ottenerne uno nuovo.
(5) Cancellare dalla lista dei cluster i due fusi al passo precedente e inserire il nuovo. Aggiornare la matrice delle distanze eliminando righe e colonne dei cluster cancellati ed inserendone una nuova per il cluster creato.
(6) Se il numero dei cluster è maggiore di uno tornare al passo (3).

Fig 6.5: Pseudo-codice di un approccio gerarchico agglomerativo

Il risultato dell'algoritmo di gerarchico è rappresentato dal dendrogramma identificato dalle fusioni effettuate al passo (4) nelle diverse iterazioni dell'algoritmo.

6.3. Clustering partizionale

Un algoritmo di clustering partizionale ottiene una singola classificazione dei dati invece di una sequenza di classificazioni come gli approcci gerarchici. I metodi di tipo partizionale presentano buone prestazioni in applicazioni con un numero elevato di oggetti da trattare. Questi metodi producono generalmente un insieme di cluster in modo da minimizzare una funzione obiettivo. La ricerca esaustiva di tutte le classificazioni possibili su un insieme di dati per la ricerca dell'ottimo è chiaramente proibitivo in termini di complessità computazionale. Tipicamente gli algoritmi di

tipo partizionale vengono eseguiti più volte partendo da diverse configurazioni e il miglior risultato ottenuto viene scelto come classificazione di output.

Tra i diversi tipi di approcci di tipo partizionale mostrati in fig. 6.1 esiste una classe di metodi chiamata *search-based*. Questa classe di algoritmi definisce il risultato tramite un'etichettatura degli oggetti da classificare in base a una funzione obiettivo. I metodi search-based possono essere ulteriormente suddivisi in deterministici e stocastici. Le tecniche deterministiche convergono al risultato perlustrando tutto lo spazio delle soluzioni (ricerca esaustiva) o parte di essi. Le tecniche di tipo stocastico ricercano in maniera pseudo-casuale la soluzione al problema e convergono asintoticamente all'ottimo. Il principio degli algoritmi stocastici che li rende in generale più potenti di quelli greedy è che i primi permettono di spostarsi verso direzioni non localmente ottime nella ricerca della soluzione. Il simulated-annealing è un approccio partizionale di tipo search-based stocastico. Questo algoritmo tende asintoticamente alla soluzione ottima e non soffre della rigidità presentata dai metodi di tipo gerarchici. Infatti utilizza un concetto di temperatura che, iterazione dopo iterazione, diminuisce. La temperatura rappresenta la probabilità di accettare soluzioni che peggiorano la classificazione, questo permette inizialmente di sorpassare i minimi locali e di convergere poi verso una soluzione accettabile.

6.4. Il Simulated Annealing

Il Simulated Annealing (SA) è stata negli anni una euristica ampiamente utilizzata e discussa. Autori come Johnson et al. forniscono eccellenti descrizioni del SA con analogie e paragoni rispetto alla fisica del fenomeno di formazione delle strutture cristalline. Aarts e Korst ne inquadrano la storia, dalle origini dell'idea principale negli anni '50 fino alla sua applicazione nei problemi di ottimizzazione negli anni '80. Questi ultimi presentano un modello formale del SA e provano che l'algoritmo converge alla soluzione ottima in maniera asintotica.

6.4.1. L'algoritmo simulated annealing

È una metodologia di ricerca altamente adatta per qualunque problema di ottimizzazione non convessa, e fonda le sue basi nella statistica meccanica. È stata sviluppata originariamente da Kirkpatrick et al. per risolvere problemi di ottimizzazione combinatoriale e discreta. Il SA è nato come metodo di simulazione della tempra (annealing) dei solidi. L'annealing è il processo con il quale un solido,

portato allo stato fluido, mediante riscaldamento ad alte temperature, viene riportato poi di nuovo allo stato solido o cristallino, a temperature basse, controllando e riducendo gradualmente e lentamente la temperatura. La probabilità $P(E_i)$ in uno stato avente energia E_i è governata dalla distribuzione di Boltzmann :

$$P(E_i) = \frac{e^{-\frac{E_i}{k_B T}}}{\sum_j e^{-\frac{E_j}{k_B T}}}$$

dove k_B è la costante di Boltzmann.Si noti che ad alte temperature tutti gli stati di energia sono probabilmente possibili, mentre a basse temperature il sistema si trova sicuramente in stati di minima energia.

Metropolis et al. nel 1953 sviluppò un algoritmo per simulare il comportamento di una collezione di atomi in equilibrio termico a una particolare temperatura. Questo algoritmo ha un ruolo fondamentale per l'applicazione del metodo SA a problemi di ottimizzazione. Senza perdita di generalità, si consideri un problema di minimizzazione. La caratteristica essenziale dell'algoritmo di Metropolis è che genera un insieme di configurazioni a ogni temperatura T con la proprietà che le energie delle differenti configurazioni possono essere rappresentate dalla distribuzione di Boltzmann. Il metodo comincia da una assegnata configurazione iniziale degli atomi in un sistema con energia E_0. Vengono quindi generate successive configurazioni con piccole perturbazioni casuali della configurazione corrente. Viene deciso se accettare o rigettare la configurazione in base alla differenza fra l'energia della configurazione corrente e quella della nuova configurazione (o configurazione candidata). Tale decisione è influenzata dal fatto che le energie del sistema delle configurazioni accettate devono formare una distribuzione di Boltzmann se si è raggiunto l'equilibrio termico. L'algoritmo di Metropolis accetta sempre una soluzione candidata la cui energia E_j è inferiore a quella della configurazione corrente E_i. Per contro se l'energia E_j della configurazione candidata è più grande di quella della configurazione corrente, allora la soluzione è accettata con la seguente probabilità :

$$P(\Delta E) = e^{-\frac{\Delta E}{k_B T}}$$

dove $\Delta E = E_j - E_i$.

Ad alte temperature, l'algoritmo SA può attraversare quasi tutto lo spazio degli stati poiché pessime soluzioni vengono facilmente accettate. Successivamente, abbassandosi il valore di temperatura, l'algoritmo viene confinato in regioni sempre più ristrette dello spazio di stato dato che la distribuzione di Boltzmann collassa a sempre più piccole o basse probabilità di accettazione. Per i problemi di ottimizzazione il SA lavora come segue: ad alte temperature l'algoritmo si comporta più o meno come una random search. La ricerca salta da un punto all'altro dello spazio delle soluzioni individuandone le caratteristiche e quindi le direzioni o le aree in cui è più probabile individuare l'ottimo globale. A basse temperature l'SA è simile ai metodi steepest descent. Le soluzioni vengono localizzate nella zona del dominio maggiormente promettente. L'algoritmo SA per un problema di ottimizzazione può essere riassunto nel seguente generico algoritmo:

1. Simulated annealing (x^0 , T_0)
2. E_0 =CalcolaEnergia(x^0)
3. T= T_0
4. Mentre T> T_f
5. mentre (!equilibrioTermico)
6. x^1 =perturba(x^0)
7. E_1 =CalcolaEnergia(x^1)
8. $\Delta E = E_1 - E_0$
9. se $\Delta E \leq 0$ allora
10. accettaSoluzione = true
11. altrimenti
12. accettaSoluzione = CalcolaProb(T)
13. se accettaSoluzione = true allora
14. $x^0 = x^1$
15. $E_0 = E_1$
16. Fine mentre
17. T= α T
18. Fine mentre

Fig 6.6: Pseudo-codice del simulated annealing

6.4.2. Schema di raffreddamento

Dato che la riduzione della temperatura governa il successo dell'algoritmo nella sua ricerca delle soluzioni, è importante selezionare un appropriato schema di riduzione degli stadi di temperatura determinando quando e quanto ridurla:

- *Quando*: accade se ad una data temperatura le energie o i valori della funzione obiettivo delle soluzioni accettate approssimano la distribuzione di Boltzmann per quello stadio di temperatura. Chiaramente più elevato è il numero di soluzioni accettate, più si approssima la distribuzione di Boltzmann. Tuttavia, generare una elevata quantità di soluzioni per ogni stadio di temperatura porta ad un aumento del tempo di calcolo. A tal fine è necessario tarare un parametro detto transizione. Il numero di transizioni ad ogni stadio di temperatura deve

essere tale da assicurare che si passi da uno stadio termico al successivo solo quando si è raggiunto l'equilibrio ossia le configurazioni siano distribuite secondo la probabilità di Boltzmann.

- *Quanto*: ridurre la temperatura troppo velocemente causa la possibilità da parte dell'algoritmo di rimanere confinato in un minimo locale. D'altro canto, ridurre la temperatura molto lentamente comporterà un aumento degli stadi di temperatura con un conseguente innalzamento del tempo di calcolo. Basandosi su queste considerazioni gli analisti determinano un appropriato schema di raffreddamento in base al problema specifico. Vi sono, essenzialmente, due schemi di raffreddamento della temperatura: quello geometrico e quello logaritmico . Il primo è molto semplice, e viene utilizzato dalla maggioranza dei ricercatori: la temperatura viene ridotta, ad ogni stadio, moltiplicandola per una costante chiamata *cooling ratio*, che ha un valore compreso nell'intervallo]0,1[.

6.4.3. Criterio di interruzione

La determinazione di un appropriato criterio di interruzione è un fattore critico per il successo del SA. Il processo di annealing converge ad una soluzione accettabile se la temperatura è sufficientemente ridotta in modo graduale. Al fine di evitare un eccessivo tempo di calcolo vengono usualmente interrotte le iterazioni quando il numero di soluzioni accettate ad un certo stadio è inferiore ad un valore fissato. In tal caso ci si può aspettare una soluzione accettabile. Eventualmente, per affinare o verificare la bontà della soluzione ottenuta, è possibile utilizzare dei metodi di ricerca locale.

Tipicamente, quando non si voglia adottare un criterio di interruzione, è possibile utilizzare un valore finale della temperatura. Tale valore corrisponde dal punto di vista fisico dell'annealing al valore della temperatura ambiente quando il bagno di fusione si è solidificato in una struttura cristallina stabile e di minima energia. Tecnicamente tale valore finale della temperatura viene stabilito in base al problema che si sta affrontando.

6.4.4. Generazione delle soluzioni e soluzione finale

Anche la generazione delle soluzioni rappresenta un punto cardine per la velocità di calcolo dell'algoritmo SA. È necessario che la procedura di generazione, detta anche perturbazione, sia il più veloce possibile. Il numero elevato di transizioni per ogni stadio di temperatura, se associato ad un elevato tempo di generazione di

soluzioni candidato, porterebbe sicuramente ad un aumento esponenziale del tempo di calcolo. Ecco perché solitamente con questo metodo vengono utilizzate tecniche di perturbazione locale. Infatti tali tecniche consentono la generazione di soluzioni in tempi rapidissimi, anche se il loro valore non è molto distante da quello della soluzione corrente. Ciò aggiunge un altro non irrilevante vantaggio: la possibilità che la configurazione candidato sia anch'essa una soluzione ammissibile. Anche se per il SA è stata dimostrata la convergenza, si ricorda che questa si ha al prezzo di un elevato tempo di calcolo; di conseguenza ridurre tale tempo implica una più elevata velocità di riduzione della temperatura e un numero inferiore di transizioni, con la probabilità di interrompere il processo di annealing prima che sia arrivato a convergenza. In tal caso, dato il fluttuare del valore della soluzione, è una buona prassi, per ottenere una maggiore bontà della soluzione finale, conservare la migliore configurazione individuata man mano che l'algoritmo lavora. Ciò assicura che tale soluzione migliore possa essere recuperata, anche se il processo termina con una soluzione di valore peggiore.

6.5. Algoritmi di classificazione implementati

Nell'introduzione di questo capitolo sono state riportate le diverse metodologie di classificazione e, successivamente, si è analizzata con un maggior livello di dettaglio la tecnica di classificazione gerarchica di tipo agglomerativo e il simulated annealing. La scelta delle tecniche di clustering è stata suggerita dal dominio applicativo e dalla complessità del problema. Inizialmente si è scelto di implementare un algoritmo semplice di classificazione e ben comprensibile ad utenti esperti del dominio biologico: gli approcci di tipo gerarchico soddisfano i requisiti sopra citati e sono estremamente utilizzati nel campo biologico, inoltre questi algoritmi necessitano la definizione di una funzione di similarità tra gli oggetti da classificare, e tramite questa è possibile restituire una matrice in output con i valori della funzione per ogni coppia di oggetti. Il dendrogramma e la matrice di similarità sono dei buoni strumenti per l'analisi della classificazione ottenuta. Si è poi scelto di implementare un secondo algoritmo che ponesse al centro della funzione obiettivo il pattern come oggetto correlante di più proteine. Dato l'elevato numero di pattern di superficie si è scelto di implementare un algoritmo di classificazione partizionante e stocastico per ovviare al problema della complessità. Nel simulated annealing la complessità viene regolata tramite la costante di decremento della temperatura e tramite il numero di iterazioni per ogni ciclo. Questi due parametri permettono

all'utente di scegliere se privilegiare la bontà del risultato o la velocità di esecuzione.

In tabella 6.1 viene introdotta la simbologia adottata nella formalizzazione degli algoritmi.

Tab. 6.1*:* Leggenda dei simboli utilizzati negli algoritmi.

Oggetti	**Descrizione**
D_j	Proteina j-esima
$D = \{D_1,...,D_n\}$	Database di proteine
P_i	Pattern i-esimo
$P = \{P_1,...,P_m\}$	Insieme dei pattern
$\|P_i\|$	Dimensione del pattern (numero di regioni che lo costituiscono)
$Supp(P_i)$	Supporto del pattern (contiene l'elenco delle proteine che supportano il pattern i-esimo considerato)
$\|Supp(P_i)\|$	Dimensione del supporto del pattern i-esimo (numero di proteine che costituiscono il supporto)
C_i	Cluster i-esimo contenente proteine
$\|C_i\|$	Numero di proteine che costituiscono il cluster
Cl_i	Classificazione i-esima di proteine divise in cluster
$Cl = \{Cl_1,...,Cl_k\}$	Insieme delle classificazioni ottenute dall'algoritmo gerarchico
$\|Cl_i\|$	Cardinalità classificazione i-esima (numero di cluster che costituiscono la classificazione)
$M(Cl_i)$	Matrice di Similarità della classificazione Cl
$M_{x,y}(Cl_i)$	Coordinate (x,y) del valore massimo di similarità inter-cluster relative alla Matrice di Similarità della Classificazione i-esima
$Sim_{x,y}$	Valore di Similarità tra il cluster i-esimo e j-esimo
H_i	Iperarco i-esimo
$\|H_i\|$	Dimensione dell'iperarco (numero di proteine toccate)

6.6. Algoritmo gerarchico

Come introdotto in sez. 6.2 l'algoritmo gerarchico ha come punto di partenza la classificazione in cui ogni singolo oggetto risiede in un cluster (*singleton*), successivamente coppie cluster vengono fuse insieme in maniera iterativa fino a quando il criterio di stop viene soddisfatto. In fig. 6.7 è riportato l'algoritmo di clustering implementato.

```
1.  HierarchicalClustering( D ,  P )
2.  {
3.     Cl_1 =CreateSingletons( D );
4.     AddClustering( Cl , Cl_1 );
5.
6.     i=1;
7.     while(| Cl_i |> 1){
8.        M_i =GetSimilarityMatrix( Cl_i );
9.        (x,y)=GetBestValue( M_i );
10.       Cl_{i+1} =MergeClusters( Cl_i , x, y);
11.       AddClustering( Cl , Cl_{i+1} );
12.       i++;
13.    }
14.    return  Cl ;
15. }
```

Fig 6.7: Ciclo base del algoritmo gerarchico.

In riga 3. la funzione `CreateSingletons(` D `)` crea la classificazione Cl_1 di partenza dove ogni proteina risiede in un cluster diverso, questa viene poi inserita (riga 4.) nell'insieme Cl . Il blocco di istruzioni (righe 8.-12.) viene eseguito fino a quando non si raggiunge la condizione di stop $|Cl_i|=1$, ossia quando la classificazione restituita dall'ultimo ciclo è composta da un unico cluster contenente tutte le proteine.

Ad ogni iterazione del ciclo `while` (riga 7.) viene inizialmente calcolata la matrice di similarità M_i (riga 8.) tramite la funzione `GetSimilarityMatrix(` Cl_i `)` che valuta la distanza dei cluster di Cl_i attraverso una funzione obiettivo che sarà discussa in sez. 6.6.1. Il massimo valore di questa matrice (riga 9.) indica i due cluster più simili che dovranno essere fusi per creare il nuovo tramite la funzione funzione `MergeClusters(` Cl_i `, x, y)` (riga 10.). La classificazione restituita dalla funzione Cl_{i+1} è ottenuta da quella passata come parametro, Cl_i, dove viene effettuata una la fusione dei due cluster indicati dai restanti parametri `(x, y)`. Per

ultimo la nuova classificazione viene inserita nell'insieme *Cl* (riga 11.) per poi passare alla valutazione della condizione di stop (riga 7.).

Il dendrogramma risultante è individuato dall'insieme di classificazioni *Cl* ottenute dalla sequenza di fusioni, *Cl* viene poi restituito dall'algoritmo (riga 14.) prima della sua terminazione.

6.6.1. funzione obiettivo

$DB = \{D_1, \ldots, D_n\}$	Database di proteine
$D_s = \{r_s^1, \ldots, r_s^a\}$	Insieme delle regioni costituenti la proteina D_s
$DP_i = \{P_i^1, \ldots, P_i^b\}$	Ogni oggetto P_i^l rappresenta l'insieme dei pattern della proteina D_i di livello l, ovvero costituiti da l regioni
$P_i^l = \{p_{i,1}^l, \ldots, p_{i,c}^l\}$	Pattern di livello l nella proteina D_i
$SP_{i,k}^l = \{sp_{i,k}^{l,1}, \ldots, sp_{i,k}^{l,d}\}$	Insieme di pattern che supportano $p_{i,k}^l$
$prot(sp_{i,k}^{l,q}) = D_x$	Funzione che restituisce la proteina del pattern $sp_{i,k}^{l,q}$

Fig 6.8: Legenda dei simboli utilizzati nella funzione obiettivo del gerarchico

$$sim(D_i, D_j) = \frac{f_{sim}(D_i, D_j) + f_{sim}(D_j, D_i)}{2}$$

$$f_{sim}(D_i, D_j) = \frac{\sum_{l=1}^{|D_i|}(f_{sigm}(l, \frac{|D_i|}{3}, 0.4)\frac{\sum_{k=1}^{|D_i|} isOverlaid(r_i^k, D_j, l)}{|D_i|})}{|D_i|}$$

$$isOverlaid(r_i^k, D_j, l) = \begin{cases} 1 & \text{se } \exists t \in [1, \ldots, |P_i^l|] \text{ t.c. } regionInPattern(r_i^k, p_{i,t}^l) \wedge isShared(p_{i,t}^l, D_j) \\ 0 & \text{altrimenti} \end{cases}$$

$$regionInPattern(r_i^k, p_{i,t}^l) = \begin{cases} vero & \text{se la regione } r_i^k \text{ partecipa alla formazione di } p_{i,t}^l \\ falso & \text{altrimenti} \end{cases}$$

$$isShared(p_{i,t}^l, D_j) = \begin{cases} vero & \text{se } \exists u \in [1, \ldots, |SP_{i,t}^l|] \text{ t.c. } prot(sp_{i,t}^{l,u}) = D_j \\ falso & \text{altrimenti} \end{cases}$$

Fig 6.9: Funzione obiettivo del gerarchico

6.7. Algoritmo basato sul simulated annealing

I dettagli dell'algoritmo di classificazione basati sul simulated annealing sono riportati in fig. 6.10. L'algoritmo necessita di un insieme di parametri maggiore rispetto al più semplice gerarchico, oltre a quelli riportati in tab. 6.1 abbiamo: la classificazione di partenza Cl_0; la temperatura iniziale e finale T_0 e T_f; il

coefficiente di decremento della temperatura α; il numero di iterazioni n per il raggiungimento dell'equilibrio termico.

```
1.  SimulatedAnnealing( D , P , Cl0 , T0 , Tf , α , n )
2.  {
3.    T = T0 ;
4.    Cl_Best = Cl0 ;
5.    while ( T > Tf )
6.    {
7.       i=0;
8.       while (i < n )
9.       {
10.         i++;
11.         Cl_New =Perturb( Cl0 );
12.         if (GetScore( Cl_New )>GetScore( Cl_Best ))
13.            Cl_Best = Cl_New ;
14.         ΔE =GetScore( Cl_New )-GetScore( Cl0 );
15.         if (( ΔE > 0 )or(Rnd(0,1)< e^(-ΔE/T) ))
16.            Cl0 = Cl_New ;
17.      }
18.   T = αT ;
19.   }
20.   return Cl_Best ;
21. }
```

Fig 6.10: Algoritmo simulated annealing

Nella computazione, l'algoritmo necessita di alcune variabili di appoggio, tra le quali abbiamo: la temperatura corrente T; la miglior classificazione identificata Cl_{Best}; la classificazione candidata ad ogni iterazione Cl_{New}.

L'algoritmo segue fedelmente quello proposto in . Inizialmente la temperatura del simulated annealing è alta (riga 3.), per ogni valore di temperatura viene effettuato un ciclo (righe 9.-17.) che effettua una ricerca delle soluzioni possibili alla temperatura T. Le soluzioni sono generate tramite la funzione `Perturb(`Cl_0`)` (riga 11.) che effettua una perturbazione pseudo-casuale della classificazione passata come parametro. Questa funzione risulta essere molto importante per l'efficacia dell'algoritmo e per la sua velocità di esecuzione, maggiori dettagli su questa funzione saranno dati in sez. 6.7.4. Successivamente viene valutata la differenza ΔE di bontà delle due soluzioni (riga 14.) tramite la funzione `GetScore(`Cl_x`)` che calcola lo score della classificazione immessa come parametro secondo la funzione

obiettivo riportata in sez. 6.7.3. La soluzione ottenuta dalla perturbazione viene accettata (riga 15.) se è migliore della precedente ($\Delta E > 0$) oppure, se peggiore, può venire accettata con una probabilità pari a $e^{-\frac{\Delta E}{T}}$. Le condizioni di accettazione saranno maggiormente discusse in sez. 6.7.5.

Raggiunto l'equilibrio termico, ovvero effettuati un numero di cicli pari al valore n, viene abbassata la temperatura di un coefficiente α. Se la temperatura raggiunta è maggiore di quella finale T_f viene ripetuto il ciclo (righe 9.-17.).

Notare che, durante l'intera esecuzione dell'algoritmo, viene tenuta traccia della migliore classificazione in Cl_{Best}. Questa è la soluzione restituita dall'algoritmo prima della sua terminazione.

6.7.1. Scelta dei parametri dell'algoritmo

Come già descritto in 6.3 il simulated annealing è una tecnica di tipo stocastico che tende asintoticamente alla soluzione ottima del problema. I parametri dell'algoritmo proposto permettono di favorire il tempo di calcolo alla bontà della soluzione o viceversa. Naturalmente è indispensabile trovare un giusto compromesso tra efficacia ed efficienza. In è riportata una discussione per una scelta adatta dei parametri:

- quale classificazione iniziale Cl_0 scegliere,
- come deve essere determinato il valore iniziale di temperatura T_0,
- qual è il criterio di equilibrio, rappresentato nell'algoritmo dal numero n di iterazioni,
- che valore deve avere il fattore α di decremento della temperatura,
- quando il sistema è considerato freddo ossia qual è il valore di temperatura finale T_f.

Una buona scelta di questi cinque parametri può portare al raggiungimento di risultati accettabili in un minor tempo di calcolo. La scelta dei parametri può essere guidata dai seguenti criteri:

- scelta di Cl_0: anche se la scelta del punto iniziale può essere casuale è consigliabile partire da una buona classificazione. Nel progetto di ricerca si è scelto come punto iniziale la miglior classificazione ottenuta dall'algoritmo gerarchico, questo non vincola comunque il simulated annealing alla località di

questa soluzione in quanto inizialmente è libero di accettare soluzioni distanti e peggiori visto che l'alta temperatura T_0 lo permette.

- Scelta di T_0: intuitivamente la temperatura iniziale deve essere sufficientemente alta da poter uscire dalla zona di minimo locale individuata dalla soluzione iniziale. Ciò significa che inizialmente anche soluzioni estremamente cattive devono avere una buona probabilità di essere accettate, ossia $e^{-\frac{\Delta E}{T}} \approx 1$ anche per ΔE fortemente negativi. Allo scopo di determinare il valore T_0 adatto è stata utilizzata una procedura `GetT0(`Cl_0`)` così definita: inizialmente il valore di T_0 è posto uguale a 0, successivamente viene effettuata una sequenza di perturbazioni alla classificazione Cl_0 passata come parametro. Dopo ogni modifica di Cl_0 viene ricalcolato il valore di T_0 secondo la seguente equazione:

$$T_0 = \frac{\overline{\Delta E}^{(+)}}{\ln\left(\frac{m_2}{m_2 x_0 - m_1(1 - x_0)}\right)}$$

dove x_0 rappresenta il rapporto di accettazione iniziale desiderato, ossia il rapporto tra il numero di mosse accettate e il numero di mosse totali iniziali. In questo progetto il valore di x_0 è posto a 0.95. m_1 e m_2 rappresentano il numero di mosse positive e negative ottenute durante l'esecuzione di `GetT0(`Cl_0`)`. $\overline{\Delta E}^{(+)}$ è il costo medio sulle m_2 mosse negative. Questo calcolo di T_0 è stato introdotto in ed è stato provato che il calcolo converge presto ad un valore stabile.

- Scelta di α: tipici valori di α, in letteratura, fluttuano nell'intervallo [0.8,…,0.99]. Nel nostro progetto α è stato scelto pari a 0.9.
- Scelta di n: il numero n di iterazioni per il raggiungimento dell'equilibrio termico calcolato tramite la formula $n = \beta |N|$. β rappresenta un coefficiente che varia nell'intervallo [5,…,8] e N è l'insieme degli oggetti da classificare.
- Scelta di T_f: il sistema è considerato freddo quando viene raggiunta la temperatura finale che, nel nostro progetto, è posta uguale a 0.001. Per chiarezza nell'algoritmo non è stato riportato un ulteriore criterio che porta a considerare freddo il sistema. Questo criterio dice che se nelle ultime due esecuzioni del ciclo più interno (righe 9.-17.) non si è verificata una variazione di energia, ossia

nessuna perturbazione è stata accettata, allora il sistema è considerato freddo e l'algoritmo termina.

6.7.2. Il concetto di iperarco

Prima di illustrare la funzione obiettivo e la metodologia di perturbazione adottata è indispensabile introdurre il concetto di iperarco. Un iperarco è un arco che unisce, o tocca, due o più nodi. In questo dominio i nodi sono rappresentati dalle proteine del database mentre gli iperarchi rappresentano i pattern complessi di superficie condivisi da diverse proteine. In fig. 6.11 è schematizzato un iperarco che connette le proteine D_2, D_4, D_{10} e D_{15}.

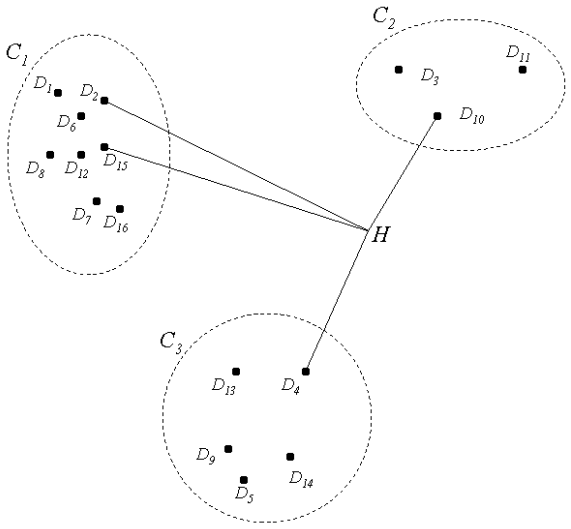

Fig 6.11: Esempio di iperarco

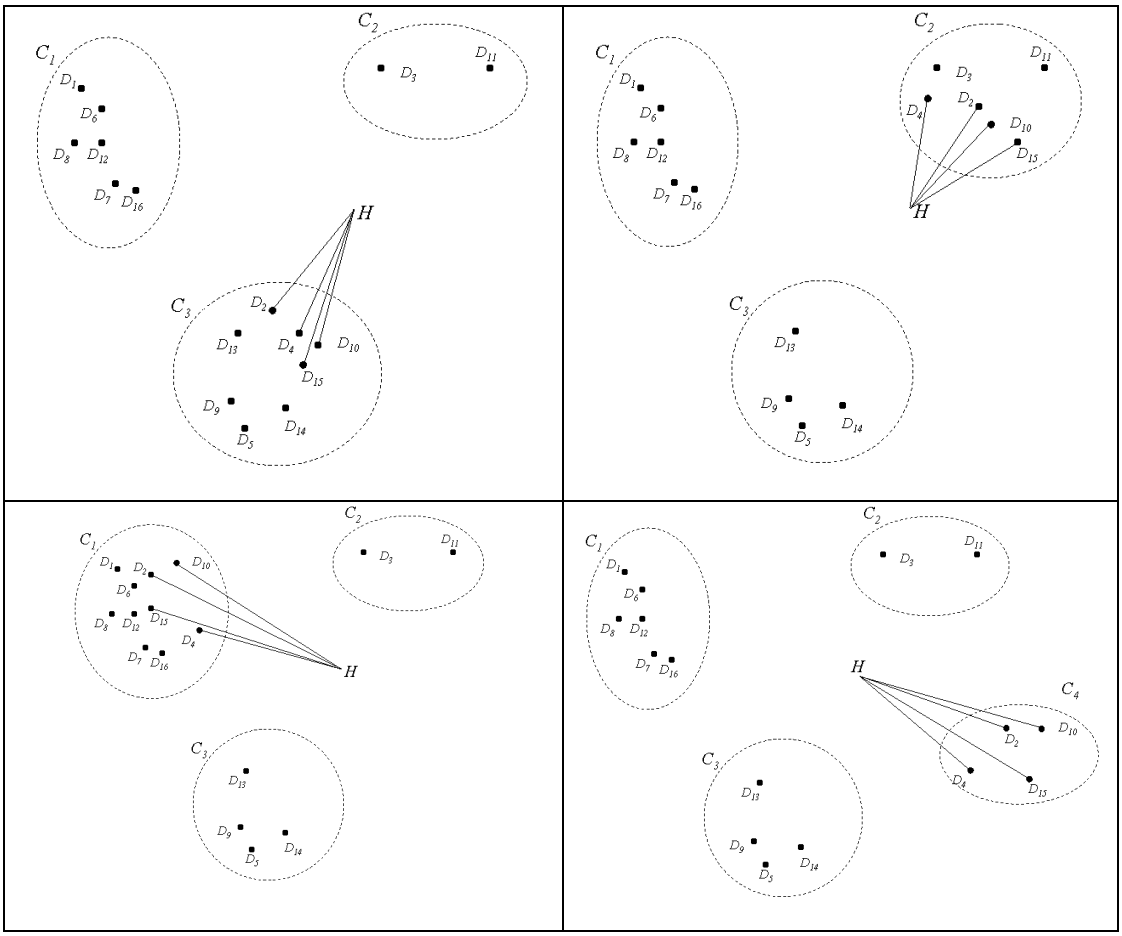

Fig 6.12: perturbazioni possibili

Dal momento che un iperarco schematizza un pattern e il suo supporto risulta chiaro che la cardinalità dell'insieme degli iperarchi è pari a quella dell'insieme dei pattern e ogni iperarco è etichettato con l'estensione dell'area totale e il numero di regioni del pattern a lui associato.

6.7.3. Funzione obiettivo

La funzione di score del gerarchico è definita dalla seguente formula:

$$f_{score} = \sum_{1}^{N^\circ HyperEdges} f_c \times f_s \times f_h$$

Formula 3.4

Si analizzano ora le tre sottofunzioni che compongono la funzione di score: f_c f_s e f_h.

$$f_c(P_i) = \frac{|P_i|}{|P_{\max}|}$$

Formula 3.5

f_c pesa l'importanza dell'iperarco all'interno del database, essa controlla quanto risultano essere grandi i pattern toccati dall'iperarco (da quante regioni sono composti i pattern, si osservi che per ogni iperarco questo numero è costante) rispetto al pattern di dimensione massima del database.

$$f_s(H_i, \mathrm{n}) = \frac{\sum_{x=1}^{\mathrm{n}} \frac{(|H_i| \in C_X)^2}{|C_x|}}{|H_i|}$$

Formula 3.6

f_s descrive quanto le proteine dell'iperarco considerato rappresentano i cluster toccati (*n*): se l'iperarco è tutto contenuto all'interno di un unico cluster e se all'interno del cluster ci sono solo ipernodi toccati dall'iperarco questo indice offre i migliori risultati; viceversa, più l'iperarco tocca cluster diversi, e più ognuno di essi contiene ipernodi diversi da quello dell'iperarco considerato, più l'indice peggiora.

$$f_h(H_i, n) = \frac{|H_i| - n + 1}{|H_i|}$$

Formula 3.7

f_h descrive quanto l'iperarco è spezzato, ovvero l'indice migliora se l'iperarco è contenuto in unico cluster, e peggiora più esso tocca cluster distinti.

Tutti e tre i fattori della *formula 3.4* prima di essere moltiplicati tra loro vengono normalizzati utilizzando regola:

$$f = \frac{f_x \quad f_{\min}}{f_{\max} \quad f_{\min}}$$

Formula 3.8.

in questa maniera si ottengono valori compresi nell'intervallo [0,1].

6.7.4. Perturbazione di una classificazione

In fig. 6.10 è stato riportato l'algoritmo di classificazione basato sul tecnica del simulated annealing. La natura stocastica di questa tecnica richiede di realizzare una metodologia di perturbazione pseudo casuale per spostarsi nello spazio delle soluzioni. La scelta di questa tecnica risulta critica per quanto riguarda l'efficace e l'efficienza dell'algoritmo:

- efficienza: una perturbazione costosa in termini di tempo porterebbe ad un rallentamento eccessivo dell'esecuzione, infatti, la funzione di perturbazione risiede nel ciclo più interno dell'algoritmo (fig. 6.10, riga 11.). Inoltre il simulated annealing necessita di un lento decremento della temperatura e un elevato numero di iterazioni per ogni suo valore allo scopo di raggiungere soluzioni accettabili. Tutto questo si riflette in un enorme numero di perturbazioni richieste;
- efficacia: l'algoritmo di generazione delle perturbazioni deve poter generare soluzioni all'interno di tutto lo spazio di ricerca. Un algoritmo errato di perturbazione potrebbe restringere lo spazio di ricerca perdendo delle soluzioni e diminuendo l'efficacia.

Queste considerazioni suggerirebbero di effettuare una scelta puramente casuale della perturbazione per garantire la massima ampiezza di ricerca. È comunque consigliabile pilotare leggermente la scelta verso buone soluzioni secondo la funzione obiettivo, è stato dimostrato che in questa maniera l'algoritmo guadagna in efficacia. Infatti, a parità di perturbazioni effettuate, il numero di soluzioni non accettabili generate diminuisce favorendo la perlustrazione di zone "buone".

Come descritto in sez. 6.7.3 la funzione obiettivo è composta da due sottofunzioni fondamentali. Data una classificazione Cl e un insieme di iperarchi H, una prima funzione f_{inter} valuta quanto gli iperarchi sono spezzati (se toccano più di un cluster), una seconda f_{intra} valuta invece quanto gli iperarchi sono rappresentativi dei cluster toccati. Le due sottofunzioni suggeriscono di perturbare la classificazione cercando di migliorare questi due aspetti. La tecnica di perturbazione adottata sceglie in maniera casuale un iperarco H_i dall'insieme H e ottimizza una di questa due sottofunzioni. In altre parole scelto H_i pone tutte le sue proteine in un cluster toccato in modo da ottimizzare f_{inter}; oppure prende tutte le proteine toccate e le pone in un nuovo cluster in modo da ottimizzare f_{intra}. In fig. 6.12 vengono

mostrate diverse possibili perturbazioni dati un database, una classificazione e un iperarco scelto in maniera casuale.

Così definita la funzione di perturbazione rispetta entrambe le proprietà descritte all'inizio di questo sottoparagrafo: effettua una perturbazione veloce; facendosi guidare dalla funzione obiettivo, favorisce la scelta di soluzioni che spingono l'algoritmo verso l'ottimo.

6.7.5. Probabilità di accettazione

Una caratteristica degli algoritmi basati sulla tecnica del simulated annealing è quella di accettare, con una certa probabilità, anche soluzioni peggiori secondo la funzione obiettivo. Questa probabilità, nel simulated annealing, varia in funzione della temperatura del sistema e del salto energetico da compiere. È chiaro che la probabilità di accettazione diminuisce al diminuire della temperatura o all'aumentare del salto energetico. La funzione di probabilità (fig. 6.10, riga 15.) è definita dalla formula:

$$e^{-\frac{\Delta E}{T}}$$

dove ΔE rappresenta il salto di energia calcolato come differenza della bontà delle due classificazioni, T rappresenta la temperatura. In fig. 6.13 e 6.14 vengono riportate l'andamento della probabilità al crescere di di ΔC per temperature T=70 e T=20.

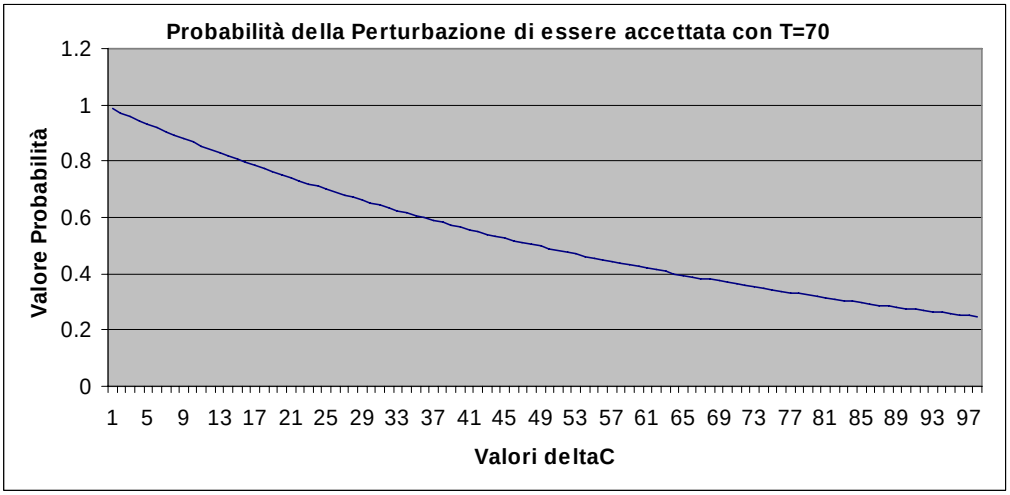

Fig 6.13: probabilità alla temperatura T=70

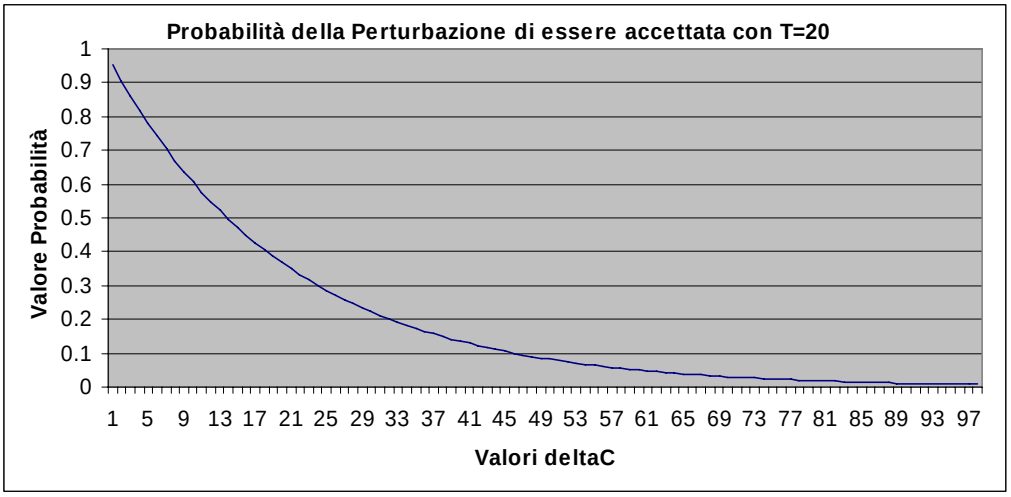

Fig 6.14: probabilità alla temperatura T=20

7. Risultati sperimentali

7.1. Dataset utilizzati

7.1.1. Proteine conformi

Il primo dataset utilizzato per testare l'intero approccio alla classificazione è formato da due insiemi di proteine. Ogni insieme è stato creato prendendo una proteina seme e generando tanti modelli a lei conformi.

La generazione di modelli conformi si basa sul concetto che la proteina, in natura, non ha una conformazione rigida ma la sua struttura risulta essere leggermente elastica, questo offre la possibilità alla proteina di modificare leggermente la propria struttura, e conseguentemente la propria superficie, rimanendo in un basso stato di energia. In più le catene laterali degli aminoacidi presentano una libertà di movimento che va ad aumentare le possibili conformazioni della proteina nel suo ambiente.

Il file PDB descrive strutturalmente la proteina in uno dei sui tanti stati possibili, un po' come fornirne una fotografia; l'analisi dei modelli conformi restituisce un insieme di file PDB dove ognuno di essi descrive la proteina in uno stato energicamente stabile.

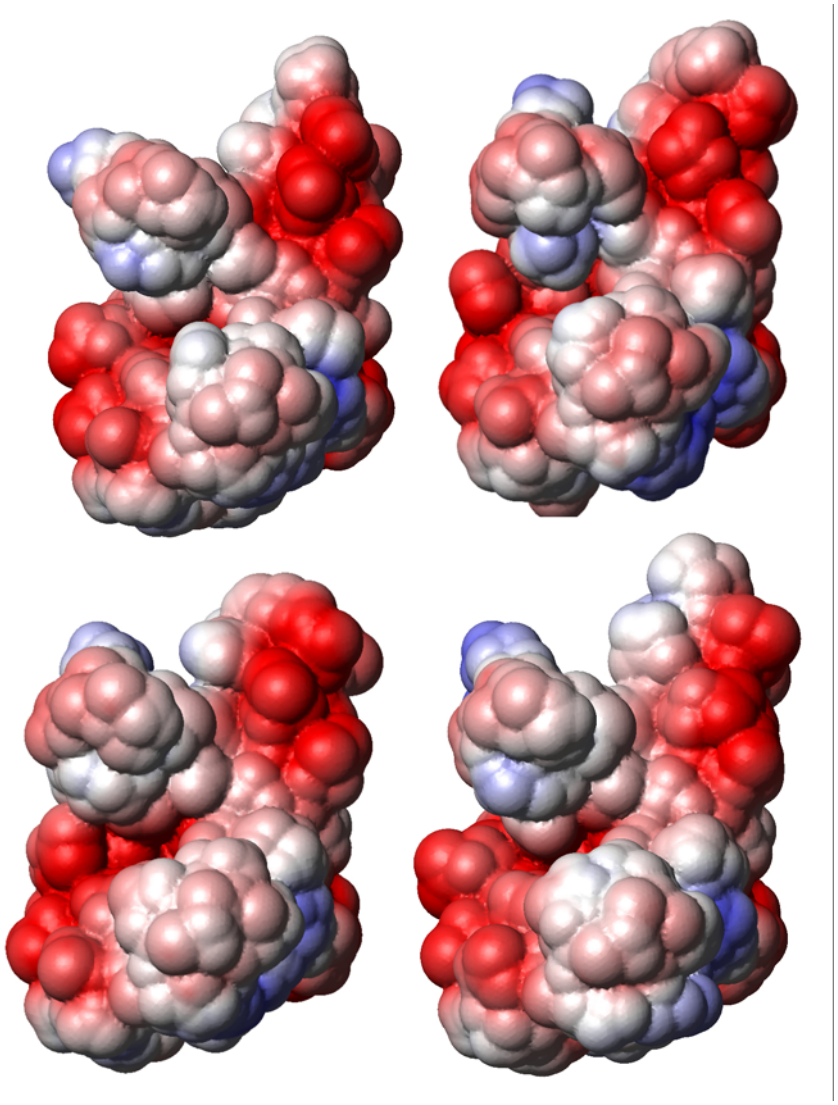

Fig 7.1: esempi di proteine conformi.

La proteina di riferimento, chiamata anche seme, e l'insieme dei suoi conformi, rappresentano un insieme del dataset utilizzato. Tali insiemi sono generati dalle proteine descritte in tab. 7.1 e da 20 proteine conformi al proprio seme. L'intero dataset conta quindi 42 proteine totali.

Tab. 7.1: descrizione dei semi dei conformi

Proteina	**Superfamiglia**	**famiglia**	**#Modelli**
1CLL	EF-Hand	Calmodulin-like	20
1QR0	Phosphopantetheinyl transferase	Phosphopantetheinyl Transferase SFP	20

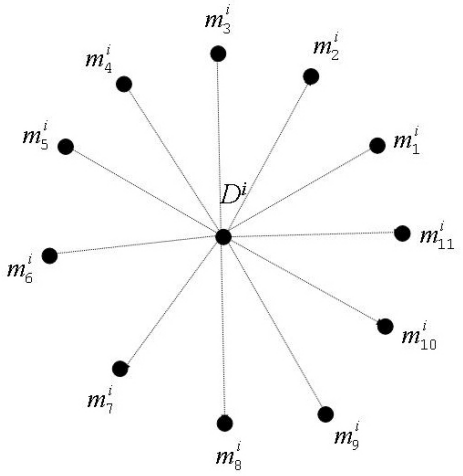

Fig 7.2: distanze tra conformi e seme

In fig. 7.2 viene schematizzata la distanza delle proteine interne ad un insieme, il seme è rappresentato al centro in quanto genera ogni altro modello. La distanza di ogni modello dal seme viene riscontrata solo tramite piccole differenze di forma tra le due superfici. Le proprietà fisico-chimiche di superficie non presentano assolutamente delle variazioni e questo rende, naturalmente, tutti i conformi biologicamente uguali. In fig. 7.1 viene mostrato un seme (1CLL) e un sottoinsieme dei conformi da lui generati.

7.1.2. Proteine mutate

Il secondo dataset adottato è stato ottenuto da quattro differenti proteine (fig. 7.3) che sono state sottoposte, in maniera progressiva, a delle mutazioni allo scopo di ottenere 98 modelli che sono via via differenti dal seme che le ha generate.

Tab. 7.2: descrizione dei semi dei mutanti

Proteina	**Superfamiglia**	**famiglia**	**#Mutazioni**
1CLL	EF-Hand	Calmodulin-like	12
1IRJ	EF-Hand	S100	13
2PVB	EF-Hand	Parvalbumin	11
1QR0	Phosphopantetheinyl transferase	Phosphopantetheinyl Transferase SFP	13

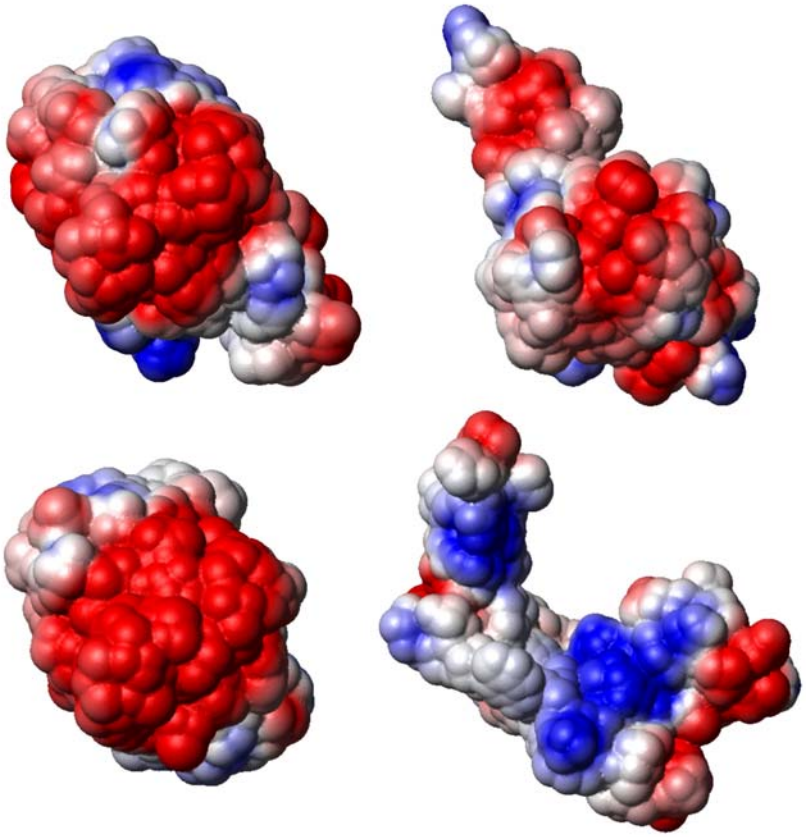

Fig 7.3: potenziale elettrostatico per i quattro semi

Le caratteristiche dei semi sono riportate in tab. 7.2, tutti sono simili per dimensione e tre di questi provengono dalla stessa superfamiglia secondo la classificazione SCOP .

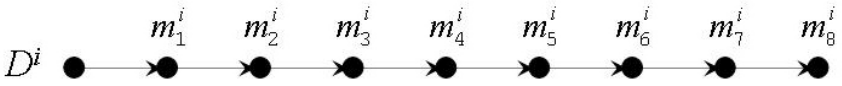

Fig 7.4: distanze tra mutanti e seme

Una singola classificazione consiste nel rimpiazzare cinque aminoacidi contemporaneamente che sono vicini sulla superficie proteica fino a quando tutti gli aminoacidi di superficie sono stati rimpiazzati. Le mutazioni sono state effettuate secondo la tecnica dell'*homology modelling* che garantisce di ottenere modelli con una configurazione stabile, il controllo delle strutture molecolari ottenute è stato effettuato con il tool PROSA II . Si noti che le 98 proteine di questo dataset non sono rappresentative, in termini di struttura, dell'intero PDB (sez. 2.1.4) ma

risultano rappresentative dal punto di vista delle proprietà di superficie in quanto presentano naturalmente l'intero insieme di aminoacidi in superficie e quindi coprono l'intera gamme di valori per le proprietà fisico-chimiche considerate in questa ricerca.

Nel dataset dei mutanti, in maniera più dettagliata, ogni proteina-seme genera un gruppo di modelli di proteine-mutanti distribuiti lungo due catene: una è generata utilizzando delle mutazioni conservative, l'altra utilizzando mutazioni non-conservative. Le mutazioni conservative portano delle piccole modifiche in termini di forma, mentre conservano le proprietà fisico-chimiche; ad esempio rimpiazzando l'acido aspartico con l'acido glutammico, che presentano entrambi una carica negativa, viene effettuata una mutazione conservativa. Differentemente, tramite le mutazioni non conservative vengono effettuate comunque delle piccole modifiche nella forma della superficie ma principalmente vengono modificate in maniera rilevante le proprietà di superficie; sostituendo l'acido aspartico con la treonina il potenziale elettrostatico in superficie viene pesantemente modificato in quanto il primo aminoacido è carico negativamente mentre il secondo lo è positivamente. Il numero di mutanti lungo la stessa catena varia da 11 a 13 in base alla lunghezza della catena polipeptidica del seme.

All'interno delle stesso gruppo di mutanti generate da un unico seme è possibile definire una funzione di distanza tra coppie di proteine.

Definizione: Sia data una proteina $D^i \in D$ e una sua catena di mutazioni $MC^i = \{m_1^i, ..., m_n^i\}$, la distanza tra due modelli $m_j^i \in MC^i$ e $m_k^i \in MC^i$ è definita come il numero di mutazioni effettuate ad un modello per ottenere il secondo: $dist(m_j^i, m_k^i) = | j - k |$.

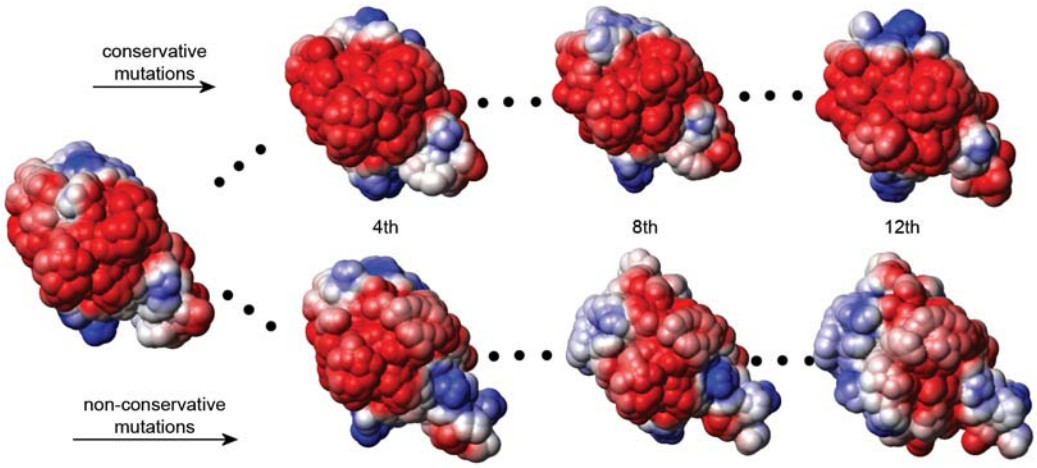

Fig 7.5: potenziale elettrostatico per sei mutanti, tre conservativi e tre non conservativi

In fig. 7.5 è mostrata la distribuzione del potenziale elettrostatico sulla superficie di 6 delle 24 mutazioni per il seme 1CLL. Si può notare che, dal punto di vista strutturale, tutti i mutanti derivanti dallo stesso seme sono fortemente correlati in quanto le mutazioni, conservative e non conservative, impattano in maniera limitata la forma della superficie. Differentemente, analizzando le proprietà di superficie, le mutazioni non conservative impattano pesantemente le proprietà fisico-chimiche a differenza delle mutazioni conservative che non alterano in maniera consistente queste proprietà. Questo porta la catena dei mutanti conservativi ad essere molto simile all'insieme dei modelli conformi definiti precedentemente in sez. 7.1.1. Considerando nuovamente i differenti tipi di mutazioni risulta evidente che la distanza tra due mutanti contigui nella catena conservativa è molto inferiore dal punto di vista biochimico alla distanza tra due modelli nella catena non conservativa, ciò significa anche che l'ultimo mutante nella catena conservativa, anche avendo avuto gli aminoacidi di superficie completamente sostituiti, presenta delle similarità con il seme a differenza dell'ultimo mutante non conservativo che si ritrova ad avere l'intera superficie completamente compromessa nelle proprietà di superficie. In fig. 7.6a viene mostrato che gli atomi di superficie delle due catene vengono modificati in maniera costante nel susseguirsi delle mutazioni.

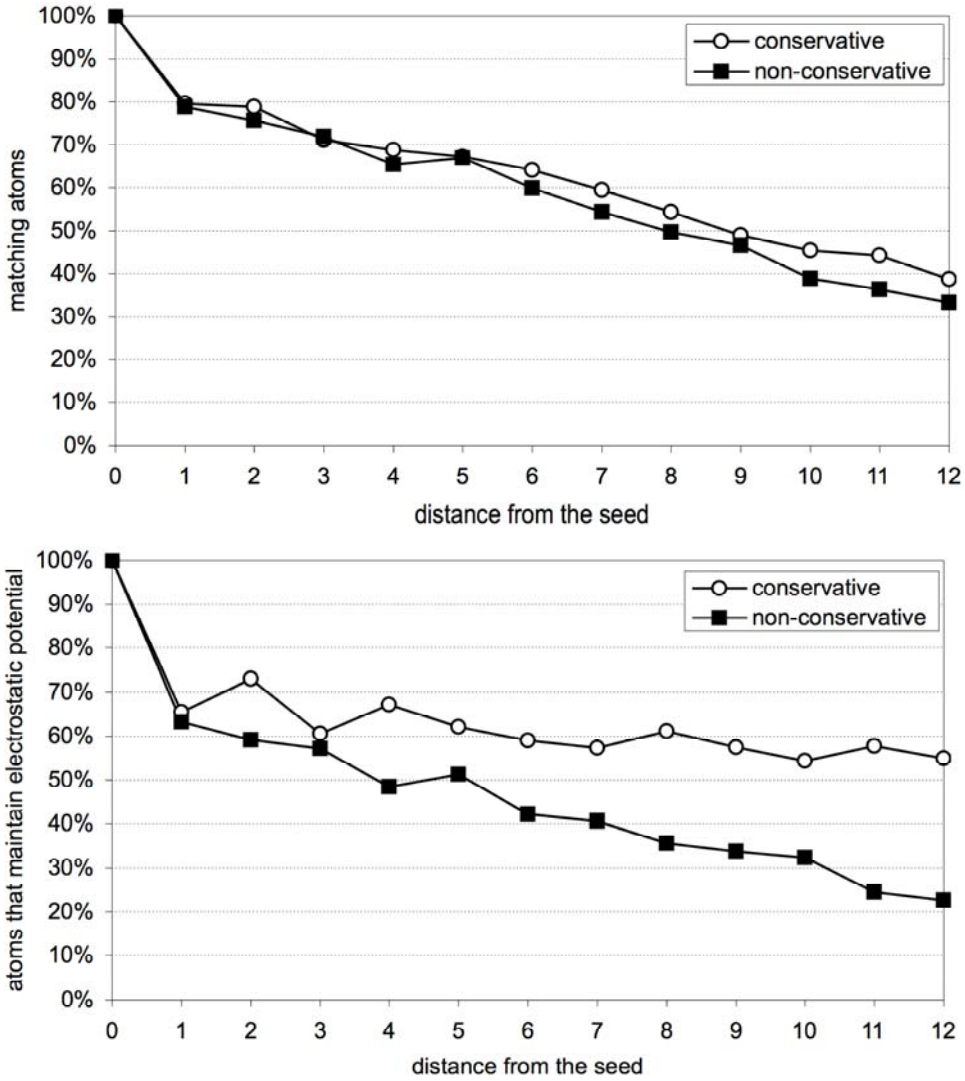

Fig 7.6: (a) percentuale degli atomi conservati in superficie, (b) percentuale degli atomi di superficie che mantengono invariata la proprietà di potenziale.

Questo significa che entrambi i tipi di mutazioni modificano un numero pari di atomi di superficie. In fig. 7.6b viene invece mostrato come varia la proprietà di potenziale elettrostatico lungo la catena. Risulta evidente che l'impatto su questa proprietà dato dalle mutazioni non conservative è maggiore di quello dato dalle conservative, infatti il numero di atomi che mantengono la stesso potenziale decrementa maggiormente per i mutanti non-conservativi.

7.1.3. Proteine reali

Il terzo dataset di proteine utilizzato è formato da tutte proteine reali. Questo dataset conta 25 proteine estratte dal PDB ed è stato scelto in quanto già utilizzato per testare l'approccio alla classificazione di proteine basato sulla superficie

introdotto in . La scelta di queste proteine è stata suggerita da considerazioni sulla classificazione SCOP. Infatti si sono scelte proteine appartenenti a 5 famiglie estremamente diverse in modo da inserire nel dataset proteine rappresentative anche dal punto di vista strutturale a differenza dei due dataset descritti in sez. 7.1.1 e 7.1.2. Le proteine scelte appartengono alle famiglie: hemoglobins; ureases; crambin-like; seryl-tRNA synthetases e hydrolases.

7.2. Analisi dei risultati

7.2.1. Clustering

I test sperimentali, effettuati sui tre algoritmi di clustering, sono mirati alla valutazione degli indici di omogeneità, regolarità e copertura introdotti in sez. 4.2.5. È altrettanto importante valutare la robustezza dei metodi.

In tab. 7.3 sono riportati i valori medi degli indici di valutazione qualitativa dei clustering, i commenti su questi valori sono rimandati alla fine di questa sezione.

Tab. 7.3: Valori medi delle funzioni obiettivo del clustering

	avg hom(C)	avg reg(C)	avg cov(C)
Region growing	1	0.49	0.55
Template matching	0.85	0.74	0.44
Overlapped region growing	0.91	0.72	0.75

La *robustezza* è definita come la capacità di un algoritmo di ottenere risultati simili se sottoposto a input simili. In questa circostanza, un algoritmo di clustering superficiale è robusto se restituisce clustering simili date proteine simili. Un Algoritmo robusto definisce clustering simili anche in presenza di rumore su superfici proteiche o in presenza di piccoli cambiamenti che non devono essere riscontrati nel clustering.

La robustezza sarà calcolata utilizzando il dataset delle proteine conformi in quanto tutte le superfici sono dal punto di vista biologico uguali ma, analizzando i dati, si possono scorgere differenze numeriche. In fig. 7.1 è possibile apprezzare le piccole differenze di superficie nelle diverse proteine conformi.

Per poter misurare la robustezza è stato utilizzato il coefficiente di Jaccard leggermente modificato come descritto in seguito. Siano prese due proteine conformi D^i e D^j e i relativi insiemi di ULR, U^i e U^j, ottenuti tramite un

algoritmo di clustering. La caratteristica delle proteine conformi, che ci permette di calcolare la robustezza, è che preso un atomo di superficie su una è sempre possibile trovare il corrispondente sulla seconda. In base a questa corrispondenza è possibile definire SS come il numero di coppie di atomi che in entrambe le proteine appartengono allo stesso cluster e SD come il numero di coppie che appartengono allo stesso cluster su una proteina mentre non lo sono sull'altra. Il coefficiente di Jaccard è definito come segue:

$$JAC = \frac{SS}{SS + SD}$$

da notare che, nel calcolo di tale coefficiente, è stata introdotta una ULR fittizia in entrambe le proteine contenente tutti gli atomi che non sono stati assegnati a nessun cluster, questo permette di calcolare il coefficiente in maniera più corretta in quanto anche gli atomi non assegnati su una superficie devono rimanere tali anche nella seconda se considerati non caratterizzanti dall'algoritmo di clustering.

In tab. 7.4 sono riportati i valori del coefficiente di Jaccart ottenuti dai diversi algoritmi di clustering.

Tab. 7.4: Valori medi del coefficiente di Jaccart

	JAC
Region growing	0.29
Template matching	0.32
Overlapped region growing	0.54

I risultati sperimentali sui diversi algoritmi di clustering provano una maggiore capacità dell'*overlapped region growing* nel catturare le caratteristiche della superficie proteica. Questa capacità è dovuta principalmente al fatto che esso conduce verso un clustering maggiormente coprente rispetto agli altri due mantenendo comunque i valori di omogeneità e regolarità a livelli confrontabili. Anche la prova di robustezza indica in questo algoritmo quello meno sensibile alle piccole modifiche di superficie. La maggior robustezza è ottenuta per la possibilità di associare gli atomi grigi (in riferimento al metodo di discretizzazione delle proprietà descritto in sez. 4.2.4) a più ULR. Infatti un atomo grigio conteso tra due

regioni può spostarsi da una all'atra se sottoposto a lievi variazioni di superficie, questo effetto è meno evidente se lo stesso atomo ha la possibilità di appartenere ad entrambe le regioni.

7.2.2. Mining

L'algoritmo di mining è stato testato utilizzando il dataset dei mutanti, tramite il quale è possibile valutare come la similarità di superficie decresce. La dimensione dei pattern infatti decresce all'aumentare della distanza tra mutanti. Si ricorda che in questo dataset è possibile definire la distanza tra proteine come il numero di mutazioni effettuate tra loro (vedi sez. 7.1.2).

In tab. 7.5 è riportata una valutazione statistica dei pattern estratti sul dataset. Ogni riga riporta il numero di pattern trovati per livello ($\#Pat$), la dimensione media del loro supporto ($\#Sup$), la media dell'eterogeneità del supporto calcolata come la percentuale dei pattern supportati da mutanti di semi diversi ($Het\%$), la percentuale dei pattern su mutanti conservativi ($Cons\%$).

Tab. 7.5: Valutazione statistica dei pattern estratti

Livello	#Pat	#Sup	Het%				Cons%
			1	2	3	4	
1	111267	85.13	0.01	0.06	0.10	99.83	50.43
2	208582	25.39	1.07	3.61	13.88	81.45	51.92
3	86028	3.91	40.71	34.18	19.94	5.17	59.35
4	45601	2.42	94.86	5.05	0.09	0.00	66.01
5	26714	2.15	99.82	0.18	0.00	0.00	66.07
6	11783	2.05	100.00	0.00	0.00	0.00	69.22
7	3649	2.02	100.00	0.00	0.00	0.00	69.36
8	770	2.01	100.00	0.00	0.00	0.00	67.14
9	92	2.00	100.00	0.00	0.00	0.00	61.96
10	6	2.00	100.00	0.00	0.00	0.00	66.67

I risultati indicano che i pattern danno una buona indicazione delle similarità tra superfici proteiche, in particolare l'importanza del pattern incrementa proporzionalmente al suo livello: infatti mentre i pattern corti sono condivisi da

molti mutanti anche se di semi diversi, pattern lunghi sono maggiormente rappresentativi di una sola catena (insieme dei mutanti di un seme) dove la similarità è maggiore; i pattern di livello 5 sono condivisi tra proteine provenienti dallo stesso seme per il 99.82% dei casi. Da notare che oltre il 60% dei pattern lunghi sono condivisi tra mutanti conservativi, riflettendo il fatto che quelli non-conservativi sono soggetti ad alterazioni maggiori nella superficie.

7.2.3. Classificazione

Il dendrogramma restituito nella fase di classificazione permette di dare una valutazione generale di tutto l'approccio. È possibile infatti confrontare la classificazione con le caratteristiche conosciute del dataset dei mutanti. Una ulteriore prova dell'approccio sarà effettuata su proteine reali estratte dal PDB. I criteri di scelta che hanno portato alla definizione di questo insieme sono descritti in sez. 7.1.3. Il dataset dei mutanti ci perfette di valutare l'efficacia del metodo mentre il secondo è da considerarsi una prima applicazione su dati reali.

Per verificare l'efficacia del metodo viene introdotto l'errore intra-seme (*ISe*). Un *ISe* occorre quando una classe include mutanti non consecutivi nella catena delle mutazioni di uno stesso seme; ad esempio una classe che presenta il primo, il terzo e il quinto mutante di un seme comporta due *ISe*. Viene poi definita la misura di sparpagliamento (*scattering*) che indica se le catene dei mutanti sono ben suddivise o vedono le proprie proteine sparse in classi diverse. Date g classi $\{C_1,...,C_g\}$, lo scattering è definito come la media del rapporto del numero di *ISe* in ogni classe e la cardinalità della classe:

$$\frac{1}{g}\sum_{i=1}^{g}\frac{dist_i + 1\ \#C_i}{\#C_i}$$

dove $dist_i$ è la massima distanza tra due mutanti della classe C_i. In fig. 7.7 è mostrato il dendrogramma per il dataset dei mutanti. L'etichetta associata ad ogni mutante rappresenta:

- la prima cifra indica il seme che ha lo ha generato (0, 1, 2 e 3);
- la seconda cifra indica il tipo di mutazione che lo ha generato (1 conservativa, 2 non-conservativa)
- le ultime due indicano la posizione nella catena delle mutazioni (valori elevati indicano una distanza maggiore).

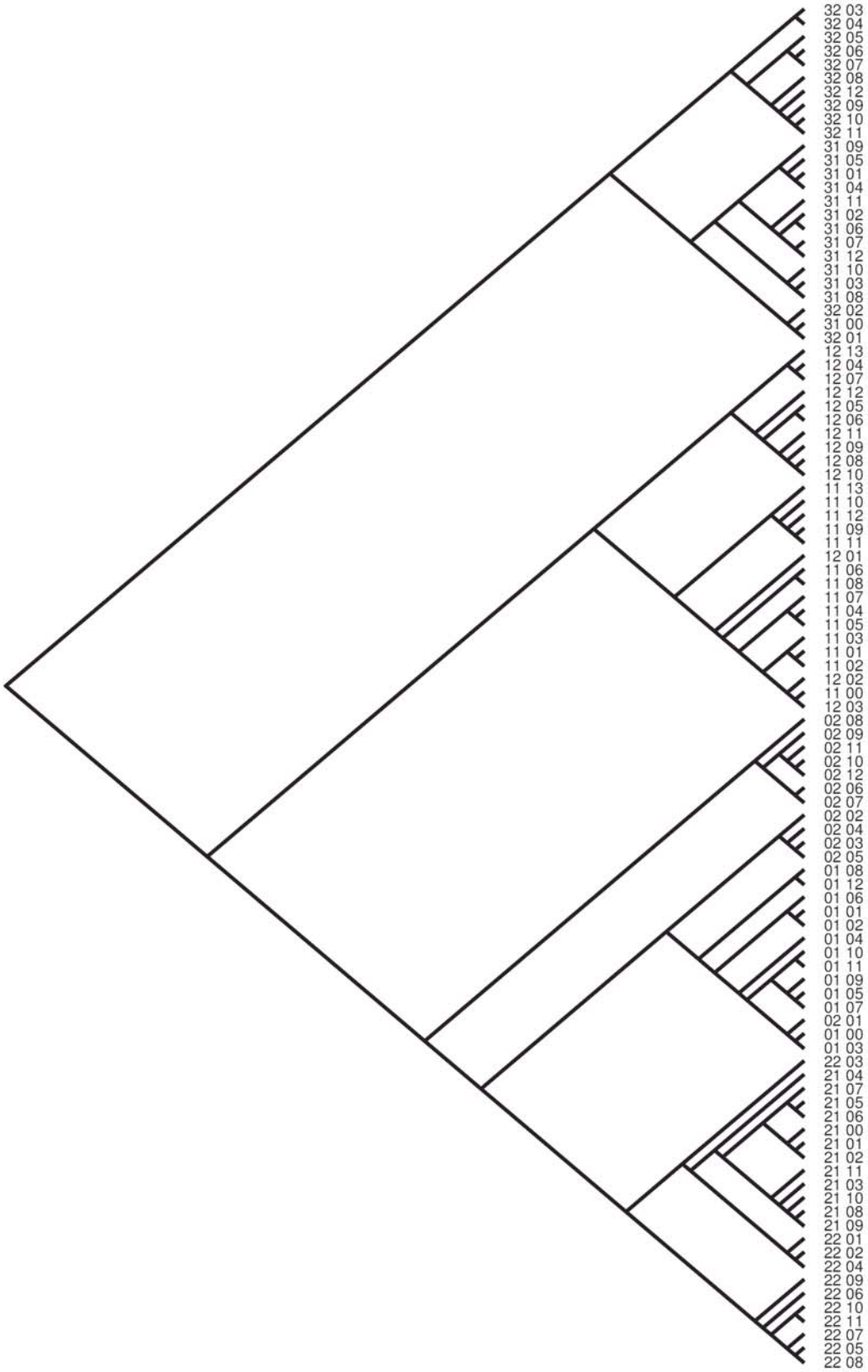

Fig 7.7: dendrogramma per il dataset dei mutanti.

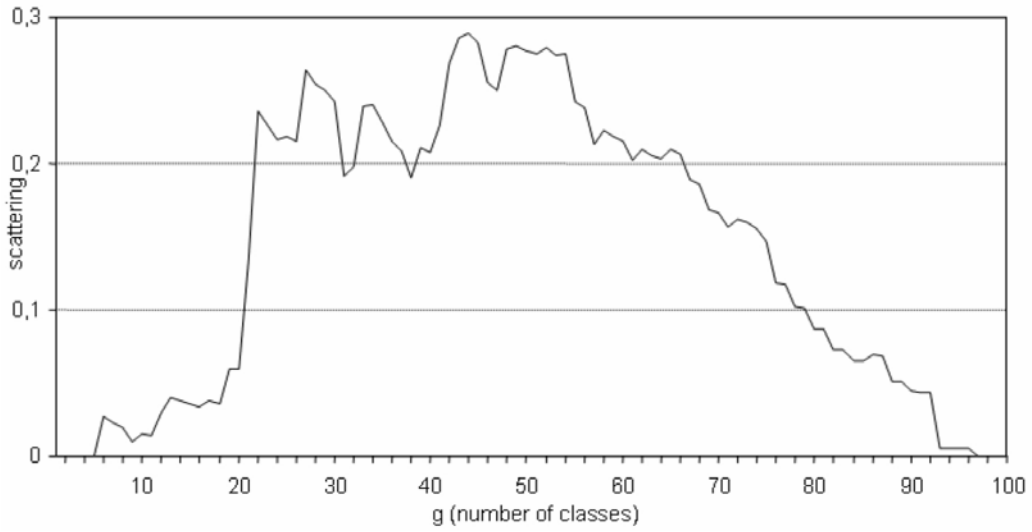

Fig 7.8: scattering per diversi divelli del dendrogramma.

La fig. 7.8 mostra come lo scattering varia all'incrementare del numero g di classi (corrispondenti a diverse profondità del dendrogramma); alla sinistra del grafico esiste una sola classe contenente tutti i mutanti, mentre a destra tutte le classi contengono un'unica proteina (tutti singeton). Lo scattering rimane sempre ridotto mostrando come la classificazione riesce a cogliere le differenze di superficie nelle catene dei mutanti, infatti il numero medio di *ISe* per cluster rimane sempre sotto lo 0.3. Da una analisi più approfondita risulta che il numero di *ISe* commessi nei mutanti conservativi è tre volte superiore rispetto a quello dei non-conservativi. Questo prova il fatto che le lievi differenze introdotte dalle mutazioni conservative rendono più difficoltosa la distinzione delle superfici.

Di seguito saranno riportate delle valutazioni aggiuntive emerse discutendo i risultati con gli esperti del dominio applicativo. Un'analisi più approfondita del risultato della classificazione può essere effettuata tramite la matrice di similarità riportata in fig. 7.9.

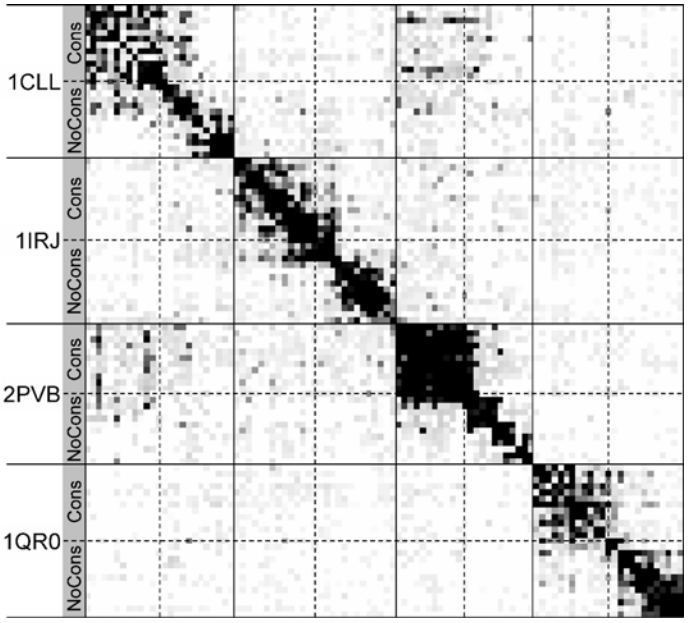

Fig 7.9: matrice di similarità tra mutanti, colori scuri indicano una similarità maggiore

Tale matrice riporta graficamente i valori di similarità ottenuti tramite la funzione descritta in sez. 6.6.1. Dalla figura emerge quanto segue:

1. I valori di similarità tra il seme e i suoi mutanti conservativi è, in media, quattro volte superiore dei valori di similarità con i non-conservativi. Ad ogni modo il valore medio di similarità tra il seme e tutti i suoi mutanti è 20 volte superiore rispetto al valore medio di similarità tra un seme e i mutanti non da lui generati. Questo conferma l'effettiva capacità del metodo di cogliere i diversi livelli di similarità di questo dataset.
2. La similarità tra mutanti conservativi generati dai semi 1CLL e 2PVB risultano essere superiori alla media degl'altri semi. È stato interessante scoprire che questa elevata similarità ha una spiegazione biologica, la quale è rimandata al termine di questo elenco.
3. La media delle similarità tra mutanti generati dai tre semi appartenenti alla stessa superfamiglia (vedi fig. 7.9) sono tre volte superiori di quelli del quarto seme proveniente da un'altra superfamiglia. Il metodo è stato in grado di riconoscere le due superfamiglie del dataset. Secondo il dendrogramma in fig. 7.7 prima dell'ultima fusione tutti i mutanti relativi ai semi di tipo *EF-hand* sono disposti in una classe e i restanti in un'altra.

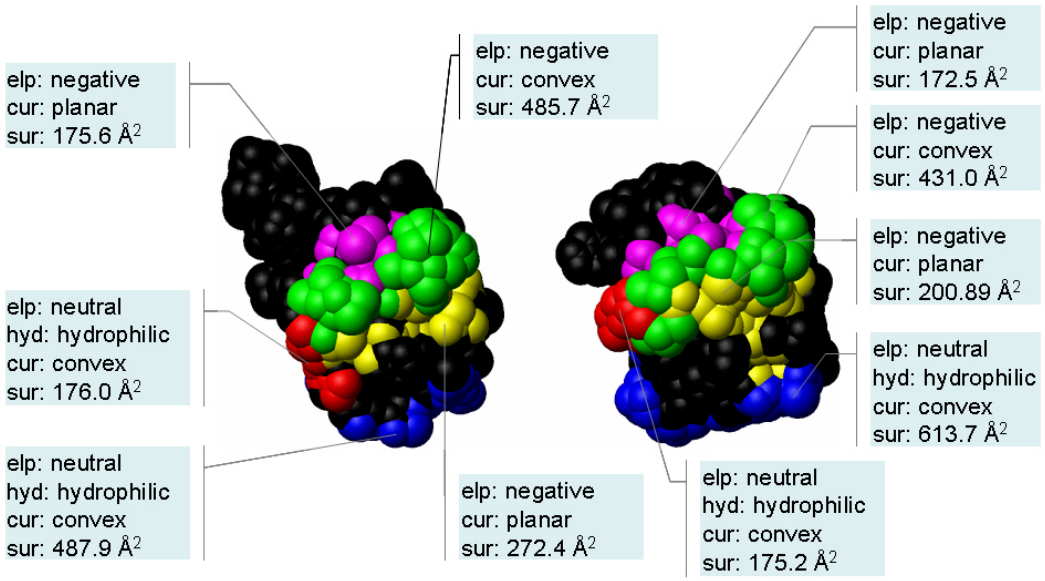

Fig 7.10: pattern condiviso tra i semi 1CLL e 2PVB, le ULR sono etichettate con i valori delle loro proprietà

L'alta similarità riscontrata tra conservativi generati dai semi 1CLL e 2PVB è dovuta ad una comune funzionalità, ossia entrambe le proteine hanno la capacità di legare il calcio. Come mostrato in fig. 7.10 esiste un pattern condiviso tra le due superfici proteiche, questo identifica la zona interessata nell'interazione col calcio. Questa similarità di superficie rende tutti i mutanti conservativi molto simili tra loro in quanto rimane conservata. Questo accade anche nel processo evolutivo quando una interfaccia viene mantenuta nel corso del tempo se necessaria alla vita dell'organismo.

Un ulteriore test è stato effettuato su un insieme di proteine reali. In fig. 7.11a è mostrata la matrice di similarità mentre in fig. 7.11b è riportato il dendrogramma relativo.

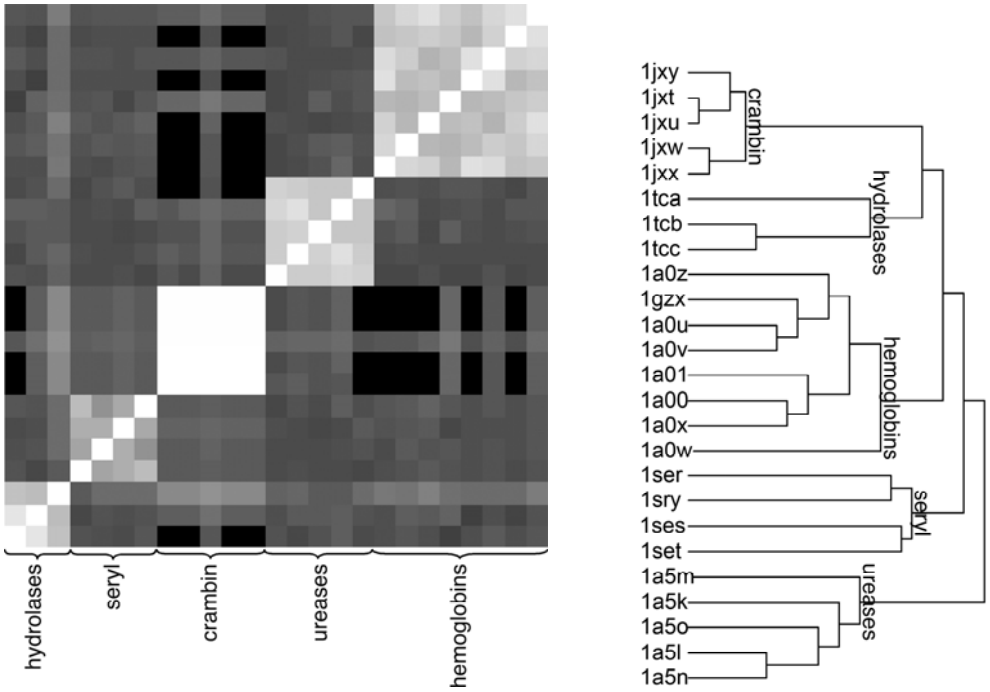

Fig 7.11: (a) matrice di similarità e (b) dendrogramma della classificazione ottenuta per le cinque famiglie,

Nella matrice, le cinque famiglie sono ben riconoscibili (cinque quadrati più chiari), questo indica che la similarità intra-famiglia è molto maggiore della inter-famiglia. Anche nel dendrogramma le cinque famiglie sono state correttamente classificate confermando i buoni risultati dei test precedenti.

8. Un tool per l'analisi di superfici proteiche

Il progetto di ricerca descritto in questa tesi si pone come obiettivo l'individuazione di similarità di superficie tra diverse proteine. Come descritto nel secondo capitolo, il problema della classificazione di superficie rimane allo stato attuale un campo aperto di ricerca e ancora oggigiorno non esistono classificazioni di riferimento per la validazione del nostro approccio. Una via alternativa che prova l'efficacia del metodo è la visualizzazione e valutazione dei pattern complessi di superficie, e quindi di similarità evidenziate su diverse superfici, da parte di esperti del dominio. A tale scopo è stato necessario studiare le applicazioni attualmente disponibili per la visualizzazione di proteine e implementare un tool ad hoc per l'analisi dei pattern complessi di superficie. Tale tool, chiamato PDBVision, è attualmente in via di sviluppo e si è rivelato estremamente utile per il testing di ogni singolo algoritmo e dell'approccio globale.

In questo capitolo: verranno introdotti inizialmente i maggiori tool disponibili per la visualizzazione di superfici proteiche e le eventuali caratteristiche trattate; saranno poi descritte le funzionalità richieste da questo strumento; verrà descritta la realizzazione di un progetto che comprende tutti i passi del progetto di ricerca descritto nei capitoli 3.

8.1. Tool esistenti

8.1.1. MolMol

MolMol è l'acronimo di MOLecule analysis and MOLecule display. È un programma grafico che permette lo studio di macromolecole biologiche. Gli autori del programma sono Reto Koradi, informatico che ha concluso il dottorato nel gruppo di biofisica del prof. Wuthrich, il PD Martin Billeter, docente privato nell'istituto per la biologia molecolare e fisica, e il prof. Kurt Wuthrich, coordinatore del gruppo.

MolMol è in grado di visualizzare, analizzare e manipolare strutture tridimensionali di proteine e acidi nucleici derivati dalla risonanza magnetica nucleare (NMR). In fig. 8.1 viene mostrata un'istanza della finestra principale del tool. Ogni elemento dell'interfaccia verrà brevemente illustrato:

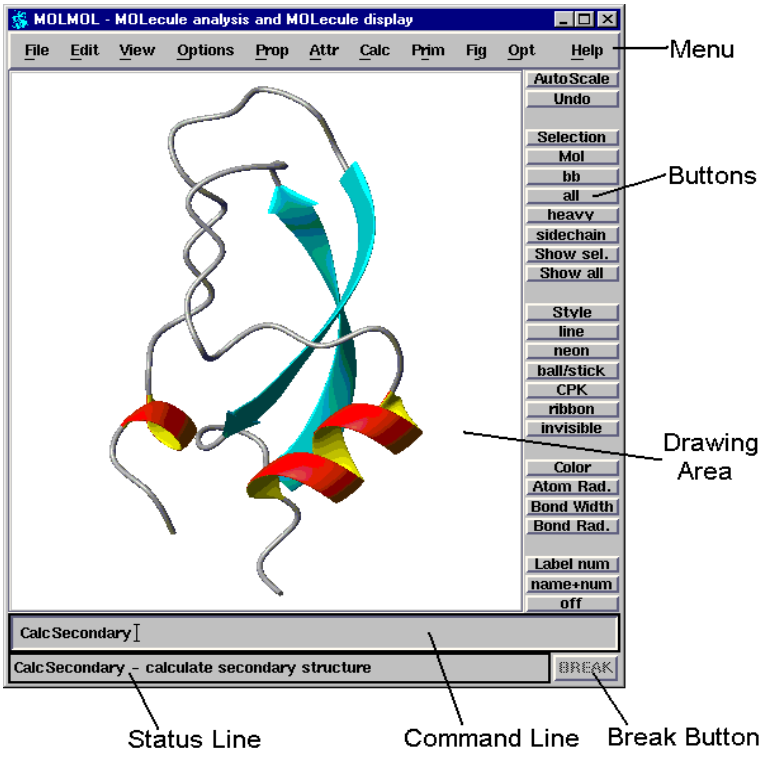

Fig 8.1: Interfaccia principale

Ogni visualizzazione grafica di molecole avviene nella *Drawing area* all'interno della quale è possibile interagire effettuando trasformazioni quali zoom e rotazione della struttura mediante il mouse.

Il menu contiene tutti i comandi di MolMol. Il menu *File* contiene comandi relativi all'input e all'output delle molecole in differenti formati, alla stampa di grafici e all'esecuzione di macro (che possono essere definite anche dall'utente). Attraverso il menu *View* è possibile cambiare la visualizzazione globale della struttura modificandone i parametri. Il menu *Calc* permette di effettuare calcoli e analisi delle molecole. L'elemento *Buttons* da accesso velocemente a comandi frequentemente utilizzati. All'interno della *Command Line* è possibile digitare il comando desiderato (alternativo al menu). La *Status Line* visualizza semplicemente messaggi di aiuto, mentre *Break Button* termina un comando che si sta eseguendo.

Questo tool è in grado di leggere differenti formati di file come: DIANA, DG, PDB, etc. L'ultimo formato è stato introdotto in sez. 2.1.4 e risulta essere quello più diffuso. Oltre alla visualizzazione, tramite MolMol è possibile modificare la molecola per poi salvarla naturalmente sempre come file PDB. Esempi di modifiche che si possono apportare sono:

- Aggiunta e rimozione di residui

- Aggiunta e rimozione di atomi
- Aggiunta e rimozione di legami
- Aggiungere e sottrarre la distanza tra atomi

La visualizzazione dell'immagine può essere modificata ed adattata alle proprie esigenze tramite operazioni di zoom, modifica del colore di sfondo, aggiunta e rimozione di piani di taglio, *clipping planes*, che eliminano porzioni della molecola. Inoltre è possibile modificare l'angolatura dalla quale si osserva la molecola e modificare luce e nebbia per percepire meglio l'illusione della tridimensionalità. Il tool comprende una serie di comandi che modificano gli attributi di oggetti come gli atomi, i legami, le distanze tra gli atomi, etc. È possibile modificare il raggio e il colore dei vari oggetti che compongono la molecola; inserire texture, applicare ombre e stili differenti.

È possibile stampare e salvare differenti diagrammi in vari formati: bmp, jpeg, Tiff, VRML, etc.

La lista di comandi di calcolo prevede la possibilità informazioni di vario genere relative alla molecola (o alla lista di molecole) correntemente analizzata. Ad esempio l'utente può calcolare la struttura secondaria di una proteina (MolMol utilizza l'algoritmo pubblicato da Kabsch/Sander); può selezionare un gruppo di atomi e assegnare ad essi una superficie; può calcolare la distanza tra due atomi vicini tra loro oppure trovare la coppia di atomi con la distanza minima.

Anche le trasformazioni sono complete. È possibile infatti ruotare la proteina fissando uno o più assi, centrarla e scalarla nel modo che la visualizzazione riempia la finestra nel miglior modo possibile.

MolMol utilizza il principio di *selezione-azione*, il che significa che prima occorre selezionare un gruppo di oggetti, poi è possibile eseguire una serie di comandi su di essi. Gli oggetti che possono essere selezionati sono:

- Molecole
- Residui
- Atomi
- Legami
- Angoli
- Distanze
- Primitive (annotazioni, superfici, etc.)

Ogni oggetto elencato è parte di una gerarchia:

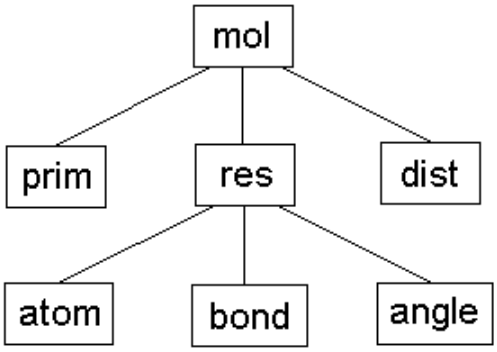

Fig 8.2: Gerarchia utiilzzata da Mol Mol

Dalla fig. 8.2 si evince che gli atomi fanno parte di residui e che i residui fanno parte di molecole. Ciò ha un effetto diretto sulla selezione: se si seleziona un oggetto, l'oggetto padre nella gerarchia viene a sua volta selezionato risalendo via via alla radice. Per ogni oggetto selezionato è poi possibile visualizzarne le proprietà che le descrivono **[MOL06]**.

8.1.2. Protein explorer

Protein Explorer è programma freeware che permette di trovare, visualizzare ed esplorare una struttura molecolare tridimensionale. Una delle caratteristiche che lo contraddistingue da altri tool è la seguente: Protein Explorer viene eseguito su un comune browser web. Per questo motivo viene definito *tool-web.*

All'apertura del software viene visualizzata la schermata iniziale, chiamata *FirstView*, la quale permette una facile e veloce visualizzazione di proteine e catene di acidi nucleici oltre alla verifica della presenza di acqua, legami e legami covalenti.

La finestra principale è composta da tre frame. Quello principale viene chiamato *Molecular Image Frame* e viene utilizzato per mostrare la struttura molecolare corrente. Per apprezzare meglio la tridimensionalità dell'immagine, l'utente può ruotare la molecola attorno ai propri assi o visualizzarla da differenti angolature.

Il secondo frame è chiamato *Control Panel Frame* dove saranno presenti i controlli per esplorare l'immagine molecolare. L'organizzazione di questi comandi è semplice e il pannello è corredato con diverse informazioni utili per l'utente inesperto. Inoltre qui verranno elencate le informazioni messe a disposizione dall'autore del file che descrive la struttura della molecola. Generalmente questo tipo di informazioni vengono prelevate dal campo HEADER del file pdb in input.

Infine vi è il *Message Frame* nel quale verranno visualizzate le informazioni di atomi, catene e residui di aminoacidi selezionati dall'immagine.

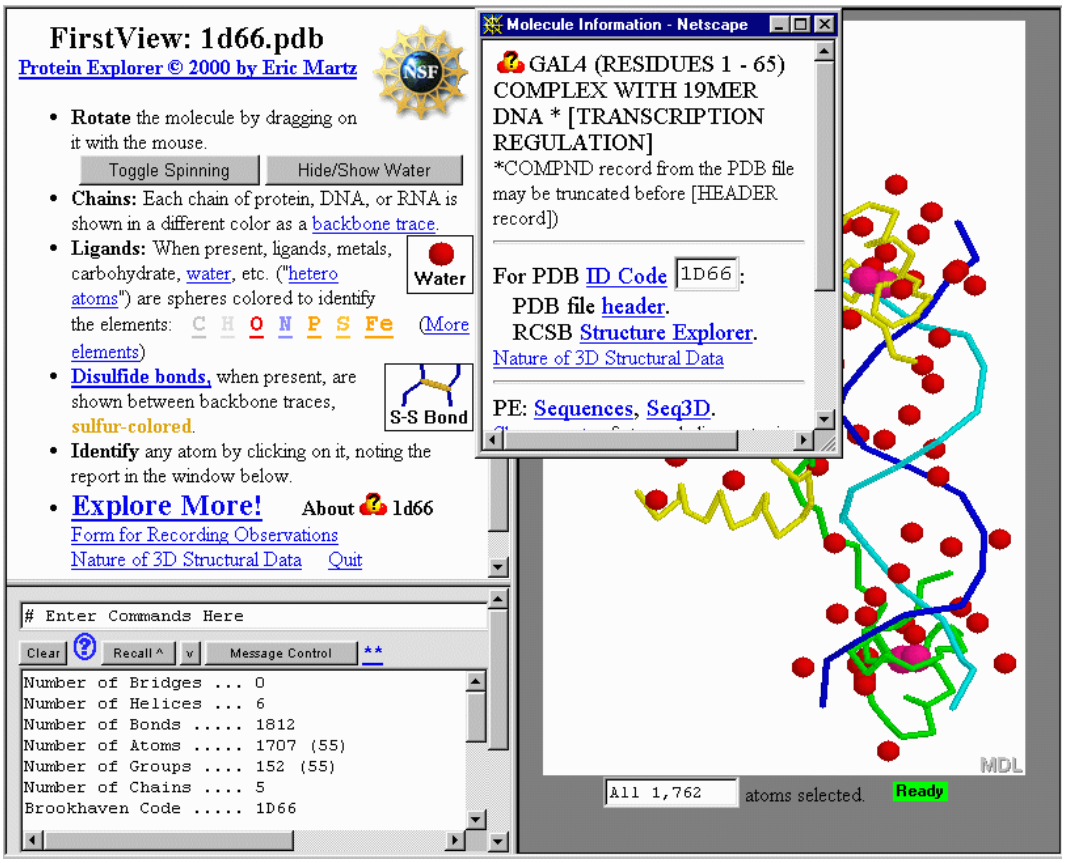

Fig. 8.1: First View in Protein Explorer

Inizialmente l'immagine mostrata sarà composta da sfere rosse, le quali non sono altro che la rappresentazione di atomi di acqua. Questi atomi, come ogni atomo presente nella molecola, possono essere visualizzati o nascosti. L'enfasi posta su questo tipo di atomi è dovuta all'importanza di conoscere se l'acqua sia presente o meno all'interno di una struttura. Facendo scomparire ogni atomo, la visualizzazione della molecola è limitata allo scheletro delle catene proteiche e del DNA. In questa modalità di visualizzazione è possibile mettere in evidenza eventuali legami chimici tra atomi.

Passando alla modalità di visualizzazione Quick View, l'utente è in gradi di personalizzare l'immagine molecolare in modo da rispondere alle proprie esigenze. Non appena si passa a questa modalità, i comandi che si trovano nel Control Panel e le informazioni contenute nella Message Box cambiano diventando più specifiche. Più in dettaglio, attraverso un menu di tipo pull-down è possibile selezionare un sottoinsieme di atomi modificarne la visualizzazione. Per esempio, il tipo di

visualizzazione *showfill* rappresenta gli atomi come sfere solide con raggio di Van der Waals. È possibile colorare in modo differente gli atomi a seconda della loro proprietà, quali la polarità, la struttura, la catena di aminoacidi di appartenenza, la carica, la temperatura, etc.

Protein explorer è un software potente e semplice da utilizzare per la visualizzazione di strutture molecolari, ma non è in grado di eseguire certe operazioni quali la modifica di legami all'interno della struttura e la visualizzazione di molecole specificate solamente da una sequenza di aminoacidi o di nucleotidi **[PEX06]**.

8.1.3. WebMol

WebMol è stato progettato e realizzato per visualizzare ed analizzare informazioni strutturali contenute in file di tipo PDB. Può essere eseguito come *applet java* oppure come applicazione di tipo *stand-alone*. La molecola descritta dal file di input viene rappresentata tramite linee che simulano i legami atomici. Tra le caratteristiche grafiche si sottolinea la possibilità di evidenziare in modo differente atomi in base al tipo, di mostrare la struttura secondaria, le catene di aminoacidi. L'utente utilizzerà il mouse per effettuare rotazioni, traslazioni e tagliare porzioni di molecola.

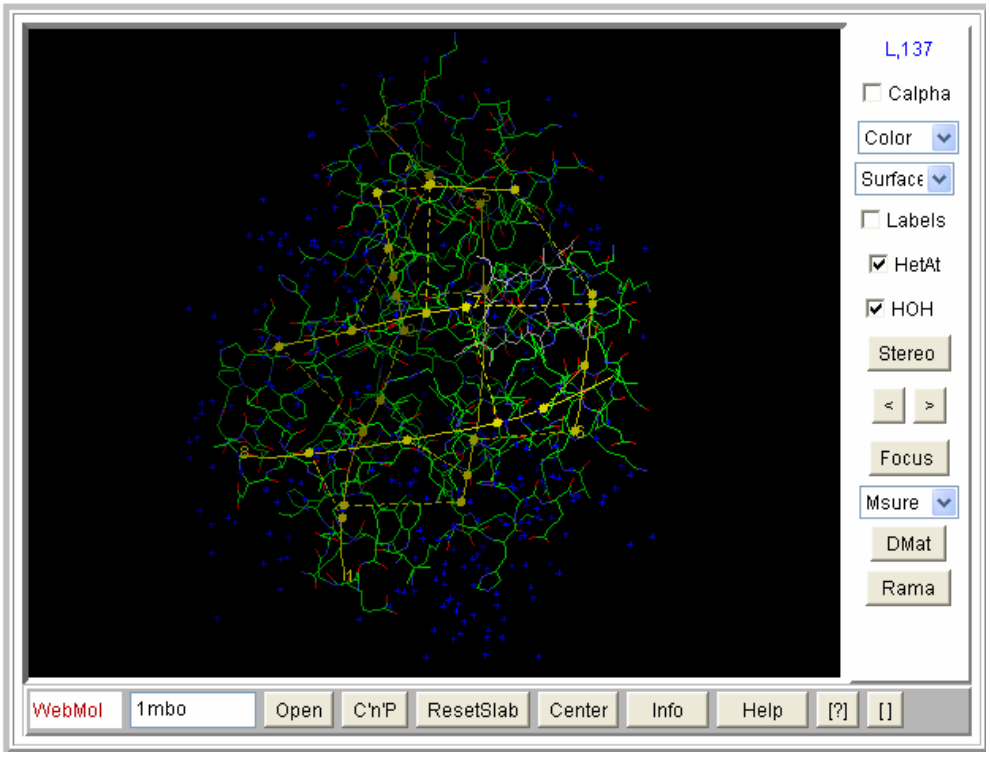

Fig. 8.2: Finestra principale

Tra le funzioni principali si annoverano la misurazione di distanze, angoli e andoli diedri; l'analisi di legami peptidici, di strutture secondarie; il rilevamento di legami di idrogeno con la catena principale; la possibilità di visualizzare e stampare diagrammi di Ramachandran **[WMO06]**.

8.1.4. Deep View Swiss-PDB Viewer

Swiss-Pdb Viewer è un'applicazione che fornisce un'interfaccia che permette di analizzare differenti proteine nello stesso tempo. A differenza di altri tools, l'utente ha la possibilità di sovrapporre proteine al fine di individuare o dedurre allineamenti strutturali, comparare le loro posizioni nello spazio e altri aspetti interessanti. Grazie alla grafica intuitiva è semplice estrarre informazioni su mutazioni di amminoacidi, legami, angoli e distanze tra atomi.

In più, Swiss-Pdb Viewer è strettamente legato a un omologo sistema chiamato Swiss-model, un server sviluppato dall'istituto di bioinformatica svizzero (SIB) in collaborazione con GlaxoSmithKline R&D e il Structural Bioinformatics Group.

L'applicazione carica e visualizza più molecole simultaneamente. Ogni molecola caricata appartiene ad un proprio livello, in modo da poter gestire singolarmente o simultaneamente una o più proteine. Ogni molecola è composta da gruppi (per esempio amminoacidi, nucleotidi, sottostrati, …) ognuno dei quali è composto da atomi, le cui coordinate spaziali sono prese direttamente dal file PDB in input. Ogni volta che viene effettuata una trasformazione sulla molecola (traslazioni e rotazioni) le coordinate di quest'ultima vengono modificate in accordo con le operazioni effettuate. È possibile quindi salvare la vista corrente oppure tornare alla configurazione originale.

Lo spazio di lavoro, o *workspace* è composto da differenti finestre, ognuna con la propria classe di funzionalità:

- La *main window* visualizza la molecola in input. Attraverso il mouse e il menu principale è possibile modificare la visualizzazione ed effettuare analisi
- Il *control panel* offre un modo semplice per selezionare e modificare le proprietà di gruppi componenti la proteina
- La *swiss-model window* mostra l'allineamento delle proteine selezionate

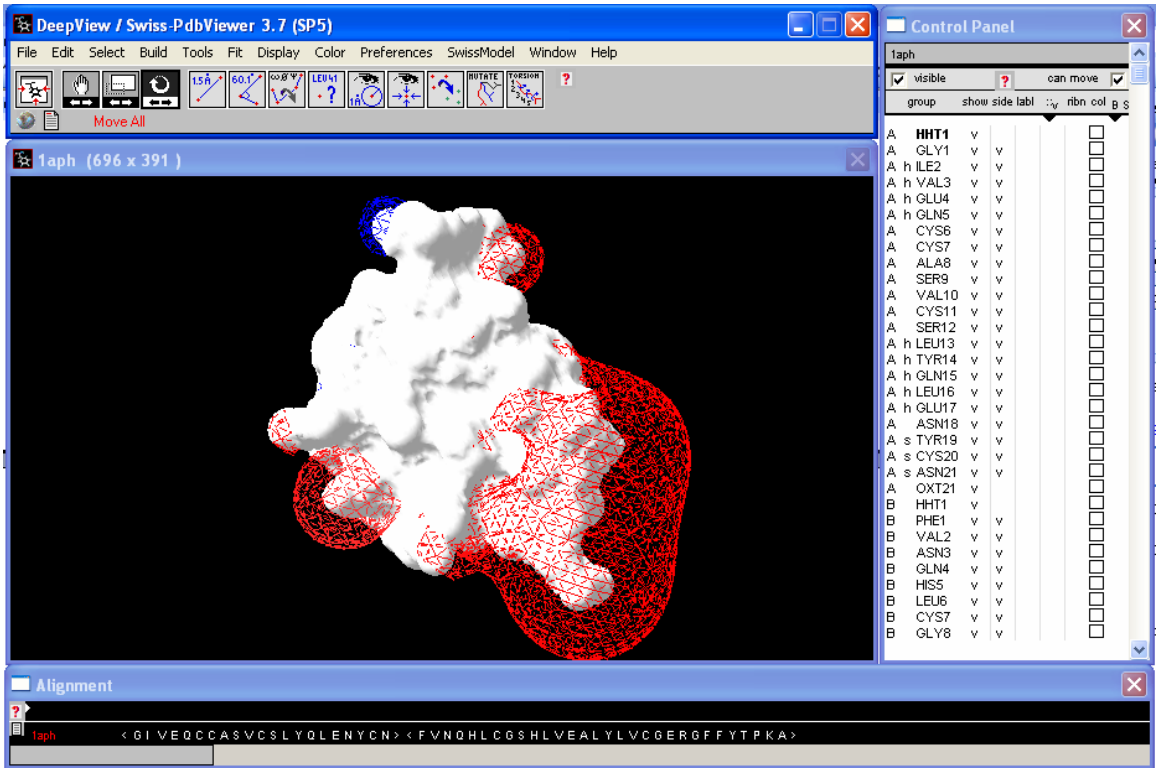

Fig. 8.3: Finestra principale Swiss-Pdb Viewer

La voce *Tool* riveste particolare importanza all'interno del menu principale. Comprende una serie di comandi e di funzioni per visualizzare e manipolare dati relativi alla proteina:

- Calcolo della distanza tra una coppia di atomi
- Misura dell'angolo tra una terna di atomi
- Misurare l'angolo di torsione tra legami
- Selezione e visualizzazione di informazioni relativi a uno specifico atomo
- Visualizzare/Nascondere gruppi di atomi ad una distanza personalizzabile dall'atomo selezionato
- Selezionare un atomo e utilizzandolo come baricentro della visualizzazione
- Mutare catene attraverso la selezione di tutti gli stati coerenti in cui un amminoacido può trovarsi. Questa funzionalità permette di valutare velocemente l'effetto di una mutazione, senza doverla verificare in laboratorio.
- Modificare la torsione di elementi quali angoli di amminoacidi e *hetero* atomi (come ossigeno e zolfo).

Tra i differenti tipi di visualizzazione di una molecola, risulta molto interessante la visualizzazione relativa al potenziale elettrostatico proteico, il quale è

responsabile del processo che da alla proteina la propria forma funzionale e conformazione. Swiss-Pdb Viewer fornisce due tipi differenti di visualizzazione:

- Proteina come mappa di potenziale elettrostatico, attraverso la quale è possibile modificare i valori del potenziale in superficie
- Proteina come superficie molecolare colorata dal potenziale elettrostatico (gradazioni di blu per potenziali positivi, gradazioni di rosso per potenziali negativi). Questo tipo di visualizzazione è più utile ai fini di comparare le superfici proteiche di proteine differenti **[SPV06]**.

8.2. Funzionalità supportate

PdbVision vuole essere uno strumento per la visualizzazione e l'analisi di superfici proteiche. L'obiettivo principale che si propone è quello di supportare la rappresentazione graficamente di superfici proteiche, con le relative proprietà, e di fornire un supporto per l'analisi proteica tramite pattern complessi di superficie. Nel tool sono quindi integrati i diversi algoritmi introdotti nei capitoli 4, 5 e 6.

Di seguito verranno descritte le funzionalità supportate dal tool.

8.2.1. Caricamento della proteina

L'applicazione si appoggia ad un archivio di proteine. Ogni proteina viene descritta tramite tre file: un file in formato PDB che contiene le proprietà strutturali della proteina; un file ottenuto dal dall'algoritmo di clustering (sez. 4.3.3) contenete gli soli atomi di superficie e relative proprietà, le regioni col dettaglio delle proprietà medie di superficie e gli archi che descrivono le posizioni relative di coppie di regioni come descritto in sez. 5.3.1; un ultimo file che contiene le informazioni su pattern e supporti della proteina ottenuti dall'algoritmo di mining (sez. 5.3.2).

Il programma PDBVision fornisce un modo per selezionare un insieme di proteine dall'archivio e di caricare tutti i dati ad essa legati.

8.2.2. Visualizzazione della proteina

L'applicazione fornisce una interfaccia per la visualizzazione di una molecola, ossia l'insieme degli atomi di superficie visualizzati come sfere. Il raggio di queste è pari al raggio atomico di ogni singolo atomo aumentato del raggio del solvente. Questa rappresentazione della superficie viene chiamata superficie accessibile al solvente ed è descritta in sez. 2.4. L'applicazione utilizza le funzionalità messe a

disposizione dalle librerie grafiche openGL per la gestione di materiali, luci, ombre e fornisce, così, una visualizzazione tridimensionale della molecola.

L'applicazione mette a disposizione dell'utente dei metodi per effettuare delle trasformazioni di rotazione e traslazione sulla molecola visualizzata. Una caratteristica molto importante ai fini di una buona analisi superficiale è quella di propagare le trasformazioni a più proteine nello stesso momento, viene infatti offerta la possibilità di collegare più viste aperte su diverse proteine per gestire la stessa trasformazione su tutte le visualizzazioni e facilitare l'utente nel confronto di superfici.

Sono disponibili tre livelli di analisi:

(1) Analisi a livello di atomo

(2) Analisi a livello di regione

(3) Analisi a livello di pattern complesso di superficie

A livello di atomo il visualizzatore mostra la proteina come un insieme di atomi considerando ognuno di essi come un'entità a se. A livello di regione il visualizzatore fornisce una rappresentazione distinta per ogni regione diversificandole tramite la colorazione con la stessa tinta di tutti gli atomi che ne fanno parte. Naturalmente le diverse regioni assumono tinte differenti. In questo caso l'applicazione tralascia il concetto di atomo, considerando la regione come entità a se. A livello di pattern, l'applicazione fornisce all'utente un'interfaccia adeguata per poter visualizzare i pattern proteici e i loro supporti, allo scopo viene aperta una finestra sulla proteina che presenta il pattern e una su ogni altra che compare nel suo supporto.

8.2.3. Selezione e visualizzazione delle proprietà

Sempre all'interno dell'architettura a tre livelli descritta precedentemente, il visualizzatore proteico permette la selezione e la visualizzazione delle proprietà relative agli atomi, alle regioni e ai pattern. Analizzando la proteina al primo livello, un atomo selezionato deve essere evidenziato in modo da distinguersi dagli altri, così come al secondo livello è possibile selezionare ed evidenziare una singola regione, naturalmente in entrambi i casi è supportata la selezione multipla. Ad ogni livello è comunque possibile visualizzare le proprietà associate all'atomo e alle sue

regioni tramite l'apertura di finestre di pop-up che riportano i dettagli degli oggetti selezionati.

Per le proprietà di curvatura, idrofobicità e di potenziale il tool fornisce una visualizzazione che associa ad ogni elemento un colore corrispondente al valore della proprietà selezionata. Questo deve essere effettuato a livello atomico e a livello di regione. In questo modo l'utente ha una visualizzazione globale della proteina a livello della proprietà selezionata e viene supportato graficamente nella comparazione tra proteine differenti. Inoltre l'applicazione rende disponibile la visualizzazione multipla della stessa proteina e, in questo modo, è possibile visualizzare più volte la superficie per analizzare contemporaneamente le diverse proprietà.

8.2.4. Visualizzazione di pattern e relativi supporti

Oltre alla visualizzazione di una proteina, il sistema è in grado di visualizzare i pattern complessi di superficie trovati all'interno del dataset. Per ogni pattern vengono visualizzate le regioni coinvolte e il suo supporto. La funzionalità principale di PdbVision consiste proprio nella visualizzazione del pattern complesso. All'apertura di questo vengono create due viste per ogni pattern, la prima mostra le regioni coinvolte, la seconda mostra il potenziale elettrostatico in quanto si ritiene che sia la proprietà di maggior rilievo tra tutte le altre presentate nelle sez. 2.5. In questo caso, il sistema deve permettere di visualizzare entrambe le proteine e di evidenziare le regioni che matchano tra loro, colorandole nel medesimo colore. Ora l'utente potrà allineare le due proteine tramite il colore delle regioni e bloccare gli assi in modo che le trasformazioni applicate ad una siano propagate anche all'altra molecola. In questo modo sarà possibile verificare come le regioni evidenziate in una proteina siano distribuite nello spazio in modo simile alle regioni dell'altra proteina e verificare che le regioni colorate in modo uguale abbiano proprietà simili.

8.3. Il tool PDBVision

Nei seguenti paragrafi verrà descritto come creare un progetto di analisi tramite il tool PDBVision. Il tool da infatti la possibilità di creare, salvare e caricare progetti per l'analisi dei pattern di superficie. Di seguito verranno descritti i passi necessari per la

8.3.1. Definire il set di proteine

La prima operazione da compiere per poter analizzare le proteine dal punto di vista della superficie è naturalmente la definizione del dataset da utilizzare per la ricerca dei pattern frequenti. L'applicazione si appoggia su un database di proteine definite tramite file in formato PDB, tale database deve essere specificato indicando la cartella contenente i file tra le opzioni dell'applicazione. In ogni momento è possibile modificare il path del database.

Quando il database è stato esplicitato allora è possibile scegliere un sottoinsieme di proteine

8.3.2. Visualizzare superficie e proprietà

Una volta settato il percorso si può utilizzare il comando *Apri* contenuto nel menu *File* e selezionare un file PDB. In alternativa è possibile utilizzare il Drag and Drop di Windows e trascinare uno o più file all'interno della finestra di visualizzazione. La proteina verrà visualizzata al centro della finestra e di default verranno visualizzati solamente gli atomi superficiali, in quanto sono gli atomi che interessa utilizzare al fine dell'analisi di superfici proteiche. In ogni caso è possibile visualizzare anche gli atomi interni utilizzando una voce contenuta nel menu *Visualizzazione*.

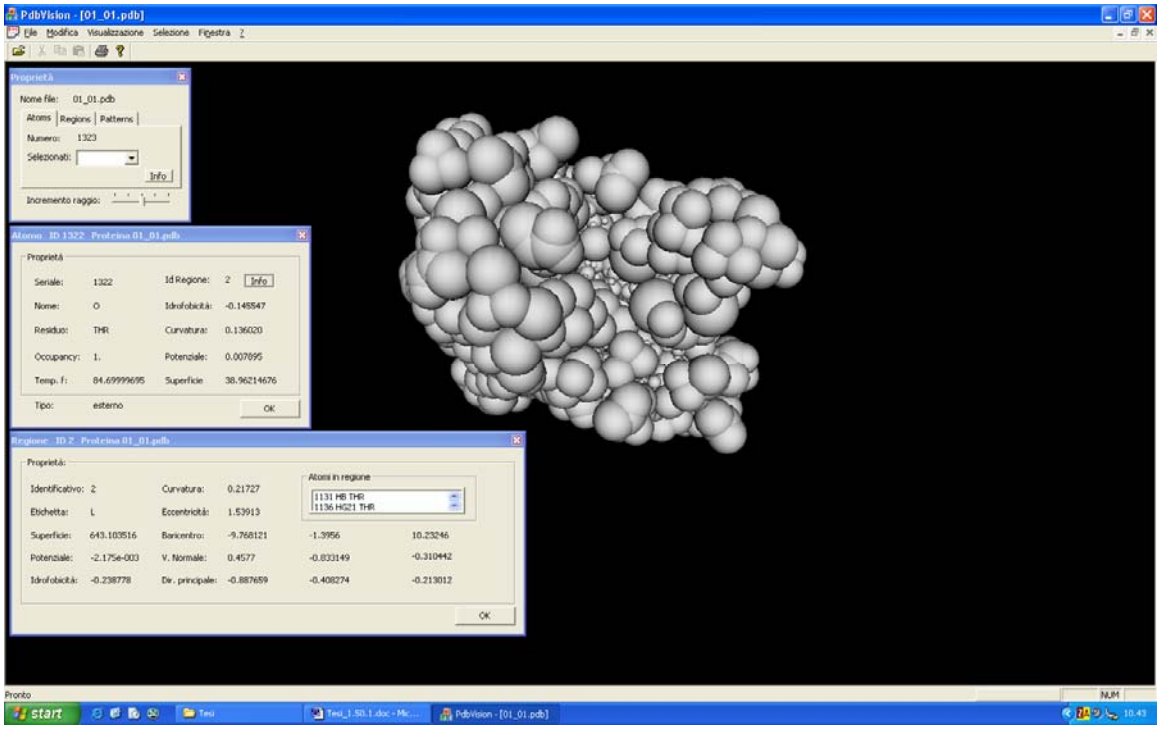

Fig. 8.4: Visualizzazione di una proteina

Attraverso il mouse è possibile ruotare la molecola. I movimenti sono fluidi e precisi.

La voce di menu *selezione* segue la logica a tre livelli con cui è stata strutturata l'applicazione. A seconda del livello selezionato, le operazioni che si eseguiranno verranno eseguite sul determinato livello:

- Atomo: gli atomi appariranno con il loro colore di default. Posizionandosi con il cursore sopra una sfera, nella barra di stato verranno visualizzate informazioni relative all'atomo puntato, che ancora non è in stato di selezione. Le proprietà riportate sono il numero identificativo, il nome dell'atomo e il tipo di aminoacido. Attraverso l'evento doppio clic è possibile selezionare l'atomo. Esso verrà colorato con il colore associato alla regione cui esso appartiene. Fanno eccezione gli atomi interni e alcuni atomi superficiali. Questi non sono associati a nessuna regione per cui verranno evidenziati tramite un colore di default. Per deselezionare l'atomo è sufficiente ripetere il doppio clic sullo stesso. Attraverso il menu contestuale, è possibile visualizzare la finestra di dialogo delle proprietà proteiche. Essa è composta da tre tabsheet che rispecchiano l'architettura a tre livelli. Di default la tabsheet visualizzata per prima è quella relativa agli atomi in cui viene riportato il numero di atomi costituente la proteina e la lista di atomi selezionati. Cliccando su info, verranno visualizzate le proprietà atomiche dell'atomo selezionato.
- Regione: ponendosi a livello di regione si nota che la proteina contiene delle porzioni colorate di nero. Posizionandosi su queste zone la barra di stato mostrerà un messaggio che informa l'utente che la zona selezionata non appartiene a nessuna regione. Questo dipende dall'algoritmo di clustering, il quale non assegna ad ogni atomo una regione di appartenenza. Posizionandosi con il cursore su una regione, la barra di stato ne visualizzerà l'identificativo. Facendo doppio clic su di essa, ogni atomo che la costituisce verrà colorato col medesimo colore in modo da evidenziare la regione selezionata. Attraverso la voce di menu *selezione* è possibile selezionare tutte le regioni con un clic oppure deselezionare tutte le regioni selezionate. Nella finestra vista in precedenza, all'interno della tabsheet relativa alle regioni vi sarà ora il numero di regioni costituenti la proteina e la lista delle regioni selezionate di cui è possibile cliccando sul tasto info visualizzarne le proprietà.
- Pattern: a questo livello posizionandosi con il mouse su una zona della proteina, la barra di stato visualizzerà la regione sotto il cursore. Utilizzando la finestra di dialogo, all'interno della tabsheet relativa ai pattern è possibile visualizzare il

numero di pattern frequenti archiviati per la proteina in analisi e la lista di questi. Selezionandone uno all'interno della lista, l'applicazione evidenzierà le regioni costituenti il pattern. Cliccando sul pulsante info saranno visualizzate le proprietà relative al pattern. Il passo successivo è quello di visualizzare il supporto del pattern, ovvero le proteine che condividono lo stesso pattern. Selezionando un supporto tra l'elenco dei supporti, l'applicazione caricherà la proteina adeguata e la visualizzerà in un'altra finestra. L'applicazione colorerà del medesimo colore le regioni con proprietà simili nelle due differenti proteine.

Cliccando con il pulsante destro del mouse apparirà un menu contestuale che permette di raggiungere varie funzionalità dell'applicazione. La prima voce a partire dall'alto è la voce *Zoom*. Attraverso il sottomenu, oppure utilizzando i tasti + e – della tastiera, è possibile fare traslare la proteina lungo l'asse Z avvicinandola od allontanandola dal punto in cui è fissata la telecamera virtuale (il punto di vista dell'utente). La seconda voce è denominata *Trasformazioni* ed è analoga alla voce *Trasformazioni* nel menu *Visualizzazione*. Attraverso di essa è possibile annullare le rotazioni e le traslazioni effettuate. Cliccando su *Posizione originale* si annullano con un solo clic le traslazioni lungo i 3 assi e le rotazioni. All'interno del menu contestuale è contenuto un sottomenu denominato *Visualizzazione* che permette una visualizzazione per proprietà della molecola. Le proprietà previste sono l'idrofobicità, la curvatura e il potenziale. La visualizzazione per proprietà si ha sia a livello atomico che a livello di regione. A livello atomico si avrà che ogni atomo sarà colorato con un colore corrispondente al valore della proprietà selezionata. A livello regionale ogni atomo sarà colorato e visualizzato tramite il colore assegnato alla proprietà relativa alla regione. Ogni proprietà possiede un proprio range di valori. Solitamente i valori possono essere sia negativi che positivi. I valori negativi vengono rappresentati tramite differenti tonalità di rosso mentre i valori positivi tramite differenti gradazioni di blu. In entrambi i casi si passa dal bianco alla tonalità più intensa del colore rosso o blu. Attraverso la voce *Proprietà* del menu Visualizzazione è possibile settare e modificare i limiti di ogni proprietà. Eventuali valori inferiori o uguali al limite inferiore verranno codificati tramite la tonalità più intensa di rosso. Analogamente valori maggiori o uguali al limite superiore verranno codificati tramite la tonalità più intensa di blu. Modificando questi valori è possibile scegliere la configurazione migliore al fine di visualizzare più correttamente e in modo più affinato le proprietà degli elementi costituenti la molecola. Questi valori

vengono memorizzati nel registro di sistema in modo da non dover ripristinare ogni volta la configurazione che si ritiene ideale.

Una particolarità di questo tool è la possibilità di visualizzare una molecola in modo differente allo stesso momento sfruttando l'architettura document/view. Attraverso il comando *Nuova finestra* contenuto nel menu *Finestra* si aprirà una nuova visualizzazione della stessa molecola. Ad esempio nella fig 8.7 è mostrata la stessa proteina visualizzata in 4 maniere differenti. Le due finestre a sinistra sono una visualizzazione a livello atomico delle proprietà curvatura e potenziale. Le due finestre a destra rappresentano la stessa molecola a livello regionale delle medesime proprietà.

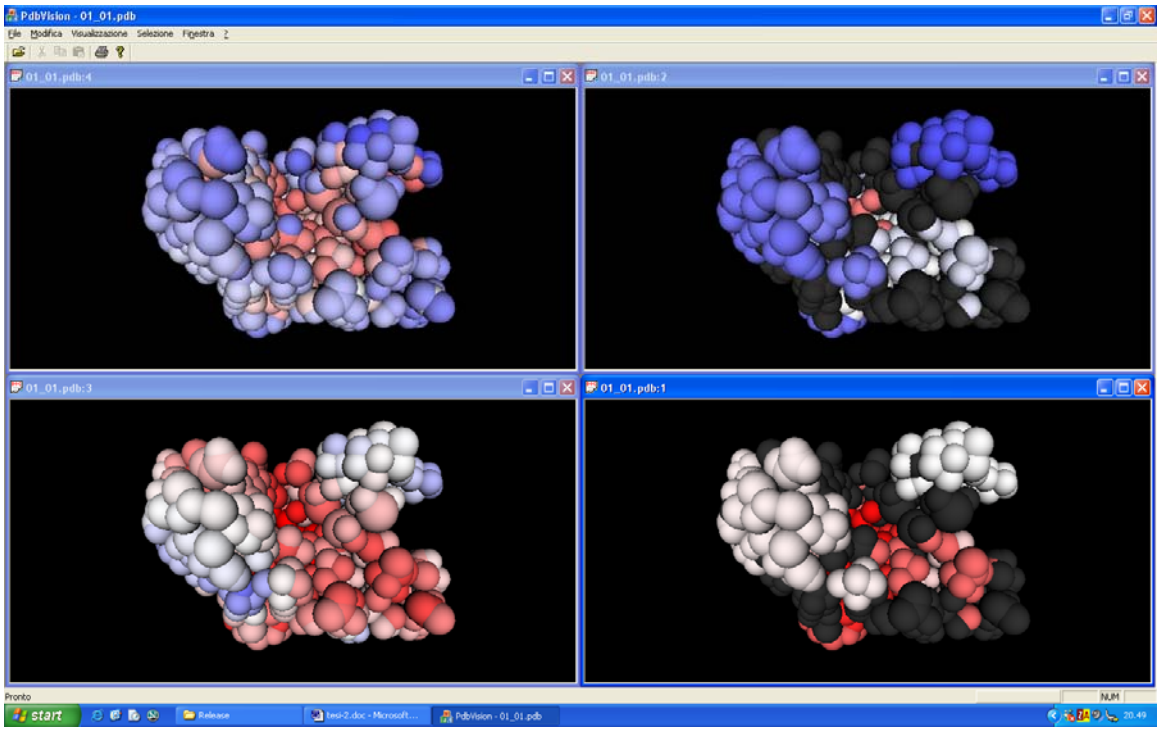

Fig. 8.5: Visualizzazione per proprietà

Posizionandosi col cursore sopra un atomo, oltre alle informazioni gia citate, verrà visualizzato il valore della proprietà assegnatagli in relazione al tipo di visualizzazione selezionato: ad esempio il valore del potenziale dell'atomo o della regione selezionati.

Posizionandosi sulla molecola e cliccando su Info atomo, sempre all'interno del menu contestuale, verrà aperta una finestra di dialogo in cui verranno riassunte le proprietà appartenenti all'atomo selezionato.e la tipologia dello stesso (interno o superficiale). La stessa funzionalità è disponibile a livello di regione. La dialog box relativa all'atomo ha un collegamento alla regione di appartenenza, mentre la dialog

box relativa alla regione contiene un collegamento agli atomi appartenenti ad essa implementato tramite una lista. In questo modo la navigazione in entrambi i sensi risulta veloce e intuitiva. L'ultima voce che conclude il menu contestuale è denominata Proprietà. Cliccando sul comando apparirà la una finestra tramite la quale è possibile incrementare o decrementare il raggio atomico. Ciò permette di addentrarsi all'interno della molecola alla ricerca di atomi interni. Inoltre sarà visibile una Tabsheet contenente le voci Atoms, Regions e Patterns gia citata in precedenza all'interno della definizione di architettura a tre livelli in questo stesso paragrafo. Infine all'interno del menu di Visualizzazione sono presenti due voci che permettono di personalizzare l'aspetto globale dell'applicazione. La prima voce riguarda il colore di sfondo. È possibile scegliere tra quattro colori di base quali il bianco, il grigio chiaro, il grigio ed il nero. La voce Definizione permette invece di aumentare o diminuire la definizione delle sfere rappresentanti l'atomo. Ogni sfera è rappresentata come un mappamondo tramite meridiani e paralleli che fungono da linee di controllo. Queste determinano la superficie sferica attraverso algoritmi di interpolazione. All'aumentare delle linee aumenta la precisione con cui viene disegnata la sfera. Ovviamente si ha un maggior numero di calcoli da effettuare per visualizzare in questo modo. Essendo questo numero ripetuto per ogni atomo presente nella molecola, ne risulta che ogni trasformazione è il risultato di un elevato numero di calcoli e l'aggiornamento della posizione viene rallentato drasticamente. Se non si dispone di una macchina sufficientemente potente, si consiglia di mantenere livelli bassi di definizione. Sia il settaggio dello sfondo che per il settaggio della definizione viene utilizzato il registro di sistema al fine di ricordare le ultime impostazioni utilizzate dall'utente.

9. Conclusioni

In questo lavoro di tesi è stato presentato un approccio originale per analizzare la superficie delle proteine basato sull'identificazione di regioni omogenee biologicamente rilevanti, sul mining di pattern di superficie complessi e sulla definizione di una funzione di similarità basata su tali pattern. L'originalità dell'approccio si basa sul considerare l'intera superficie proteica e ricercare similarità senza sfruttare conoscenze a priori di funzioni o di vincoli strutturali del pattern.

La complessità del problema nasce dall'eterogeneità delle superfici proteiche, dalla variabilità e instabilità delle proprietà chimiche e di forma che rende difficile lavorare con dataset reali. Il framework e gli algoritmi sviluppati, oggetto di pubblicazioni in riviste e conferenze internazionali, hanno tuttavia confermato la possibilità di estrarre dalla superficie proteica informazioni utili alla classificazione funzionale delle proteine.

L'efficacia è stata provata dai test sperimentali effettuati sia su dataset sintetici sia su insiemi di proteine reali. In particolare, le similarità identificate su un dataset composto da 25 proteine reali, sono state validate da esperti del dominio applicativo che hanno riconosciuto, nei pattern condivisi tra le diverse proteine, aree effettivamente interessanti dal punto di vista chimico-molecolare.

Il lavoro di ricerca presentato non si esaurisce col termine del corso di dottorato. I principali obiettivi per gli anni futuri possono essere elencati come segue:

- Dai test effettuati sui dataset si è stato evidenziato una instabilità residua del clustering. Sarebbe quindi utile studiare un nuovo algoritmo in grado di ovviare questo problema.
- Per aumentare l'utilità del metodo, dal punto di vista biologico, bisogna applicarlo ad un insieme più ampio di superfici di proteine reali. Sarebbe necessario studiare strutture dati accessorie allo scopo di ridurre i tempi di calcolo su grandi insiemi di proteine. Da notare che, mentre il dataset dei mutanti era particolarmente difficile da classificare per le forti similitudini tra le proteine generate e i loro semi, l'applicazione del nostro approccio a proteine reali, e quindi meno correlate, ridurrebbe l'impatto del rumore introdotto da regioni molto simili.

- Una parte del progetto, attualmente in fase di avanzamento, ha come obiettivo lo studio di come le proprietà funzionali possono essere derivate da una classificazione basata sulle sole caratteristiche di superficie, ciò porterebbe ad ottenere una classificazione complementare o alternativa a quelle esistenti.
- Infine sarebbe possibile accostare all'approccio la molecular dynamics allo scopo di migliorare la ricerca di pattern simili.

Bibliografia

[ADA91] Adams M. D. et al., "**Complementary DNA sequencing: expressed sequence tags and human genome project**", Science, 252: 1651 - 1656, 1991

[ALT90] Altschul S. F. et al., "**Basic local alignment tool**", Journal of molecular biology, 215: 403 – 410, 1990

[ALT97] Altschul S.F. et al., Gapped BLAST and PSI-BLAST: "**a new generation of protein database search programs**", Nucleic acid research, 25[17]: 3389 - 3402, 1997

[ALT98] Altman R. B., "**Bioinformatics in support of molecular medicine**", Proceedings of AMIA Symposium, 1998

[AND97] Andrade M. A., Sander C., "**Bioinformatics: from genome data to biological knowledge**",Current opinion in biotechnology, 8: 675–683, 1997

[APW01] Apweiler R. et al., "**The InterPro database, an integrated documentation resource for protein families, domains and functional sites"**, Nucleic acid research, 29[1]: 37 – 40, 2001

[ASH97] Ashburner M., Goodman N., "**Informatics – genome and genetic databases**", Current opinion in genetics and development, 7: 750 – 756, 1997

[BAK98] Baker P. G., Brass A., "**Recent developments in biological sequence databases**", Current opinion in biotechnology, 9: 54 - 58, 1998

[BEN96] Benton D., "**Bioinformatics – principles and potential of a new multidisciplinary tool**", Trends in biotechnology, 14: 261 – 272, 1996

[BIT99] Bittner M., Meltzer P., Trent J., "**Data analysis and integration: of steps and arrows**", Nature genetics, 22: 213 – 215, 1999

[BOG94] Bogusky M. S., "**Bioinformatics**", Current opinion in genetics and development, 4: 383 – 388, 1994

[BOG95] Boguski M. S., Schuler G. D., "**Establishing a human transcript map**", Nature genetics, 10: 369 - 371, 1995

[BOU00] Bourne P. E., "**Bioinformatics meets data mining: time to dance?**", Trends in biotechnology, 18: 228 – 230, 2000

[BUI01] Marcello Buratti, **Le biotecnologie, l'ingegneria genetica fra bilogia, etica e mercato.** Il Mulino 2001

[CKC99] Craven M., Kumlien J., "**Constructing biological knowledge bases by extracting information from text sources**", Proceedings of the 7th international conference on intelligent systems for molecular biology (ISMB), 1999

[DEL98] Deloukas P. et al., "**A physical map of 30.000 human genes**", Science, 282: 744 - 746, 1998

[EDD96] Eddy S. R., "**Hidden Markov models**", Current opinion in structural biology, 6: 361 – 365, 1996

[EDD98] Eddy S. R., "**Profile hidden Markov models**", Bioinformatics, 755 – 763, 1998

[EIS00] Eisenberg D. et al., "**Protein function in the post-genomic era**", Nature, 405: 823 – 826, 2000

[EMM00] Emmett A., "**The Human Genome**", The scientist: 14[15]: 1, 2000

[FIS96] Fischman J., "**Working the Web with a virtual lab and some Java**", Science, 273: 591 - 593, 1996

[GER97] Gershon D., "**Bioinformatics in a post-genomics age**", Nature, 389: 417 - 418, 1997

[GIB96] Gibrat J., Madej T., Bryant S. H., "**Surprising similarities in structure comparison**", Current opinion in structural biology, 6: 377 – 385, 1996

[GRB99] Grundy W. N., Bailey T., "**Family pairwise search with embedded motif models**", Bioinformatics, 15[6]: 463 – 470, 1999

[GRE00] Greene E. A., Henikoff S., "**Getting more from your sequence on the web**", Nature genetics, 2000

[HAG98] Hager C. et al., "**The Genome Sequence database (GSDB): improving data quality and data access**", Nucleic acid research, 26[1]: 21 - 26, 1998

[HAU98] Haussler D., "**Computational gene-finding**", Trends in biochemical sciences – supplementary guide, 12 – 15, 1998

[HOF99] Hofmann K., Bucher P., Falquet L., Bairoch A., "**The PROSITE database, its status in 1999**", Nucleic acid research, 27[1]: 215 - 219, 1999

[HOL93] Holm, L. and C. Sander. (1993) "**Protein structure comparison by alignment of distance matrices",** J. Mol. Biol. (1993)233:123-138).

[GRU00] Reed J., "**Trends in commercial bioinformatics**", Oscar Gruss, 2000

[GSU98] Guffanti A., Simon G., "**UniBLAST and the EST extractor: new WWW resources for EST data mining**", Trends in genetics, 14[7]: 293, 1998

[HEN92] Henikoff S., Henikoff J. G., "**Amino acid substitution matrices from protein blocks**", Proceedings of the National Academy of Sciences USA, 89: 10915 – 10919, 1992

[HOL94] Holm L., Sander C., "**Searching protein structure databases has come of age**, Proteins, 19: 165 - 173, 1994

[JRF00] Jurisica I., Rigoutsos I., Floratos A., "**Knowledge discovery in biological domains**", Sixth ACM SIGKDD international conference on Knowledge discovery an data mining, Boston, 2000

[JAN99] Jang W., Chen H., Sicotte H., Schuler G. D., "**Making effective use of human genomic sequence data**", Trends in genetics, 15[7], 1999

[KAH95] Kahn P., "**From genome to proteome: looking at cell's proteins**", Science: 270, 1995

[KAR93] Karlin S., Altschul S. F., "**Applications and statistics for multiple high-scoring segments in molecular sequences**", Proceedings of the National Academy of Sciences USA, 90: 5873 – 5877, 1993

[KBH99] Karplus K., Barrett C., Hugney R., "**Hidden Markov models for detecting remote protein homologies, Bioinformatics**", 1999

[KEL99] Kelley J. M. et al., "**High throughput direct end sequencing of BAC clones**", Nucleic acid research, 27[6]: 1539 – 1546, 1999

[KOO98] Koonin E. V., Tatusov R. L., Galperin M. Y., "**Beyond complete genomes: from sequence to structure and function**", Current opinion in structural biology, 8: 355 – 363, 1998

[KRE00] Kreeger K.Y., "**Retooling for bioinformatics"**, The Scientist: 14[23] : 35, 2000

[KUE99] Kuel P. M. et al., "**An effective approach for analyzing "prefinished" genomic sequence data**", Genome research, 9: 189 – 194, 1999

[LAB01] Labrador M et al., "**Molecular biology: protein encoding by both DNA strands**", Nature, 409: 1000, 2001

[LOC00] Lockhart D., Winzeler E. A., "**Genomics, gene expression and DNA arrays**", Nature, 405: 827 – 835, 2000

[MAR98] Marra M. A., Hillier L., Waterston R. H., "**Expressed sequence tags– Establishing bridges between genomes**", Trends in genetics, 14[1]: 4 - 7, 1998

[MAR99] Marshall E.,"**Do-it yourself gene watching**", Science, 286: 444 - 447, 1999

[NAT01] Nature – numero speciale sul genoma umano, 409, 2001

[NEW98] Newell W., Beck S., Lehrach H., Lyatt A., **Estimation of distances and map construction using radiation hybrids**", Genome research, 8: 493 – 508, 1998

[NOW95] Nowak R., "**Entering the postgenome era**", Science, 270:368 – 371, 1995

[NUC00] Nucleic acid research – Database issue, 28[1], 2000

[OLS95] Olson M. V., "**A time to sequence**", Science, 270: 394 - 396, 1995

[PAL99] Pandey A., Lewitter F., "**Nucleotide sequence databases: a gold mine for biologists**", Trend in biochemical sciences, 24: 276 - 280, 1999

[PAN00] Pandey A., Mann M., "**Proteomics to study genes and genomes**", Nature, 405: 837 – 845, 2000

[PAR98] Park J. et al., "**Sequence comparison using multiple sequences detect three times as many remote homologues as pairwise methods**", Journal of molecular biology, 284[4]: 1201 – 1210,1998

[PEA90] Pearson W. R., "**Rapid and sensitive sequence comparison with FastP and FastA**", Methods in Enzymology, 183: 63 – 98, 1990

[PEN99] Pennisi E., "**Seeking common language in a tower of Babel**", Science, 286: 449, 1999

[RAS97] Rastan S., Beeley L. J., "**Functional genomics: going forwards from databases**", Current opinion in genetics and development, 7:777–783, 1997

[REE00] Reed J., "**Trends in commercial bioinformatics**", Oscar Gruss, 2000

[ROS00] Roses A. D., "**Pharmacogenomics and the practice of medicine**", Nature, 405: 857 – 865, 2000

[SAN00] Sander C., "**The GeneQuiz web server: protein functional analysis through the Web**", Trends in biochemical sciences, 25: 33 - 35, 2000

[SAV00] Saviotti P. P. et al., "**The changing marketplace of bioinformatics**", Nature biotechnology, 18: 1247 – 1249, 2000

[SCH96] Schuler G. D. et al., "**A gene map of the human genome**", Science, 274: 540 - 546, 1996

[SCH97] Schuler G. D., "**Sequence mapping by electronic PCR**", Genome research, 7: 541 – 550, 1997

[SCH98] Schuler G. D., "**Electronic PCR: bridging the gap between genome mapping and genome sequencing**", Trends in biotechnology, 16: 456 - 459, 1998

[SCI01] Science – numero speciale sul genoma umano, 2001

[SEI96] Seife C., "**Do Java users live dangerously?**", Science, 273: 592, 1996

[SIE99] Siegel A. F. et al., "**Analysis of sequence-tagged-connector strategies for DNA sequencing**", Genome research, 9: 297 – 307, 1999

[SKU99] Skupski M. P., "**The Genome Sequence database: towards an integrated functional genomics resource**", Nucleic acid research, 27[1]: 35 - 38, 1999

[SMI98] Smith T. F., "**Functional genomics – bioinformatics is ready for challenge**", Trends in genetics, 14[7]: 291 - 293, 1998

[SMI00] Smith C. M., "**Bioinformatics, genomics and proteomics**", The scientist: 14[23]:26, 2000

[SOB97] Sobral B. W. S., "**Common language of bioinformatics**", Nature, 389: 418, 1997

[SPE00] Spengler S. J., "**Bioinformatics in the information age**", Science: 287: 1221 - 1223, 2000

[STA91] States D. J., Gish W., Altschul S. F., **"Improved sensitivity of nucleic acid database searches using application specific scoring scheme**", METHODS, 3[1]: 66 – 70, 1991

[STE97] Stewart E. A. et al., "**An STS-based radiation hybrid map of the human genome**", Genome research, 7: 422 – 433, 1997

[STO01] Stoesser G. et al., "**The EMBL nucleotide sequence database**", Nucleic acid research, 29[1]: 17 – 21, 2001

[TAU96] Taubes G., "**Software matchmakers help make sense of sequences**", Science, 273: 588 – 590, 1996

[WAN00] Wang Y., Geer L. Y., Chappey C., Kans J. A., Bryant S. H., "**Cn3D: sequence and structure views for Entrez"**, Trends in biochemical sciences, 25: 300 - 302, 2000

[WHE01] Wheeler D. L. et al., "**Database resources of the National Center for Biotechnology Information**", Nucleic acid research, 29[1]: 11 - 16, 2001

[WIS00] Wise M. J., "**Protein annotators' assistant**", Trends in biochemical sciences, 25: 252 - 253, 2000

[ZHA98] Zhang Z. et al., "**Protein sequence similarity searches using patterns as seeds**", Nucleic acid research, 26[17]: 3986 - 3990, 1998

WEB

[w1] Primer in molecular Genetics, DOE, 1992
http://www.ornl.gov/hgmis/publicat/primer/intro.html

[w2] Bio Informatics Technology & Systems (BITS)
http://www.bitsjournal.com/

[w4] Molecular & cellular proteomics
http://www.mcponline.org

[w5] E. Russo High-throughput techniques will likely change the field of structural biology The Scientist 14[3]:1, Feb. 7, 2000
http://www.the-scientist.com/yr2000/feb/russo_p1_000207.html

[w6] SwissProt
http://www.expasy.ch/sprot/sprot-top.html

[w7] Unigene resources, NCBI, 1997
http://www.ncbi.nlm.nih.gov/UniGene/index.html

[w8] UniGene build procedure
http://www.ncbi.nlm.nih.gov/UniGene/build.html

[w13] Unigene Collection, NCBI news, 1996
http://www.ncbi.nlm.nih.gov/Web/Newsltr/aug96.html

[w14] Markovitz V., Heterogeneous molecular biology database systems, 1995
http://gizmo.lbl.gov/HDBMS/BCK/BCK.html

[w15] EMBL
http://www.ebi.ac.uk/embl/

[w16] Celera Genomics
http://www.celera.com

[w17] Genbank feature table definition: the /db_xref qualifier, 1998

http://www.ncbi.nlm.nih.gov/collab/db_xref.html

[w20] Henikoff S., Database searching with multiple alignments, 1998
http://www.blocks.fhcrc.org/~steveh/csh-lectures.html

[w23] Client needs and strategies, Base4 Inc. white paper, 1998
http://www.base4.com/needs_and_strategies.html

[w24] Hibbard J., Research gains from IT boom, Information Week, 1998
http://www.informationweek.com/700/500pharm.html

[w26] What is bioinformatics
http://www.biology.gatech.edu/bioinformatics/whatis.html

[w27] Moxon B., Mining gene expression databases, 1997
http://www.compaq.com/solutions/onlinelab/whitepaper/wpmged.html

[w28] Graber J., Mohr S., Some examples of bioinformatic applications
http://matrix.bu.edu/BF527/Examples.html

[w30] Richardson A., Information technology trends in clinical science, Compaq white paper
http://www.compaq.com/solutions/onlinelab/whitepaper/guest04.html

[w32] Uberbacher E., Computing the genome, ORNL Review, 1998
http://www.ornl.gov/ORNLReview/v30n3-4/genome.html

[w33] Genbank feature table definition, 2000
http://www.ncbi.nlm.nih.gov/collab/FT/index.html

[w34] Unigene Collection, NCBI news, 1996
http://www.ncbi.nlm.nih.gov/Web/Newsltr/aug96.html

[w38] Rebhan M., Knowledge discovery, 1997
http://bioinformatics.weizmann.ac.il/cards/knowledge.html

[w40] Gene ontology consortium
http://genome-www.stanford.edu/GO
http://www.geneontology.org/

[w42] Cline M., Barrett C., Karplus K., Sam-T98
http://www.cse.ucsc.edu/research/compbio

[w43] SWISS-PROT, Bairoch e Apweiler, 2000
http://www.expasy.ch/sprot

[w44] The TrEMBL database
http://www.hgmp.mrc.ac.uk/Bioinformatics/Databases/trembl-help.html

[w45] NR, Non Redundant
http://www.ncbi.nlm.nih.gov/Genbank/index.html

[w46] Brookhaven Protein Databank (PDB)

http://www.rcsb.org/pdb/

[w57] EMBnet Homepage

http://www.uk.embnet.org/