# Metodi e strumenti per l'analisi della complementarietà di sezioni bidimensionali di subunità polipeptidiche in proteine multimeriche.

RELATORE: Prof. Ferrari Carlo

LAUREANDO: Thomas Premaor

A.A. 2010-2011

UNIVERSITÀ DEGLI STUDI DI PADOVA
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
TESI DI LAUREA

# Metodi e strumenti per l'analisi della complementarietà di sezioni bidimensionali di subunità polipeptidiche in proteine multimeriche.

RELATORE: Prof. Ferrari Carlo

LAUREANDO: *Thomas Premaor*

Padova, 01 Ottobre 2011

# Indice

## Sommario

*La tesi è strutturata in cinque capitoli: nel* ***capitolo 1*** *viene introdotto il problema del protein docking. Vengono descritti i possibili approcci al problema e fornita una panoramica sugli attuali algoritmi utilizzati. Infine vengono presentati gli obbiettivi del progetto. Il* ***capitolo 2*** *descrive le superfici molecolari delle proteine, le loro rappresentazioni e lo strumento software PyMol, utilizzato per generare la superficie di Connolly. Il* ***capitolo 3*** *espone nei dettagli i metodi utilizzati per la realizzazione delle sezioni bidimensionali della proteina e la rappresentazione in coordinate curvilinee delle stesse. Nel* ***capitolo 4*** *viene descritto il metodo e la metrica utilizzata per la ricerca della complementarietà tra due sezioni. Nel* ***capitolo 5*** *vengono presentati i risultati del metodo utilizzato. Infine l'elaborato termina con le conclusioni del lavoro eseguito, dando spazio alle considerazione sorte a posteriori.*

# Capitolo 1

# Introduzione

## 1.1 Protein docking

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. La bioinformatica strutturale è una sottodisciplina della proteomica che sfrutta 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. E' molto difficile prevedere il funzionamento di una proteina solo sulla base della struttura primaria, 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. Negli ultimi anni si sono sviluppate molte aree di ricerca della bioinformatica strutturale:

- Predizione Strutturale: ha come scopo la determinazione della struttura tridimensionale partendo dalla sequenza amminoacidica della catena polipeptidica.

- 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.

- Allineamento strutturale: ha come scopo la comparazione di due o più strutture polipeptidiche per cercare similitudini nella loro conformazione

spaziale.

- Protein Docking: consiste nel determinare la struttura di un complesso proteico formato da due o più proteine senza l'ausilio di misure sperimentali.

Il Protein Docking è uno dei problemi aperti del settore della bioinformatica strutturale, esso consiste nella determinazione della struttura tridimensionale di un composto proteico data la struttura dei peptidi che lo compongono, ovvero trovare il "migliore" matching tra due molecole: un recettore ed un ligando. Per convenzione con il termine recettore si indica la molecola di dimensione maggiore, rispettivamente, con il termine ligando quella di dimensione minore. Il docking molecolare si usa per predire la struttura di complessi intermolecolari che vanno a formarsi tra due o più molecole. Il problema del docking può essere definito nel modo seguente: date le coordinate atomiche di due molecole, predire le loro corrette associazioni di legame ovvero la loro corretta interazione. Ci sono tre ingredienti chiave nel definire una procedura di docking:

- la rappresentazione del sistema
- la ricerca nello spazio conformazionale
- valutazione potenziali soluzioni

I tre aspetti del docking sono mutualmente intercorrelati: la scelta della rappresentazione del sistema (superficie) decide il tipo di algoritmo di ricerca conformazionale e il modo per valutare le possibili soluzioni.

## 1.2 Approcci al protein docking

Nel corso degli ultimi anni si sono seguiti molti approcci al problema del docking molecolare. La scelta della rappresentazione del sistema, consente di classificare il problema del docking in:

- Docking rigido
- Docking flessibile

### Docking rigido

Nel docking rigido le due molecole vengono considerate come corpi rigidi. La molecola recettore si considera fissa sul piano tridimensionale, mentre vengono valutate tutte le possibili posizioni e orientazioni del ligando nello spazio. La procedura di ricerca deve tener conto di sei gradi di libertà: tre per le traslazioni e tre per le rotazioni. Il primo algoritmo computazionalmente efficiente, per determinare la complementarietà geometrica tra due proteine, per la risoluzione del problema del docking rigido è stato presentato da Katchalski-Katzir et all.[1]. Tale metodo consiste in una procedura automatica che proietta la molecola in una griglia 3D, eseguendo una distinzione tra atomi di superficie ed interni. Calcola, utilizzando la trasformata di fourier, una funzione di correlazione che valuta il grado di sovrapposizione e penetrazione molecolare relativo a tutte le possibili orientazioni della molecola ligando. Esegue una scansione della relativa orientazione delle molecole in tre dimensioni. In questo tipo di approccio non vengono considerate mutazioni della superficie molecolare, dovute a cambiamenti della struttura terziaria, che si possono verificare durante l'interazione tra due proteine.

### Docking flessibile

L'approccio basato sul docking flessibile cerca di imitare il processo fisico di docking proteico. Camacho [2] ipotizza che il processo di docking avvinene in due passaggi: un passo di "riconoscimento", nel quale due proteine si avvicinano e si orientano verso una interfaccia di docking presente sulle superfici molecolari, ed un passo di "attacco" nel quale si formano le interazioni ad alta affinità tramite modifiche conformazionali della catena polipeptidica e del backbone. Il docking flessibile perciò considera, durante il processo di docking tra due polipeptidi, sia la complementarietà geometrica della proteina che la flessibilità della struttura terziaria della molecola. L'introduzione di queste considerazioni comporta un aumento considerevole dei gradi di libertà nello spazio delle soluzioni del sistema, ed un relativo aumento della complessità computazionale nella risoluzione del problema del docking flessibile. Per questo motivo la ricerca delle soluzioni all'interno dello spazio del sistema, non viene eseguita in modo deterministico ma si affida ad algorimi di ricerca euristici quali: Monte Carlo minimization, algoritmi genetici e tabu-search. Dato l'elevato numero di gradi di libertà, attualmente si cerca di risolvere il problema del docking semi-flessibile, nel quale il recettore

viene considerato un corpo rigido mentre si considera la flessibilità solo del ligando. L'approccio al docking flessibile attualmente viene utilizzato in ambito farmaceutico, nel quale la molecola ligando ha dimensioni molto ridotte.

La valutazione delle corrette conformazioni del ligando sul recettore, e la ricerca delle interfacce di legame, per formare un complesso proteico vengono effettuate basandosi su due criteri principali:

- Shape complementarity (complementarietà di superficie)
- Physicochemical complementarity (complementarietà fisico-chimica)

**Shape Complementarity**

La strategia generale per la simulazione del docking e la ricerca delle interfacce di legame si basa sulla complementarietà geometrica tra le superfici delle proteine. Per migliorare la correttezza della predizione delle aree di docking, oltre alle caratteristiche geometriche della superficie, sono stati introdotti ulteriori modelli. Si può valutare l'effetto di solvatazione, si utilizza un modello semplificato per valutare le interazioni elettrostatiche, vengono valutate le caratteristiche dei residui esposti sulla superficie e l'idrofobicità degli stessi [3].

**Physicochemical complementarity**

I metodi basati sulla complementarietà fisico-chimica, utilizzano modelli che simulano la meccanica molecolare della proteina, replicando il processo di folding delle due proteine, oppure cercando di rappresentare attraverso funzioni matematiche il maggior numero di forze che interagiscono nel processo di docking. L'approccio knowledge-based è basato sull'omologia tra le strutture primarie delle proteine, data la struttura primaria si riconoscono delle similarità strutturali con altre proteine per identificare i siti di legame attivi. Esistono alcuni web-server dedicati all'identificazione di tali interfacce (Rate4Site, ConSurf, FindSite). Un approccio molto utilizzato si basa sulla definizione di modelli energetici per descrivere l'interazione tra proteine. I principali modelli sono AMBER,CHARMM,CVFF,COMPASS.

## 1.3 Stato dell'arte

Per valutare le attuali tecniche e i metodi utilizzati in ambito scientifico, per risolvere il problema del docking preoteico, in particolare il problema del docking proteina-proteina, si è deciso di analizzare i risultati degli esperimenti effettuati durante CAPRI. La competizione CAPRI [4] (Critical Assessment of Predicted Interactions) è un ottima occasione per valutare le capacità degli attuali algoritmi di docking, poichè è una serie continua di eventi, in cui ricercatori di tutto il mondo tentano di effettuare il docking sulle stesse proteine assegnate. I Round si svolgono circa ogni sei mesi. Ogni round contiene da uno a sei complessi bersaglio proteina-proteina le cui strutture vengono determinate sperimentalmente e vengono mantenute segrete. Ai gruppi di ricercatori vengono fornite solamente le strutture atomiche delle due proteine che formano il composto. Ogni gruppo di ricerca, una volta applicato l'algoritmo di docking, consegnano ai valutatori un massimo di 10 modelli di docking possibile. La valutazione della struttura del composto viene effettuata considerando la distanza tra gli atomi di carbonio delle due proteine sull'interfaccia di docking. Al termine di tutte le valutazioni viene organizzata una conferenza nella quale vengono presentati i risultati della competizione, i gruppi di ricerca presentano i propri algoritmi di docking e si confrontano i vari metodi utilizzati. L'ultimo meeting organizzato dalla comunità CAPRI si è svolto a Barcellona nel Dicembre 2009.
Valutando i risultati dell'ultima competizione CAPRI sono stati studiati gli algoritmi di docking più efficienti, ovvero quelli che hanno ricevuto una valutazione più elevata sui modelli proposti. Da una prima analisi dei vari algoritmi proposti, si nota una combinazione di vari metodi per la soluzione di problemi generici di docking proteina-proteina, utilizzando diversi approcci al problema, focalizzando l'attenzione sul docking flessibile o semi-flessibile. In particolare l'analisi dei risultati evidenzia come tutti gli algoritmi con alti livelli di successo, utilizzano un approccio multistage al docking [5]. Il primo passo consiste in una ricerca semplificata sullo spazio conformazionale o in una ricerca basata sul docking-rigido, il secondo passo consiste nell'individuazione di una o più regioni di interesse candidate a interfacce di docking, il terzo passaggio consiste nel raffinamento della struttura, che viene eseguito considerando la flessibilità della struttura proteica, ed il quarto e ultimo passo consiste nella selezione del modello finale.

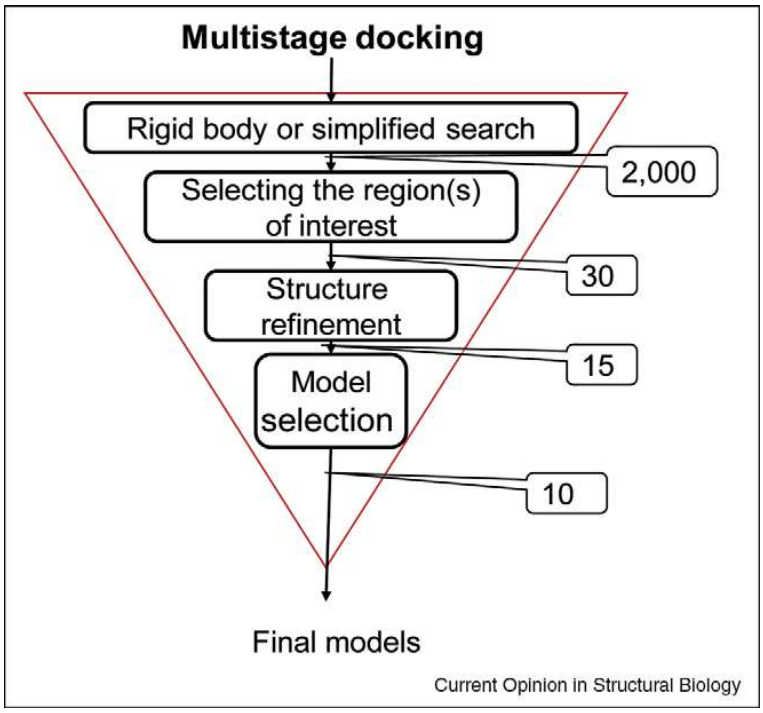


Figura 1.1: Approccio multistage al docking proteina-proteina; i riquadri a destra indicano il numero di modelli generati ad ogni passo

### Algoritmi di docking proteina-proteina

L'algoritmo CLUSPRO [6] è il più efficiente tra i metodi di predizione automatica del docking proteico proposti nell'ultimo CAPRI. CLUSPRO è un web server sviluppato dal laboratorio di bioinformatica strutturale della Boston University. L'approccio al docking proteina-proteina consiste in tre passi. Inizialmente viene eseguito PIPER [7], un algoritmo di docking rigido basato su correlazione tramite FFT, esteso valutando i potenziali di interazione elettrostatica. Successivamente le 1000 migliori conformazioni vengono raggruppate in cluster utilizzando la funzione RMSD come misura di distanza [8]. Nel terzo passo viena analizzata la "stabilità" del cluster [9], viene valutato ogni minimo locale di energia di interazione, eseguendo una simulazione Monte Carlo e scartando le conformazioni non native.

L'algoritmo HADDOCK [10, 11] è un software di docking sviluppato da Alexandre Bonvin, Utrecht University. Il protocollo di docking consiste in tre passi: una ricerca basata su docking rigido valutando la presenza di residui attivi o passivi sulla superficie molecolare, un raffinamento basato su docking semi-flessibile, durante il quale gli angoli torsionali e spaziali degli atomi della catena-peptidica e del backbone, relativi ai residui presenti sull'interfaccia, sono liberi di muoversi. Successivamente viene eseguito un raffinamento valutando l'energia di solvatazione nell'area di docking.

ZDOCK è una metodologia di docking proteina-proteina che utilizza una FFT (fast Fourier transform) per cercare tutte le possibili interfacce di legame delle subunità monometriche; l'algoritmo ricerca all'interno dell'intero spazio rotazionale e traslazionale di un monomero rispetto all'altro. Per ogni rotazione l'algoritmo sonda rapidamente lo spazio traslazionale usando la FFT. La valutazione di tutti i possibili complessi è basata principalmente sulla misura geometrica della complementarietà di superficie all'interfaccia proteica; in secondo luogo viene calcolata una energia libera di solvatazione, chiamata energia di contatto atomico, derivante da un test-set di complessi cristallografici ad alta risoluzione; infine viene calcolato un componente elettrostatico basato sulla formula di Coulomb e sul campo di forze CHARMM19.

GRAMM-X è un web server per il docking proteina-proteina, che utilizza la FTT e la complementarietà di superficie, la funzione di scoring energetica è

numericamente equivalente ad una energia interatomica basata sul potenziale di Lennard-Jones. Per eliminare le conformazioni non corrette viene applicato un filtro generato attraverso una Support Vector Machine.

PATCHDOCK è un algoritmo di docking ispirato a tecniche utilizzate in computer vision, infatti il docking viene comparato all'assemblamento del jigsaw puzzle. Il primo passo dell'algoritmo è l'individuazione delle patch geometriche presenti sulla superficie (knob, flat, hole) a seconda della loro curvatura. Successivamente vengono eseguiti due passaggi, Geometric Hashing e Pose Clustering, che permettono di associare le patch delle due proteine. La valutazione dell'accuratezza del match si basa sulla distanza tra le patch. Vengono scartate le conformazioni che presentano delle penetrazioni tra le patch ed infine le informazioni biologiche relative alla superficie vengono utilizzate per valutare il modello migliore.

## 1.4 Obbiettivo del progetto

L'obiettivo del progetto consiste nella realizzazione di un algoritmo di docking rigido proteina-proteina, basato sulla complementarietà di superficie. L'approccio seguito durante il progetto prevede il sezionamento della proteina in slice bidimensionali. Durante la fase di progettazione si è deciso di limitare l'applicazione del metodo esclusivamente a proteine che evidenziano un determinato asse di sviluppo. Tale asse di sviluppo coinciderà in seguito con l'asse di sezionamento della proteina. Il primo passaggio consiste nel rappresentare e analizzare le caratteristiche geometriche della superficie. Per analizzare la curvatura della superficie lungo il perimetro della sezione è stata introdotta una rappresentazione in coordinate curvilinee. Successivamente, con l'obiettivo di evidenziare l'andamento principale della curvatura lungo il perimetro, si è deciso di eseguire un filtraggio del perimetro (smoothing) eliminandone le rugosità.
Il primo obiettivo del progetto consiste nel valutare la complementarietà tra coppie di sezioni relative a recettore e ligando, per identificare l'orientazione di una sezione rispetto all'altra. Il confronto tra le sezioni si basa sulla rappresentazione in coordinata curvilinea.
L'algoritmo sviluppato deve perciò realizzare i seguenti punti:

- Generazione della superficie molecolare

- Sezionamento della superficie in slice bidimensionali.
- Filtraggio della superficie.
- Valutazione della complementarietà tra sezioni.
- Orientazione delle sezioni.

L'obiettivo sucessivo prevede di estendere i risultati della ricerca su coppie di sezioni a entrambe le proteine, valutando la complementarietà di un insieme di sezioni rispetto all'altro, identificando un interfaccia di docking e l'orientazione di una proteina sull'altra.

# Capitolo 2

# Superficie della proteina

La superficie molecolare è un elemento fondamentale nell'analisi del docking proteina-proteina, poichè rappresenta l'interfaccia di iterazione tra le due molecole. Nel presente capitolo vengono presentate le principali definizioni di superfice molecolare introdotte negli ultimi anni. Viene data una breve descrizioni dei metodi per il loro calcolo e rappresentazione. Infine viene presentato il software PyMol, utilizzato nel progetto per il calcolo e la visualizzazione della superficie proteica, il formato PDB e il formato VRML.

## 2.1 Superficie molecolare

Le molecole sono comunemente modellate attraverso una collezione di atomi rappresentati da sfere, con raggio uguale al loro raggio di Van der Waals. Tre tipi di superfici sono state definite basandosi su tale rappresentazione:

- Superficie di Van der Waals (VW Surface)
- Superficie accessibile al solvente (SAS Surface)
- Superficie esclusa al solvente (SES Surface)

#### Superficie di Van der Waals

La superficie di Van der Waals (VWS) è stata la prima definizione di superficie molecolare. La WDVS corrisponde all'unione della superficie esposta degli atomi rappresentati attraverso sfere di raggio uguale a quelli di Van der Waals **??**.

| Elemento | Raggio di Van der Waals (Å) |
|---|---|
| H | 1.2 |
| N | 1.5 |
| O | 1.4 |
| F | 1.35 |
| P | 1.9 |
| S | 1.85 |
| Cl | 1.8 |

Tabella 2.1: Raggi atomici di Van der Waals

### Superficie accessibile al solvente

Le proteine non esistono in modo isolato, ma si trovano comunemente nelle soluzioni (solvente), in particolare l'acqua. Poichè la superficie di Van der Waals contiene molti atomi e zone non accessibili al solvente, si è reso necessario definire una superficie molecolare che tenga in considerazione le caratteristiche del solvente nel quale si trova immersa la proteina. La superficie accessibile al solvente (SAS) è definita come il luogo del centro di una probe (sfera-sonda), rappresentante una molecola del solvente (ad esempio, una molecola d'acqua), che rotola lungo la superficie di Van der Waals di una proteina. Se le molecole d'acqua sono modellate come sfere di raggio 1,4 Å, allora la SAS di una data molecola può essere trovata aumentando il raggio di ogni atomo di 1,4 Å, e prendendo la superficie di Van der Waals del gruppo di atomi gonfiati. La superficie accessibile al solvente permette di valutare le proprietà di idropatia (idrofobicità e idrofilicità) degli amminoacidi presenti sulla superficie di una proteina, evidenziando le zone idrofobiche non racchiuse internamente alla struttura.

### Superficie esclusa al solvente

Mentre la VDWS contiene troppi atomi interni e patch che non sono accessibili al solvente, la SAS contiene aree che dovrebbero essere occupate dal solvente. Per superare questo inconveniente, Richards ha dato una definizione di superficie molecolare che consiste in un insieme di patch di contatto e patch rientranti. Una sfera-sonda rappresentante il solvente, viene fatta rotolare sugli gli atomi di una

proteina. La superficie risultante è: qella di Van der Waals, quando la probe e l'atomo sono a contatto (SCS Solvent Contact Surface) generando una patch di contatto, oppure quella del probe quando si trova a contatto con più di un atomo (RS Reentrant Surface) generando una patch rientrante. La superficie esclusa al solvente è anche conosciuta come superficie di Connolly (CS), o semplicemente superficie molecolare (MS).

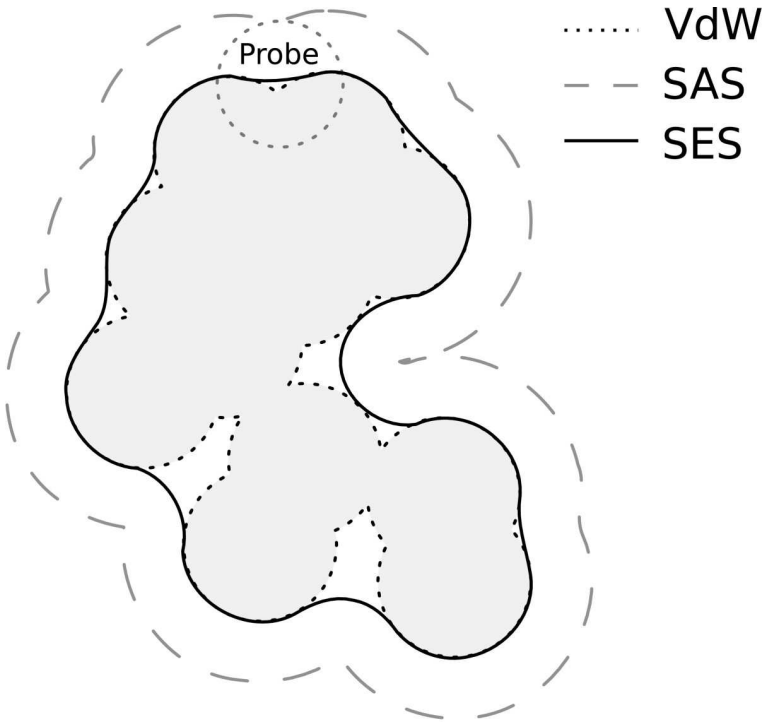


Figura 2.1: Superfici molecolari:Van der Waals (VdW), accessibile al solvente (SAS), esclusa al solvente (SES)

## 2.2 Calcolo della superficie molecolare

Connolly et al. definiscono 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 della superficie, la parte di superficie di Van der Waals toccata dalla sfera quando si trova in questo stato rappresenta la patch convessa di superficie a contatto con il 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.

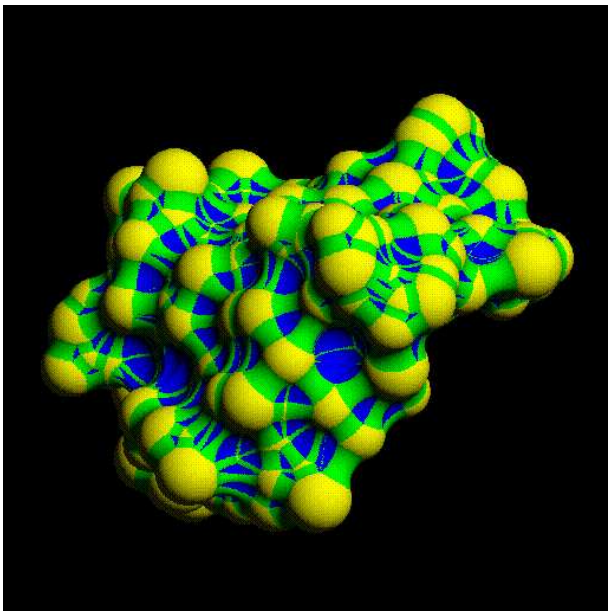

Figura 2.2: Superficie di Connolly. Sono evidenziate le varie patch che formano la superficie: patch convessa (giallo), patch a sella (verde) e patch concava (blu).

## 2.3 Rappresentazione della superficie

Esiste una netta distinzione tra la rappresentazione della struttura molecolare e della sua superficie. La struttura molecolare è generalmente rappresentata: atomo per atomo (CPK-wiew) per evidenziare il volume occupato dalla proteina e le caratteristiche chimiche dei singoli atomi, come una serie di elementi strutturali tramite diagramma di Richardson (ribbon diagrams), consentendo di rap-

presentare la struttura bel backbone definita dai legami covalenti, o cartoon per evidenziare i domini di struttura secondaria presenti nella proteina. Al contrario, la superficie molecolare, definita come l'insieme di punti che sono "accessibili" ad un dato solvente, è una più complessa struttura tridimensionale ed è generalmente rappresentata attraverso una mesh tridimensionale.

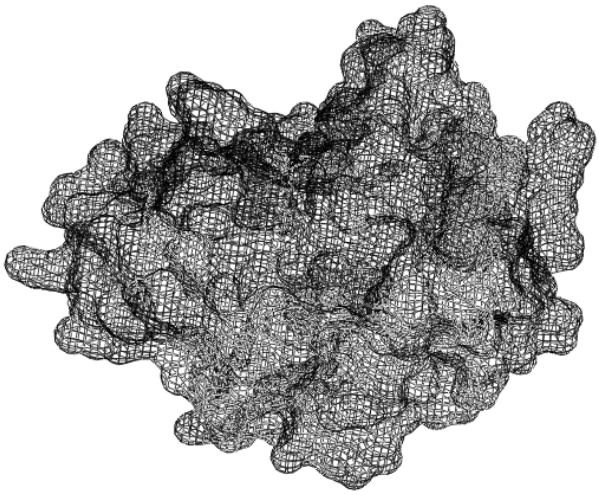

Figura 2.3: Superficie di Connolly rappresentata da una mesh triangolare

Una mesh triangolare è una collezione di vertici, spigoli e facce che definiscono la forma di un oggetto poliedrico nella computer grafica 3D e nella modellazione solida. Le facce consistono di triangoli e ciò semplifica molto il rendering. Le mesh sono primitive grafiche che consentono di risolvere con grande efficienza i procedimenti di visualizzazione delle forme modellate: sono strisce di triangoli con cui approssimiamo superfici curve.

## 2.4 PyMol

PyMol è un software libero (opensource) di grafica 3D utilizzato per la rappresentazione di biomolecole in biochimica, bioinformatica, e nella biologia strutturale. Formati di dati comuni consentono a PyMOL di girare su sistemi operativi diversi (multipiattaforma), con un rendering della grafica di alta qualità. La parte iniziale del nome, Py, si riferisce al linguaggio di programmazione Python utilizzato dal programma. Nella realizzazione del progetto, il software PyMol è stato utilizzato per il calcolo e la visualizzazione della superficie molecolare della proteina. PyMol consente di generare la superficie di Connolly date le coordinate atomiche della proteina, contenute nel file in formato PDB (.pdb). La superficie di connolly viene calcolata da PyMol utilizzando una probe di raggio 1,4 Å. La rappresentazione (rendering), della superficie calcolata, consiste in una mesh triangolare. La mesh triangolare generata viene successivamente salvata in formato VRML 2.0 (.wrl).

## 2.5 Formato PDB

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. Gli esperimenti che permettono di risolvere la struttura proteica sono essenzialmente la diffrazione a raggi X di una proteina cristallizzata e la risonanza magnetica nucleare di (piccole) proteine in soluzione. Il formato maggiormente utilizzato nell'analisi strutturale proteica è il Brookhaven Protein Databank, detto 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 (http://www.pdb.org). 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 PDB 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. 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 04-10-2011 conta 76.288 proteine.
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. 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.

```
SEQRES   1 A   34  ACE CYS GLY GLY VAL GLN ALA GLU GLU GLN LYS LEU ILE
SEQRES   2 A   34  SER GLU GLU ASP LEU LEU ARG LYS ARG ARG GLU GLN LEU
SEQRES   3 A   34  LYS HIS LYS LEU GLU GLN LEU NH2
SEQRES   1 B   34  ACE CYS GLY GLY MET ARG ARG LYS ASN ASP THR HIS GLN
SEQRES   2 B   34  GLN ASP ILE ASP ASP LEU LYS ARG GLN ASN ALA LEU LEU
SEQRES   3 B   34  GLU GLN GLN VAL ARG ALA LEU NH2
CRYST1    1.000    1.000    1.000  90.00  90.00  90.00 P 1           1
ORIGX1      1.000000  0.000000  0.000000        0.00000
ORIGX2      0.000000  1.000000  0.000000        0.00000
ORIGX3      0.000000  0.000000  1.000000        0.00000
SCALE1      1.000000  0.000000  0.000000        0.00000
SCALE2      0.000000  1.000000  0.000000        0.00000
SCALE3      0.000000  0.000000  1.000000        0.00000
ATOM      1  N   CYS A   3      38.240   2.099  -5.150  1.00  5.05           N
ATOM      2  CA  CYS A   3      38.673   3.471  -5.402  1.00  4.41           C
ATOM      3  C   CYS A   3      39.052   3.647  -6.869  1.00  4.23           C
ATOM      4  O   CYS A   3      39.919   2.945  -7.390  1.00  4.59           O
ATOM      5  CB  CYS A   3      39.880   3.821  -4.530  1.00  4.15           C
ATOM      6  SG  CYS A   3      39.450   3.592  -2.787  1.00  4.20           S
ATOM      7  H   CYS A   3      38.846   1.478  -4.695  1.00  5.43           H
ATOM      8  HA  CYS A   3      37.863   4.144  -5.165  1.00  4.68           H
ATOM      9  HB2 CYS A   3      40.707   3.175  -4.784  1.00  4.37           H
ATOM     10  HB3 CYS A   3      40.160   4.850  -4.700  1.00  4.24           H
ATOM     11  N   GLY A   4      38.392   4.595  -7.528  1.00  4.02           N
ATOM     12  CA  GLY A   4      38.666   4.861  -8.937  1.00  4.00           C
ATOM     13  C   GLY A   4      39.468   6.147  -9.099  1.00  3.54           C
ATOM     14  O   GLY A   4      40.471   6.185  -9.812  1.00  3.77           O
```

Figura 2.4: Esempio formato file PDB

## 2.6 Formato VRML2.0

VRML è l'acronimo di "Virtual Reality Modeling Language", si tratta di un formato di file per descrivere oggetti 3D interattivi. Tutte le entità in un mondo vrml sono rappresentate da dei nodi, il nodo Shape serve per rappresentare un oggetto. Il nodo Shape contiene al suo interno due fields (campi) che sono il field appearance e il field geometry: il field appearance definisce le caratteristiche fisiche dell'oggetto, mentre il field geometry ne definisce la forma. Il nodo IndexedFaceSet, all'interno del field geometry, rappresenta una forma tridimensionale costituita da un insieme di facce (poligoni), secondo la struttura face-vertex mesh. Il nodo IndexedFaceSet contiene due fields: il campo coord e il campo coordIndex. Il campo coord specifica i punti che costituiscono i vertici dei poligoni che formano l'oggetto. In particolare il campo coord contiene il nodo Coordinate il cui field point contiene i punti sottoforma di coordinate tridimensionali (x,y,z). Per rappresentare una faccia si indicano i vertici che la compongono. Questo viene fatto nel field coordIndex. In questo campo sono indicati i numeri dei vertici che compongono la faccia. Per indicare che non vi sono altri vertici che compongono una faccia viene riportato il numero -1. Il nodo Normal specifica, attraverso il campo vector, il vettore normale alla superficie per ogni vertice dei poligoni che formano l'oggetto. Attraverso il nodo IndexedFaceSet è possibile dunque rappresentare una mesh tridimensionale in formato VRML. In particolare, per una mesh triangolare, il nodo Coordinate rappresenta tutti i vertici della mesh, il field coordinateIndex è formato da una quaterna di indici (v1, v2, v3, -1) e il nodo Normal associa ad ogni vertice il relativo vettore normale alla superficie. La Figura 2.5 descrive la rappresentazione di una mesh triangolare in formato VRML2.0.

```
#VRML V2.0 utf8
Shape {
 appearance Appearance {
  material Material { diffuseColor 1.0 1.0 1.0 }
 }
 geometry IndexedFaceSet {
  coord Coordinate {
   point [
3.646517 -11.742987 -2.687943,
4.452619 -11.851198 -2.418610,
4.385262 -12.182285 -2.787567,
.... ]
        }
  coordIndex [
0 1 2 -1,
3 4 5 -1,
... ]
 normal Normal {
   vector [
0.4230 0.7404 -0.5224,
0.2648 0.8114 -0.5210,
0.3976 0.5933 -0.7000,
... ]
        }
  normalIndex [
0 1 2 -1,
3 4 5 -1,
... ]
 }
}
```

Figura 2.5: Mesh triangolare rappresentata in formato VRML2.0

# Capitolo 3

# Metodi e Strumenti

In questo capitolo vengono presentati i principali metodi e strumenti, utilizzati durante la realizzazione dell'algoritmo di ricerca della complementarietà tra sezioni di proteine. Il primo passo è lo slicing della proteina partendo dalla rappresentazione della superficie realizzata attraverso PyMol. Successivamente per ogni sezione, dalle coordinate cartesiane dei punti sulla superficie, si è passati ad un sistema di coordinate curvilinee che descrive il perimetro di ogni slice. Il passo successivo riguarda il filtraggio tramite DFT del perimetro ed infine la ricostruzione del contorno.

## 3.1 Slicing della proteina

L'operazione di slicing della superficie molecolare consiste nel passaggio da una rappresentazione tridimensionale della proteina, ad una rappresentazione formata da un insieme di sezioni bidimensionali della stessa Figura 3.1.

Lo slicing viene effettuato considerano come asse di sezionamento l'asse di sviluppo della proteina, opportunamente allineato all'asse Z. Il metodo utilizzato per la costruzione delle sezioni della proteina, consiste nella individuazione dei punti della superficie, che intersecano il piano di taglio perpendicolare all'asse di sezionamento. Partendo dalla rappresentazione in formato VRML2.0, ad ogni lato di un triangolo della mesh viene associato un segmento, in questo modo per il triangolo con vertici $\widehat{ABC}$ vengono generati tre segmenti: $\overline{AB}$,$\overline{BC}$,$\overline{CA}$. Il piano di taglio viene rappresentato attraverso un vettore 3D, che indica la perpendicolare al piano stesso, ed un punto appartenente al piano. I punti appartenenti al

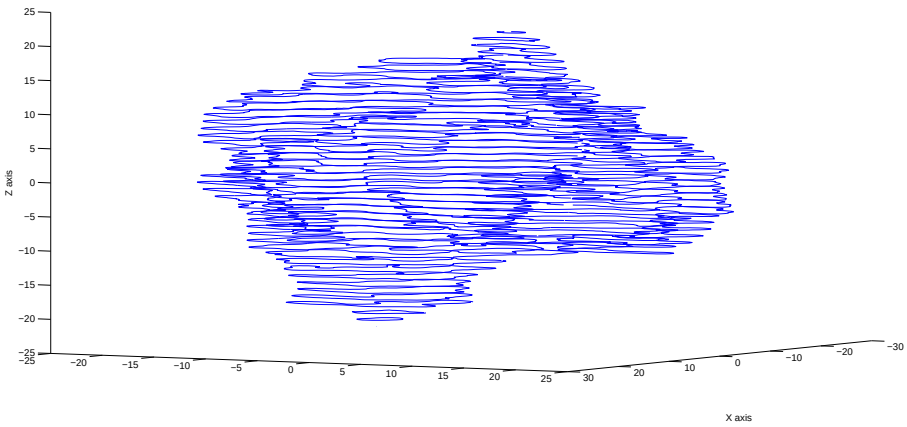


Figura 3.1: Superficie della proteina formata da un insieme di slice.

perimetro della slice sono tutti i punti per i quali un vertice del segmento poggia sul piano di taglio, oppure sono i punti di intersezione tra il piano ed il segmento che lo attraversa Figura 3.2.

Per ogni punto individuato viene tenuta traccia: delle coordinate cartesiane

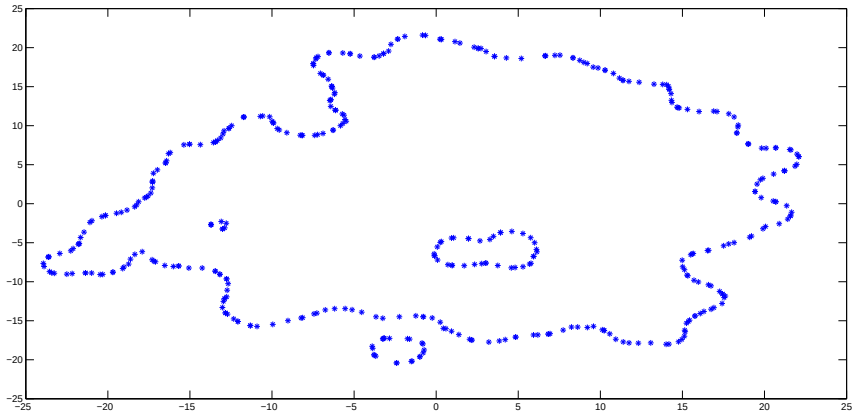

Figura 3.2: Punti su una sezione

$(x, y)$ calcolate, del triangolo a cui il segmento appartiene e le coordinate $(x, y, z)$ del vettore normale alla superficie del vertice più vicino al punto di intersezione. Una volta individuati i punti della superficie appartenenti al piano è necessario ordinare e collegare tali punti in modo da rappresentare il perimetro della slice. Nella rappresentazione del contorno si è deciso di considerare come punto in-

iziale $p_0(x_0, y_0)$ il punto appartenente al perimetro che interseca la semiretta $(-\infty, 0)$, e valutare i successivi punti in senso antiorario. Due punti appartenenti al perimetro sono consecutivi se appartengono allo stesso triangolo, in questo caso viene creato un segmento tra i due punti. Una volta individuati tutti i segmenti, il perimetro della slice viene realizzato con una successione ordinata degli stessi. Il collegamento tra segmenti consecutivi del perimetro avviene valutando l'ugualianza tra due vertici appartenenti a segmenti distinti Figura 3.3. La pro-

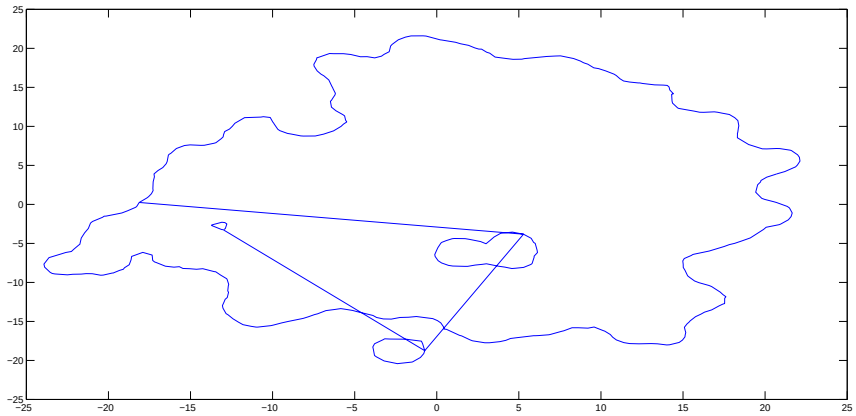


Figura 3.3: Punti ordinati di una sezione

cedura seguita in precedenza genera tutti i contorni della sezione, considerando anche eventuali isole interne o esterne al perimetro principale. Per valutare il contorno principale viene considerato esclusivamente il perimetro più lungo tra tutti quelli generati nella sezione Figura 3.4.

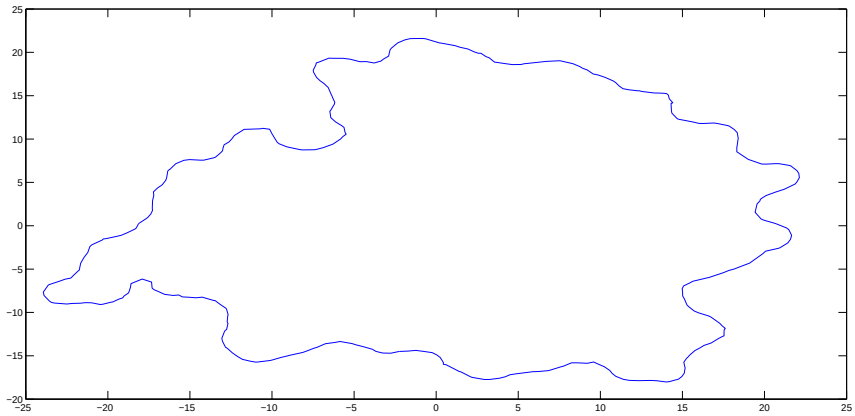


Figura 3.4: Perimetro principale di una sezione

## 3.2 Coordinata curvilinea

Una volta individuati tutti i punti appartenenti al perimetro della sezione, è necessario passare da un sistema a coordinate cartesiane, ad un sistema di coordinate curvilinee. Il cambio di sistema di riferimento permette di valutare la variazione della tangente lungo il perimetro della sezione.
Per ogni punto presente sul perimetro della sezione si applica la seguente funzione di trasformazione:

$$f : (x, y) \longmapsto (d, \theta)$$

Dove:

- $(x, y)$ sono le coordinate cartesiane dei punti situati sul perimetro.
- $d$ è la distanza percorsa lungo il perimetro partendo dal punto iniziale.
- $\theta$ è l'angolo tangente alla superficie calcolato nel punto.

Ogni segmento che costituisce il perimetro della sezione viene trasformato in un vettore applicato, perciò dato il segmento $\overline{ab}$ si ottiene il rispettivo vettore $\overrightarrow{ab}$. Il perimetro può esserere rappresentato da una successione di vettori orientati: $[\overrightarrow{ab}_0, \overrightarrow{ab}_1, ..., \overrightarrow{ab}_k, .., \overrightarrow{ab}_n]$ Figura 3.5.

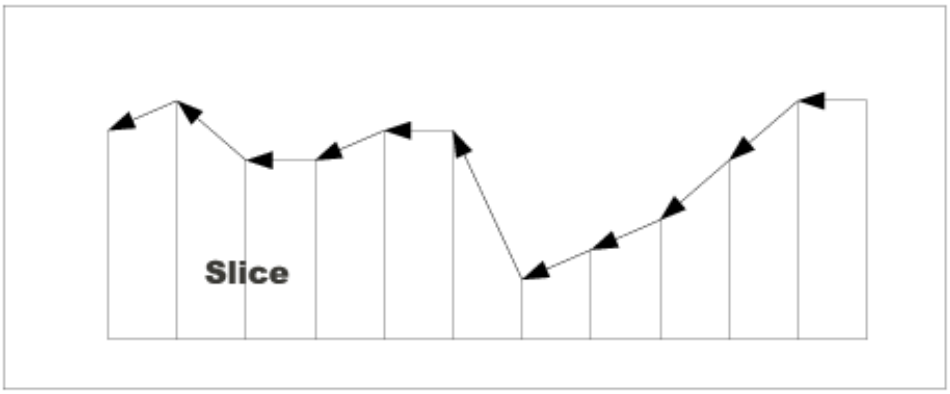


Figura 3.5: Rappresentazione del perimetro tramite successione di vettori orientati

Dato $\overrightarrow{ab}_k$ il vettore che rappresenta il $k$-esimo punto sul perimetro, il calcolo della distanza percorsa lungo il contorno, è la somma dei moduli dei vettori dal

punto iniziale al punto $k$ lungo il perimetro:

$$d_{k+1} = \sum_{i=0}^{k} |\overrightarrow{ab}_i| \quad k = 0, \ldots, N-1$$

Il vettore tangente al perimetro viene calcolato, per ogni punto del contorno, eseguendo un prodotto vettoriale tra il vettore normale al piano di taglio e la normale alla superficie nel punto. Definito $\vec{p}$ il vettore normale al piano di taglio e $\vec{n}$ il vettore normale alla superficie, il vettore $\vec{t} = \vec{p} \times \vec{n}$ , è la proiezione del vettore tangente alla superficie sul piano di taglio. Di tutti i vettori tangenti viene tenuta traccia del valore dell'angolo $\theta$ relativo alla coordinata angolare del vettore. Il valore dell' angolo $\theta$, viene ricavato dalle coordinate $(x, y)$ del vettore tangente, tramite la funzione atan2 implementata nella libreria java.math, e calcolato dalla formula:

$$\theta = 2 \arctan \left( \frac{y}{\sqrt{x^2 + y^2} + x} \right)$$

Al termite di questa operazione il perimetro è rappresentato da una successione di punti espressi in coordinata curvilinea: $[(d_0, \theta_0), (d_1, \theta_1)$ ,..., $(d_j, \theta_j)$, .., $(d_n, \theta_n)]$ Figura 3.6.

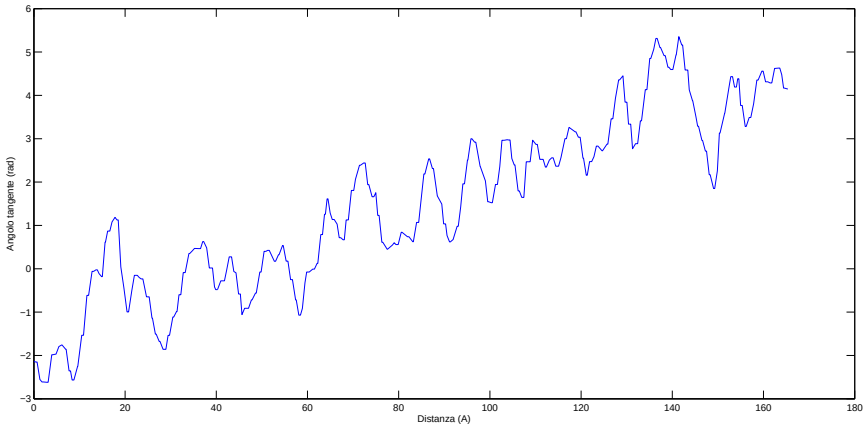


Figura 3.6: Rappresentazione del perimetro tramite coordinate curvilinee

## 3.3 Filtraggio del perimetro

Lo scopo del filtraggio della superficie (smoothing) consiste nell'evidenziare la struttura geometrica della stessa, eliminandone le rugosità presenti che non iterferiscono con il processo di docking. Considerando la rappresentazione in coordinata curvilinea del perimetro, come un segnale a "tempo" continuo, l'operazione di filtraggio consiste nell'eliminare le variazioni alle alte frequenze del valore dell'angolo tangente. Tale operazione viene effettuata eseguendo l'analisi di fourier del segnale. Il campionamento del segnale viene effettuato con un periodo di campionamento (distanza di campionamento) $d_c$ di 0,2 $\mathring{A}$ (frequenza di campionamento relativa $f_c$ pari a 5 $(1/\mathring{A})$). I valori della coordinata curvilinea relativi ai punti di campionamento vengono ricavati attraverso una funzione di interpolazione lineare,ottenendo una successione di N valori: $[\theta_0, \theta_1, ..., \theta_k, ..., \theta_{N-1}]$. A tale successione viene applicata la DFT (Discrete Fourier Transform).

$$\Theta_k = \sum_{n=0}^{N-1} \theta_n e^{-\frac{2\pi i}{N}kn} \qquad k = 0, \ldots, N-1$$

La funzione DFT e la successiva IDFT vengono implemetate attraverso l'algoritmo FFTW [12] . Il filtraggio del segnale viene efftettuato attraverso un filtro passa-basso ideale a frequenza di taglio $f_t$ fissata. La risposta in frequenza del filtro è:

$$H(f) = \begin{cases} 1 & \text{se } f \leq f_t \\ 0 & \text{se } f > f_t \end{cases} \tag{3.1}$$

Al segnale risultante $Y(f) = \Theta(f) \cdot H(f)$, viene applicata la trasformata inversa di fourier (IDFT).

$$y_k = \frac{1}{N} \sum_{n=0}^{N-1} Y_n e^{\frac{2\pi i}{N}kn} \qquad k = 0, \ldots, N-1$$

La successione $[y_0, y_1, ..., y_k, ..., y_{N-1}]$ rappresenta i valori dell'angolo tangente $\theta$ ricavati dopo l'operazione di filtraggio Figura 3.7.

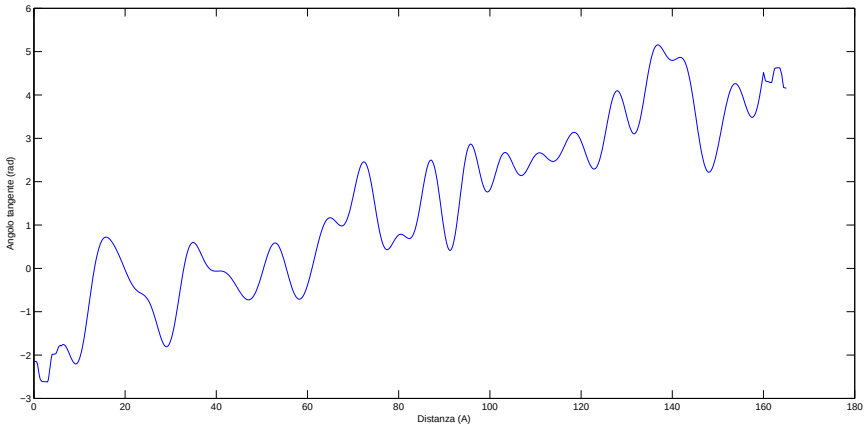


Figura 3.7: Rappresentazione del perimetro smooth tramite coordinate curvilinee

## 3.4 Ricostruzione del perimetro

Il passaggio da coordinate curvilinee a coordinate cartesiane consente di effettuare la ricostruzione del perimetro della slice.

$$f : (d, \theta) \longmapsto (x, y)$$

Dato un punto sul perimetro rappresentato in coordinate cartesiane $(x_i, y_i)$, la coordinata $x$ del punto successivo viene calcolata come $x_{i+1} = \cos(\theta_i) \cdot d_s$ e la rispettiva coordinata $y$ come $y_{i+1} = \sin(\theta_i) \cdot d_s$. Eseguendo tale operazione, partendo dal punto iniziale $p_0(x_0, y_0)$, su tutti i valori della della coordinata curvilinea $\theta_i$ con $i = 0, \ldots, N-1$, è possibile ricostruire il contorno della slice come in Figura 3.8.

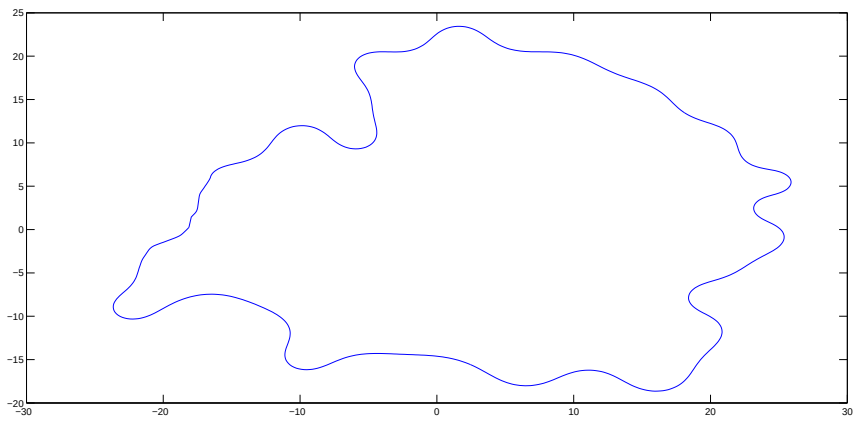


Figura 3.8: Rappresentazione del perimetro smooth tramite coordinate cartesiane

# Capitolo 4

# Valutazione della complementarietà

Nel presente capitolo viene presentato il metodo per la valutazione della complementarietà geometrica tra sezioni di superficie di due proteine, con l'obbiettivo di identificare possibili interfacce di docking. Il confronto viene effettuato, analizzando la rappresentazione tramite coordinata curvilinea, su porzioni di perimetro relative a due sezioni di proteine distinte.

## 4.1 Confronto tra perimetri

La valutazione del docking proteina-proteina, relativa a due sezioni bidimensionali, viene effettuato ricercando la complementarietà tra i perimetri di due proteine. Con l'obbiettivo di identificare delle aree di interfaccia per il docking, il confronto non viene eseguito su tutto il perimetro, ma su porzioni dello stesso, valutando la complementarietà tra porzioni dei due contorni.
Rappresentando il perimetro delle due sezioni attraverso una successione di vettori, la ricerca di un'interfaccia di docking può essere eseguita, valutando la corrispondenza dei vettori su delle porzioni di contorni. Dato un contorno $A = [\vec{a}_i, \vec{a}_{i+1}, ...., \vec{a}_{i+k}]$ composto da $k+1$ vettori e un contorno $B = [\vec{b}_j, \vec{b}_{j+1}, ...., \vec{b}_{j+k}]$ composto anch'esso da $k+1$ vettori, i due contorni sono complementari se: $\vec{a}_{i+l} = (\vec{b}_{j+k-l})^R$ per $l = 0, \ldots, k$. Il perimetro A è relativo alla proteina recettore, mentre il perimetro B alla proteina ligando.
Lo stesso concetto può essere utilizzato su una rappresentazione tramite coordi-

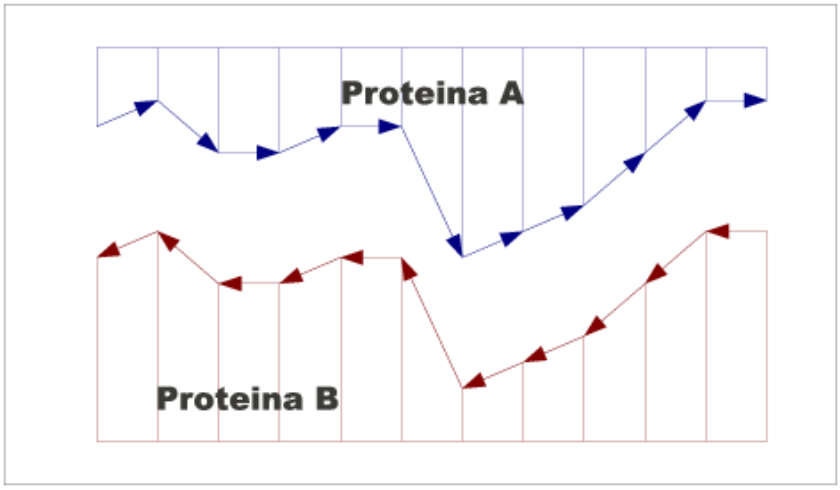


Figura 4.1: Confronto tra porzioni di due slice

nata curvilinea.
Considerando la rappresentazione della porzione di perimetro attraverso coordinata curvilinea, $A = [\theta_i, \theta_{i+1}, ...., \theta_{i+k}]$ e $B = [\theta_j, \theta_{j+1}, ...., \theta_{j+k}]$, il confronto tra due contorni può essere eseguito valuando la similarità dell'andamento tra i valori $\theta_{i+l}$ e $(\theta_{j+k-l} + \pi)$ per $l = 0, \ldots, k$.
Per facilitare la valutazione della complementarietà si è deciso di calcolare la coordinata curvilinea, di tutte le sezioni della proteina ligando, percorrendo il perimetro in senso orario, mentre il perimetro sulle sezioni della proteina recettore viene percorso in senso antiorario. Eseguendo tale operazione, il confronto tra due porzioni di perimetro di lunghezza k+1, può essere eseguito valutando la similarità dell'andamento tra i valori $\theta_{i+l}$ e $\theta_{j+l}$ per $l = 0, \ldots, k$.

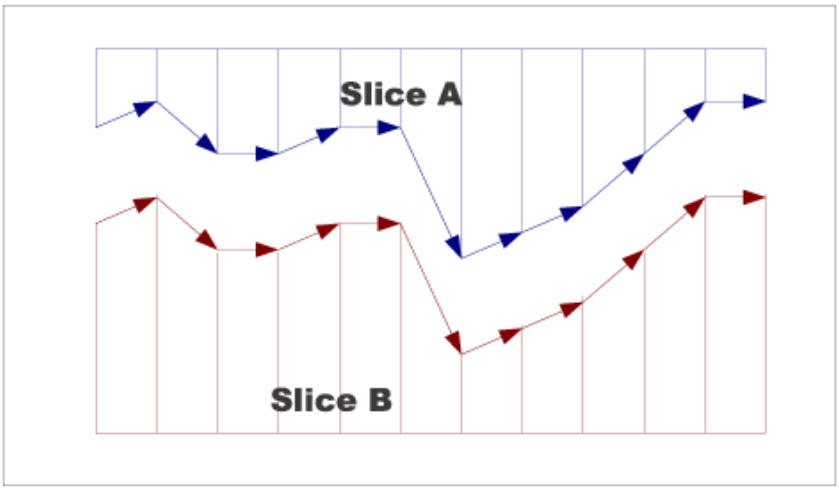


Figura 4.2: Confronto tra porzioni di due slice: slice A senso antiorario, slice B senso orario

## 4.2 Metrica utilizzata

Nella valutazione della similarità tra le due sezioni si è deciso di utilizzare la funzione RMSD (Root Mean Square Deviation), relativa ai valori di angolo tangente dei due perimetri. La funzione RMSD viene applicata a delle porzioni, dei due perimetri, aventi la stessa lunghezza. Si rende necessario, per valutare l'andamento della coordinata curvilinea, allineare i due valori iniziali dai quali inizia la valutazione della funzione RMSD. Sia $i$ il punto iniziale di valutazione per la sezione A e $j$ il punto iniziale per la sezione B, viene calcolato un valore di offset tra i valori di angolo tangente delle due coordinate curvilinee come: $o = a_i - b_j$. La funzione RMSD, dati i valori dell'angolo tangente in coordinate curvilinee delle di sezioni A e B viene così definita: Siano $A = [a_i, a_{i+1}, ...., a_{i+k}]$ e $B = [b_j, b_{j+1}, ...., b_{j+k}]$ i $k + 1$ valori dell'angolo $\theta$ rispettivamente delle slice A e B, il valore di RMSD viene calcolato come:

$$RMSD = \sqrt{\frac{\sum_{l=0}^{k}(a_{i+l} - b_{j+l} - o)^2}{k+1}}$$

Un valore basso di RMSD indica una elevata complementarietà tra le porzioni dei due perimetri, ed identifica quindi una possibile interfaccia di docking tra le due proteine.

## 4.3 Orientazione delle sezioni

L'orientazione della sezione ligando sulla sezione recettore viene effettuata attraverso l'operazione di ricostruzione della slice, partendo dalla rappresentazione in coordinate curvilinee. La slice relativa alla proteina recettore viene considerata fissa, mentre la sezione relativa alla proteina ligando viene ruotata e successivamente traslata, con l'obiettivo di allineare le interfacce identificate sui due perimetri. L'operazione di orientazione consiste in una rotazione seguita da una traslazione.
L'angolo di rotazione della proteina ligando rispetto alla proteina recettore, è il valore di offset $o$, calcolato nel processo di valutazione della comlementarietà. L'asse di rotazione della sezione coincide con il punto di origine $p_0(x_0, y_0)$ identificato durante la procedura di sezionamento. Le coordinate (x,y) dei punti ruotati sul perimetro sono calcolati come: $x_{i+1} = \cos(\theta_i + o) \cdot d_s$ e $y_{i+1} = \sin(\theta_i + o) \cdot d_s$ eseguita per tutti i valori $i = 0, \ldots, N-1$. Successivamente all'operazione di rotazione viene eseguita l'operazione di traslazione della sezione ligando sulla sezione recettore. Definiti $r$ e $l$ gli indici sulle coordinate curvilinee, relativi rispettivamente a recettore e ligando, dai quali inizia la valutazione della complementarietà, i valori $(x_t, y_t)$ del vettore di traslazione sono: $x_t = (x_r - x_l)$ e $y_t = (y_r - y_l)$. Le coordinate finali dei punti della sezione ligando sono calcolati come $x_i = x_i + x_t$ e $y_i = y_i + y_t$ per $i = 0, \ldots, N-1$.

# Capitolo 5

# Test e Risultati

## 5.1 Test effettuati

Seguendo i metodi presentati nei capitoli 3 e 4, è stato realizzato un algoritmo, scritto in linguaggio java, per l'analisi della complementarietà su sezioni bidimensionali tra due proteine. L'algoritmo riceve in input due file in formato VRML 2.0 (.wrl) contenente le informazioni sulla mesh triangolare, generata attraverso PyMol, relativi alle proteine recettore e ligando. Al termine dell'esecuzione vengono generati dei file, in formato testo (.txt), contenenti le coordinate cartesiane relative a coppie di slice opportunamente orientate.
Durante la fase di test la distanza tra i piani di taglio, sull'asse di sezionamento z, è stata fissata a 1 $\mathring{A}$. Per quanto riguarda il filtraggio del perimetro, la distanza di campionamento $d_s$ viene fissata a 0,2 $\mathring{A}$ e la frequenza di taglio $f_t$, del filtro passa basso, a 0,5 $1/\mathring{A}$.
L'algoritmo è stato testato su alcuni composti multimerici. Le coordinate atomiche dei polipeptidi che formano il composto multimerico sono state ricavate dal database PDB. Per ogni coppia di subunità polipeptidiche testate, l'analisi della complementarietà è stata valutata considerando tutte le possibili coppie slice recettore - slice ligando.
La valutazione della complementarietà tra le sezioni è stata eseguita considerando porzioni di perimetro di diversa lunghezza.

## 5.2 Risultati

I risultati vengono presentati attraverso le immagini delle sezioni recettore-ligando, generate tramite Matlab. I grafici sono realizzati attraverso la funzione plot(), applicata alle coordinate cartesiane, calcolate dall'algoritmo sviluppato.

### Composto 1A39 (ChainA + ChainB)

Vengono presentati i risultati dei test effettuati sul composto multimerico 1A39 composto da due subunità polipeptidiche A e B. Applicando l'algoritmo ed eseguendo il metodo di valutazione su una porzione di perimetro di lunghezza pari a 15 $\mathring{A}$, la coppia di sezioni che risultano maggiormente complementari (valore di rmsd pari a 0,149) sono: per il recettore (Chain A), la slice sul piano di taglio z=-1,725 $\mathring{A}$ (Figura 5.1), e per la sezione ligando (Chain B), la slice sul piano di taglio z=-2,656 $\mathring{A}$ (Figura 5.2). Il risultato finale della elaborazione dell'algoritmo viene presentato in Figura 5.3, nella quale la slice ligando è orientata correttamente rispetto alla slice recettore.

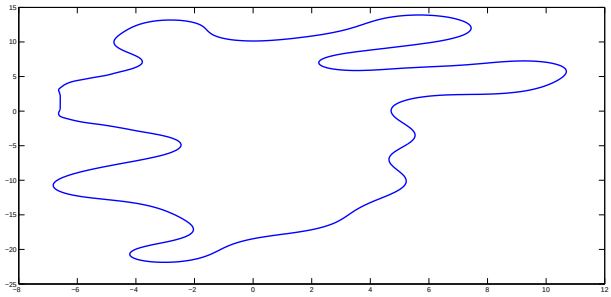


Figura 5.1: Slice recettore (1A39 Chain A) sul piano di taglio ad altezza z=-1,725

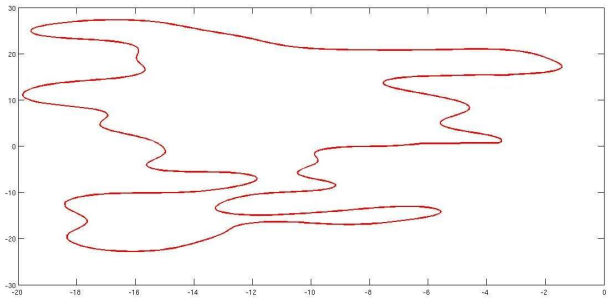


Figura 5.2: Slice ligando (1A39 Chain B) sul piano di taglio ad altezza z=-2,656

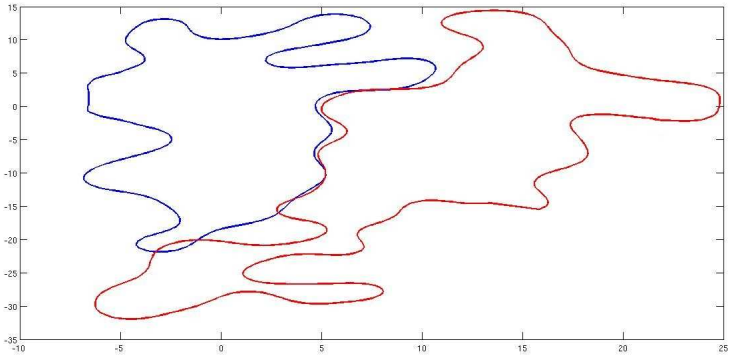


Figura 5.3: Orientazione della slice ligando sulla slice recettore

### Composto 1CGI (1CHG + 1HPT)

Vengono presentati i risultati dei test effettuati sul composto multimerico 1CGI composto da due subunità polipeptidiche 1CHG e 1HPT. Applicando l'algoritmo ed eseguendo il metodo di valutazione su una porzione di perimetro di lunghezza pari a 20 $\mathring{A}$, la coppia di sezioni che risultano maggiormente complementari (valore di rmsd pari a 0,173) sono: per il recettore (1CHG), la slice sul piano di taglio z=7,084 $\mathring{A}$ (Figura 5.4), e per la sezione ligando (1HPT), la slice sul piano di taglio z=2,866 $\mathring{A}$ (Figura 5.5). Il risultato finale della elaborazione dell'algoritmo viene presentato in Figura 5.6, nella quale la slice ligando è orientata correttamente rispetto alla slice recettore.

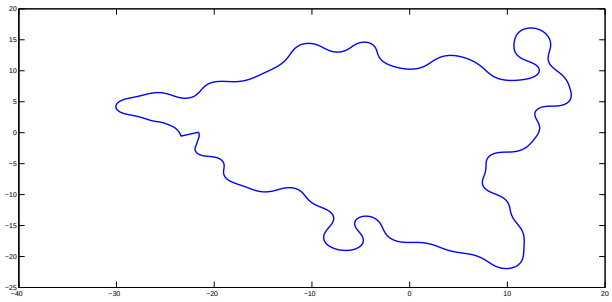


Figura 5.4: Slice recettore (1CHG) sul piano di taglio ad altezza z=7,084

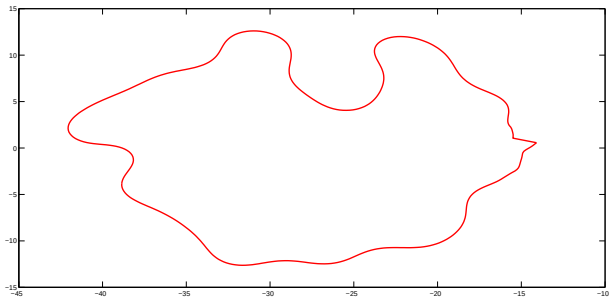


Figura 5.5: Slice ligando (1HPT) sul piano di taglio ad altezza z=2,866

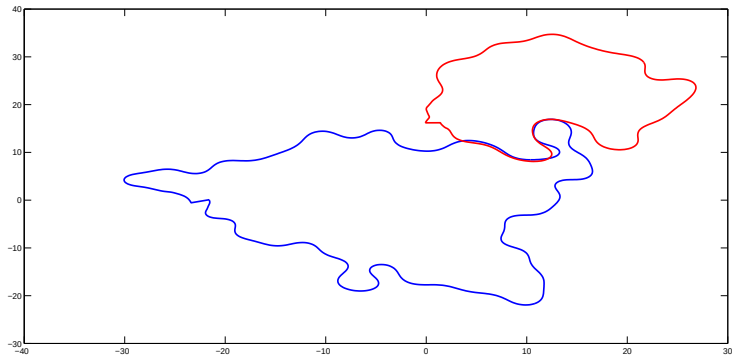


Figura 5.6: Orientazione della slice ligando sulla slice recettore

# Capitolo 6

# Conclusioni

L'obiettivo principale del progetto, che consiste nell'analisi di complementarietà su sezioni bidimensionali di proteine, può considerarsi raggiunto. Dai grafici dei risultati relativi ai test eseguiti (Figura 5.3, Figura 5.6), è possibile verificare la correttezza del metodo e dell'algoritmo implementato per l'individuazione della complementarietà tra perimetri di due sezioni bidimensionali.
Il metodo sviluppato però non tiene in considerazione di possibili sovrapposizioni tra i due perimetri, si rende dunque necessaria una verifica a posteriori, per eliminare le orientazioni che evidenziano tale problema.
L'obiettivo secondario, ovvero l'estensione del metodo a tutte le sezioni ricavate dalle due proteine, per identificare un docking proteina-proteina, rimane ancora un problema aperto. La realizzazione del progetto prevede la valutazione della complementarietà su più sezioni bidimensionali, mantenendo il corretto allineamento tra le slice, evitando il fenomeno di torsione della subunità polipeptidica. Nel corso del lavoro di tesi sono state formulate alcune ipotesi di lavoro per ricercare tale complementarietà. Una di queste prevede la ricerca della coppia di sezioni che ottiene il maggiore livello di complementarietà, e la relativa orientazione della slice ligando sul recettore, applicando l'algoritmo sviluppato. Successivamente l'orientazione sul piano (x,y) ricavata viene applicata a tutte le sezioni relative alla proteina ligando, opportunamente allineate sull'asse di sezionamento. Tale metodo però evidenzia un problema; la corretta complementarietà tra una coppia di slice non è sufficiente per definire la complementarietà tra coppie successive, perciò risulta necessario verificare l'orientazione di tutte le coppie generate. L'applicazione dell'algoritmo sviluppato perciò non risulta effi-

cace nella ricerca iniziale del docking rigido tra proteine, tuttavia può risultare uno strumento utile nella fase di selezione delle interfacce di docking, in un approccio multistage al problema. Applicando l'algoritmo esclusivamente alle aree candidate ad essere interfacce è possibile valutarne l'effettiva complementarietà geometrica, scartando le combinazioni meno favorevoli.

# Bibliografia


[1] E. Katchalski-Katzir, I. Shariv, M. Eisenstein, a. a. Friesem, C. Aflalo, and I. a. Vakser, "Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques.," in *Proceedings of the National Academy of Sciences of the United States of America*, vol. 89, pp. 2195–9, Mar. 1992.

[2] C. J. Camacho and S. Vajda, "Protein docking along smooth association pathways.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, pp. 10636–41, Sept. 2001.

[3] R. Chen, "Docking unbound proteins using shape complementarity, desolvation, and electrostatics," *Proteins*, vol. 47, no. 3, pp. 281–294, 2002.

[4] M. F. Lensink, R. Méndez, and S. J. Wodak, "Docking and scoring protein complexes: CAPRI 3rd Edition," *Proteins: Structure, Function, and Bioinformatics*, vol. 9999, no. 9999, pp. NA+, 2007.

[5] S. Vajda and D. Kozakov, "Convergence and combination of methods in protein-protein docking.," *Current opinion in structural biology*, vol. 19, pp. 164–70, Apr. 2009.

[6] D. Kozakov, D. R. Hall, D. Beglov, R. Brenke, S. R. Comeau, Y. Shen, K. Li, J. Zheng, P. Vakili, I. C. Paschalidis, and S. Vajda, "Achieving reliability and high accuracy in automated protein docking: ClusPro, PIPER, SDU, and stability analysis in CAPRI rounds 13-19.," *Proteins*, vol. 78, pp. 3124–30, Nov. 2010.

[7] D. Kozakov, R. Brenke, S. R. Comeau, and S. Vajda, "PIPER: An FFT-Based Protein Docking Program with Pairwise Potentials," *Bioinformatics*, vol. 406, no. August, pp. 392–406, 2006.

[8] S. Lorenzen and Y. Zhang, "Identification of Near-Native Structures by Clustering Protein Docking Conformations," *Bioinformatics*, vol. 194, no. August 2006, pp. 187–194, 2007.

[9] D. Kozakov, O. Schueler-Furman, and S. Vajda, "Discrimination of near-native structures in protein-protein docking by testing the stability of local minima.," *Proteins*, vol. 72, pp. 993–1004, Aug. 2008.

[10] C. Dominguez, R. Boelens, and A. M. J. J. Bonvin, "HADDOCK: a protein-protein docking approach based on biochemical or biophysical information.," *Journal of the American Chemical Society*, vol. 125, pp. 1731–7, Feb. 2003.

[11] M. V. Dijk, S. J. D. Vries, A. D. J. V. Dijk, A. Thureau, V. Hsu, T. Wassenaar, and A. M. J. J. Bonvin, "HADDOCK versus HADDOCK: New features and performance of HADDOCK2.0 on the CAPRI targets," *Proteins*, pp. 726–733, 2007.

[12] M. Frigo and S. G. Johnson, "FFTW: An adaptive software architecture for the FFT," in *Proc. 1998 IEEE Intl. Conf. Acoustics Speech and Signal Processing*, vol. 3, pp. 1381–1384, IEEE, 1998.

[13] P. E. Bourne and H. Weissig, *Structural Bioinformatics; Methods of Biochemical Analysis*, vol. 44. John Wiley and Sons, 2003.

[14] J. J. Gray, S. Moughon, C. Wang, O. Schueler-Furman, B. Kuhlman, C. a. Rohl, and D. Baker, "Protein–Protein Docking with Simultaneous Optimization of Rigid-body Displacement and Side-chain Conformations," *Journal of Molecular Biology*, vol. 331, pp. 281–299, Aug. 2003.

# Ringraziamenti

Eccomi qua, sono arrivato alla fine di questa tesi stanco ma contento, soddisfatto dell'obbiettivo che ho raggiunto.

Al termine di questo lungo percorso è giunto il momento di ringraziare tutti coloro che mi sono stati vicini.

Vorrei iniziare questi ringraziamenti dai miei genitori. Grazie mamma e papà, per la fiducia che mi avete dato.

Grazie a tutta la famiglia per il sostegno e la serenità, che mi hanno permesso di affrontare questa sfida.

Un grazie a tutti gli amici incontrati a Padova, che con le varie feste mi hanno concesso un pò di svago dallo studio.

Un ringraziamento speciale và a tutti i coinqulini e ex-coinqulini di Via Trieste, è stato bello condividere con voi questo periodo della vita.

Grazie anche a tutti gli amici Fabio, Bac, Nicola, Veronica, Paolo, Ombra, Carlet, Balda... e alla squadra di calcio a 5, è sempre un piacere stare in vostra compagnia e passare delle belle serate.

Grazie anche al ristorante "Da Brun" che ha "finanziato" gli studi... e ai colleghi di lavoro.

Spero di non aver dimenticato nessuno, e se l'ho fatto, non era nelle mie intenzioni.

A tutti voi, grazie!