

Progetto Ingegneria Informatica  
Identificazione di tasche o cavità nelle proteine  
con i metodi POCASA e PASS

Anno Accademico 2021/2022

Viola Renne	932160
Luca Romanò	934229

Docente: Prof. Gianluca Palermo

## Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
1.1	Obiettivo . . . . .	2
<b>2</b>	<b>Specifiche</b>	<b>2</b>
2.1	POCASA . . . . .	2
2.2	PASS . . . . .	3
<b>3</b>	<b>Implementazione</b>	<b>4</b>
3.1	Formati e Librerie . . . . .	4
3.1.1	Formato file PDB . . . . .	4
3.1.2	BioPandas . . . . .	5
3.2	Tecniche di estrazione di tasche o cavità . . . . .	5
3.2.1	POCASA . . . . .	5
3.2.2	PASS . . . . .	6
<b>4</b>	<b>Valutazione dei metodi</b>	<b>7</b>
4.1	POCASA . . . . .	7
4.2	PASS . . . . .	8
4.3	Confronto tra i due algoritmi . . . . .	9
4.3.1	Confronto visivo . . . . .	9
4.3.2	Confronto algoritmico . . . . .	10
<b>5</b>	<b>Conclusioni</b>	<b>10</b>
<b>6</b>	<b>Sitografia</b>	<b>10</b>

# 1 Introduzione

## 1.1 Obiettivo

Il progetto riguarda l'implementazione di due metodi, POCASA e PASS, il cui obiettivo è quello di predire i siti di legame trovando le cavità delle proteine. L'identificazione delle cavità delle proteine è un passo fondamentale per la ricerca di nuovi farmaci. Entrambi gli algoritmi utilizzano un metodo puramente geometrico, per il quale serve conoscere la struttura 3D della proteina e vengono usati per delineare quale zona vicina alla proteina rappresenta una tasca. I due metodi si differenziano per il modo con il quale questo avviene: mentre il metodo POCASA identifica la tasca tramite una griglia regolare come volume non accessibile da una sfera di grandi dimensioni, PASS usa una serie di piccole sfere sonda che vengono costruite a livelli in passaggi consecutivi riempiendo le cavità.

Qui di seguito un esempio con la proteina 1a0q e i due risultati di PASS e POCASA.

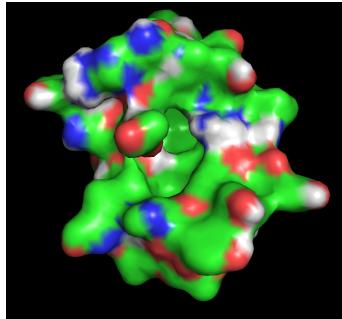
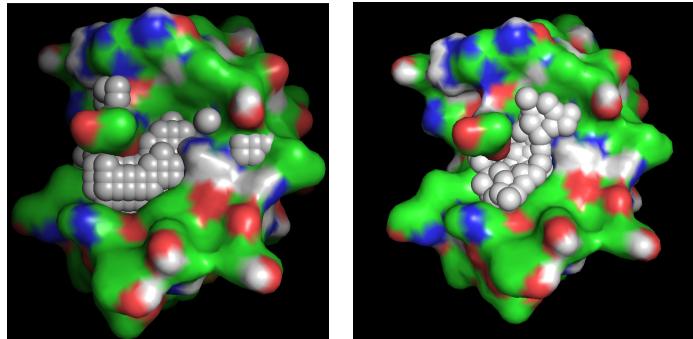


Figura 1: Proteina 1a0q



(a) Risultato POCASA

(b) Risultato PASS

Figura 2: Risultati finali

# 2 Specifiche

Di seguito vengono riportate le specifiche dei due metodi. Entrambi gli algoritmi prendono in ingresso un file PDB (il cui formato è spiegato al paragrafo 3.1.1) contenente gli atomi della proteina e la loro posizione e al termine generano un file PDB con le coordinate dei punti discreti che identificano le tasche o le cavità.

## 2.1 POCASA

POCASA è un algoritmo per prevedere i siti di legame rilevando tasche e cavità di proteine con una sfera di prova. L'algoritmo inizialmente crea una griglia e determina per ogni punto se esso è parte della proteina o meno. Successivamente viene utilizzata una sfera per definire il bordo della proteina e ciascun punto tra il bordo e la proteina viene indicato come punto appartenente a una tasca o cavità. I passi dell'algoritmo di POCASA sono descritti di seguito e sono rappresentati in figura 3.

**Eliminazione atomi idrogeno** POCASA inizia leggendo le coordinate della proteina presenti nel file PDB, elimina gli idrogeni e assegna ai raggi atomici di van der Waals dei valori superiori rispetto a quelli elementari per compensare l'eliminazione degli atomi di idrogeno.

**Creazione griglia** Il primo passo consiste nel creare una griglia tridimensionale di lato 1.0Å oppure 0.5Å. La scelta del lato della griglia viene effettuata dall'utente. La griglia verrà successivamente riempita di valori 0 e 1. Il valore 0 indica lo spazio vuoto, mentre il valore 1 indica la proteina. Al termine di questo passaggio avremo la struttura 3D della proteina rappresentata dai valori 1 nella griglia.

**Bordo della proteina** Una sfera di prova, con raggio scelto dall'utente, viene inserita in ogni punto della griglia. Se la sfera non si sovrappone a punti della griglia dal valore 1, allora tutti i punti appartenenti ad essa vanno a formare il bordo della proteina e viene attribuito loro il valore -1.

**Cavità della proteina** I punti della griglia di valore 0, ovvero quelli che non fanno parte né della proteina (valore 1) né del bordo (valore -1), vengono individuati come cavità della proteina.

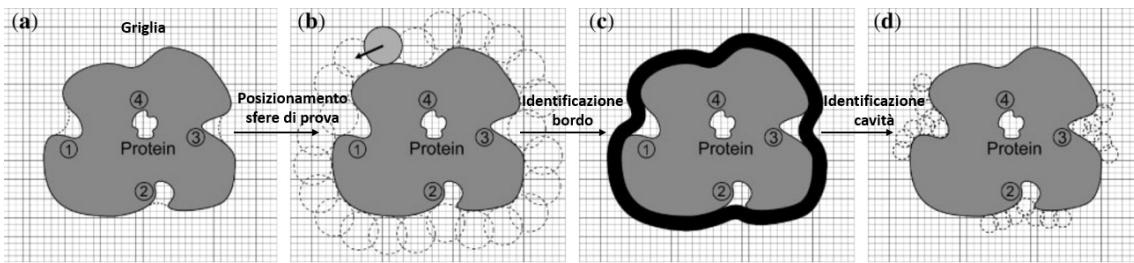


Figura 3: Algoritmo di Pocasa

**Punti di disturbo** Per rimuovere i "punti di disturbo" viene definito un parametro soglia SPF (Single Point Flag). Per ogni punto identificato come cavità, viene calcolato il numero di punti confinanti considerati cavità. Se questo valore è inferiore alla soglia, presa in input, allora questo punto viene rimosso dalla cavità.

**Raggruppamento e ordinamento** Le cavità vengono raggruppate e successivamente ordinate. A ciascun punto di una cavità viene assegnato un peso determinato dalla minima distanza di Manhattan dal bordo della proteina. Al termine di questo conteggio, a ciascuna cavità viene attribuito un punteggio determinato dalla somma dei pesi dei suoi punti. Le cavità vengono ordinate in maniera decrescente sulla base di questo punteggio che rappresenta l'ordine di importanza della tasca.

## 2.2 PASS

L'algoritmo PASS è progettato per riempire le cavità di una struttura proteica con un insieme di sfere sonda e, tra queste, identificare i centri delle tasche di legame. I passi dell'algoritmo di PASS sono descritti di seguito e sono rappresentati in figura 4.

**Atomi di idrogeno** PASS inizia leggendo le coordinate della proteina presenti nel file PDB e assegnando i raggi atomici elementari. Poiché una proteina con atomi di idrogeno esplicitamente rappresentati contiene meno volume interstiziale di una senza idrogeno, PASS assegna alcuni valori di parametro diversi nei due casi: se meno del 20% degli atomi nel file PDB della proteina sono idrogeno, allora tutti gli atomi di idrogeno vengono rimossi; altrimenti, viene mantenuto l'idrogeno.

**Primo strato** Il primo strato di sfere sonda viene calcolato facendo un loop su tutte le triplettie uniche di atomi di proteina e per ciascuna tripla vengono individuate le due posizioni in cui una sfera sonda può trovarsi tangente a tutti e tre gli atomi di proteina.

**Filtro** Una presunta sfera sonda deve sopravvivere a diversi filtri. Il primo è che la sonda non possa sovrapporsi ad alcun atomo del substrato di accrescimento. Il secondo proibisce esplicitamente alla sonda di scontrarsi con qualsiasi atomo di proteina. Il terzo filtro assicura che la sonda si trovi sufficientemente contornata da atomi della proteina. In particolare, ad ogni sfera sonda è attribuito un "conteggio di sepoltura", calcolato contando il numero di atomi della proteina che si trovano all'interno di un raggio di 8Å. Una sonda è filtrata se il suo "conteggio di sepoltura" è inferiore a una data soglia. Con l'ultimo filtro le sfere sonda vengono spaziate in modo tale che non ci siano due centri più vicini di 1Å.

**Strati supplementari** Dopo che lo strato iniziale viene calcolato, gli strati supplementari sono ottenuti iterativamente dallo strato di sfere esistenti. Ad ogni iterazione, viene calcolata una serie di nuove sfere sonda, ma con un raggio più piccolo e con l'insieme di tutte le sfere sonda trattenute dagli strati precedenti come substrato di accrescimento. Queste nuove sfere vengono filtrate come descritto in precedenza.

**Risultato** PASS continua la fase di accrescimento finché non si incontra uno strato in cui nessuna delle sfere sonda trovate sopravvive ai filtri. Il risultato di questa procedura è che le cavità nella proteina sono riempite di un insieme di sfere uniformemente spaziate. A questo punto avviene una fase di filtraggio ulteriore dove vengono eliminate le sfere con meno di quattro sfere sonda nel raggio di 2.5Å.

**ASP** Al termine della computazione delle sfere sonda, PASS trova dei punti di siti attivi (Active Site Points). Per farlo, assegna ad ogni sfera sonda un peso (Probe Weight) e successivamente esse vengono disposte in ordine decrescente sulla base del PW. Le sonde vengono identificate come ASP se hanno un peso maggiore di 1100 e se si trovano a una distanza maggiore di 8Å da un ASP già identificato (e dunque con un valore di PW maggiore).

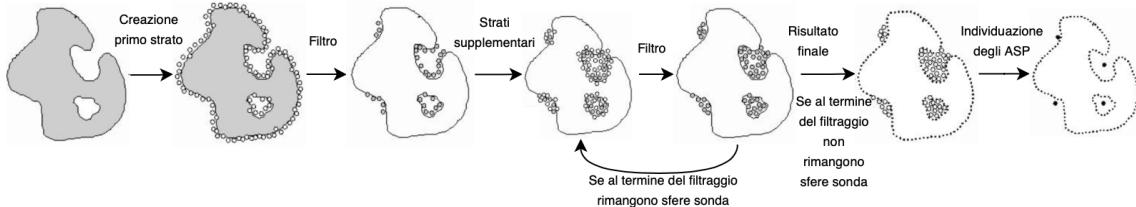


Figura 4: Algoritmo di PASS

## 3 Implementazione

### 3.1 Formati e Librerie

La struttura 3D della proteina viene fornita attraverso file con estensione PDB 3.1.1. Per la lettura di questi file viene utilizzata la libreria BioPandas 3.1.2. Inoltre, questa libreria viene anche utilizzata per ottenere il campo ATOM dei file PDB e per rimuovere gli idrogeni quando si verificano le condizioni spiegate nella sezione "Specifiche".

#### 3.1.1 Formato file PDB

Protein Data Bank è un formato file standard per macromolecole biologiche. L'estensione dei file è .pdb il cui formato è column specific, ovvero ogni colonna deve contenere uno specifico valore. Al fine di questo progetto, sono di particolare importanza i campi ATOM dei file PDB forniti in input. Un esempio è illustrato di seguito:

ATOM 1 N ILE A 16 -8.048 9.475 20.073 1.00 4.91 N

ATOM indica che si tratta di un atomo standard della proteina. Un altro campo possibile è HETATM che sta per heteroatom, ovvero un atomo non standard per la proteina. Nel nostro progetto i campi più rilevanti sono le coordinate degli atomi (campi sette, otto e nove) e l'elemento (campo dodici). Le coordinate ci consentono di conoscere la posizione dell'atomo e l'elemento è correlato al raggio della sfera che rappresenta l'atomo stesso.

### 3.1.2 BioPandas

BioPandas è una libreria python che permette di lavorare con strutture molecolari utilizzando Pandas DataFrames. BioPandas legge le strutture molecolari da file con coordinate 3D, come ad esempio PDB, e le trasforma in Pandas DataFrame. Le principali funzioni di BioPandas che abbiamo utilizzato sono:

```
# Lettura di un file , in questo caso con path = './protein.pdb'
pdb.read_pdb('./protein.pdb')
# Selezione dei soli campi ATOM e delle sole colonne per le
# coordinate e per l'elemento
pdb.df[ 'ATOM' ][[ 'x_coord', 'y_coord', 'z_coord', 'element_symbol' ]]
# Eliminazione degli atomi di idrogeno
pdb.df[ 'ATOM' ][pdb.df[ 'ATOM' ][ 'element_symbol' ] != 'H']
```

## 3.2 Tecniche di estrazione di tasche o cavità

### 3.2.1 POCASA

L'implementazione di POCASA, come descritto in sezione 2.1, si suddivide in diversi passaggi:

**Riga di comando** All'utente viene richiesto di inserire vari parametri elencati di seguito:

- lato della griglia che può assumere valore 1.0Å oppure 0.5Å;
- il path del file con estensione .pdb contenente la proteina di cui si vogliono identificare le cavità;
- il parametro SPF che deve essere un valore intero da 0 a 26;
- il parametro Top N che indica quante cavità visualizzare nel file di output;
- un valore booleano per decidere in che modo attribuire il peso ai vari punti di una cavità. Se scelto il valore 1, ai punti viene attribuito come peso la minima distanza di Manhattan dal bordo della proteina. Se scelto il valore 0, si considera peso unitario.

**Inizializzazione della griglia** Le coordinate dei centri di ciascun atomo vengono traslate, ciascuna sulla base del minimo e massimo valore tra tutti gli atomi di proteina e per ciascuna coordinata. In questo modo è possibile visualizzare la griglia in una matrice con indici tra zero e la differenza tra il massimo e minimo per ogni coordinata. Viene definito un fattore di scala che avrà valore 1 nel caso in cui il lato della griglia è 1 Å, 2 se il lato è di 0.5Å. In questo secondo caso, tutti i valori della matrice ( $x, y, z$ ) vengono considerati come  $(x/2, y/2, z/2)$  utilizzando il fattore di scala. Successivamente si prosegue nel riempire la griglia con valori 1 nei punti che si sovrappongono con atomi della proteina. Quindi, per ciascun atomo della proteina vengono controllati tutti i possibili punti della griglia che si potrebbero sovrapporre ad esso. Per sapere se la sovrapposizione si è verificata, viene calcolata la distanza tra l'atomo e i punti della griglia e, se questo valore risulta minore del raggio dell'atomo, allora al punto considerato viene attribuito il valore 1. I punti che non si sovrappongono con gli atomi della proteina manterranno valore 0.

**Creazione del bordo** Viene creato il bordo della proteina facendo muovere una sfera di prova di raggio dato nei vari punti della griglia. Per ogni punto appartenente alla sfera, cioè con distanza dal centro della sfera inferiore al suo raggio, viene controllata la presenza di una sovrapposizione con la proteina (punti della griglia dal valore 1). Se è presente, allora si muove il centro della sfera. Se invece non è presente alcuna sovrapposizione, tutti i punti della sfera vengono contrassegnati come bordo della proteina e si assegna loro il valore -1.

**Rimozione dei punti di disturbo** Tutti i punti che hanno mantenuto il valore iniziale di 0 sono punti potenzialmente appartenenti ad una cavità. Successivamente viene effettuato un filtraggio che si basa su un parametro scelto inizialmente, chiamato "SPF". Per ciascun punto che potenzialmente rappresenta una cavità vengono contati il numero di punti di valore 0 attorno ad esso. Se il conteggio è minore della soglia "SPF", allora quel punto non sarà più considerato come cavità.

**Identificazione dei gruppi di cavità** Per associare ogni punto a una specifica cavità viene utilizzato l'algoritmo BFS. Esso parte da un punto e si propaga a tutti i punti adiacenti associando ad ogni

gruppo un valore maggiore o uguale a due, che verrà poi associato ad una lettera nel file di output.

**Creazione della lista contenente le cavità** I gruppi di cavità vengono ordinati in maniera decrescente tramite l'algoritmo "counting sort". Ogni punto viene pesato a seconda della scelta effettuata dall'utente, descritta al paragrafo 3.2.1. Viene restituita una lista di punti di cavità e, per ciascuno di essi, la lettera identificativa del gruppo a cui appartiene e tutte le altre informazioni necessarie.

### 3.2.2 PASS

L'algoritmo di PASS, descritto nella sezione 2.2, è composto da tre passaggi principali: l'identificazione delle sfere sonda, il filtraggio di queste e l'individuazione degli ASP. Di seguito viene esposta l'implementazione di queste tre parti.

**Identificazione delle sfere sonda** Le sfere sonda del primo strato vengono trovate come descritto al paragrafo 2.2. Per ciascuna tripletta, si cerca di individuare due sfere sonda tangenti a tutti e tre gli atomi di proteina. Il procedimento degli strati supplementari è descritto al paragrafo 2.2. Durante l'implementazione di questa parte, vengono svolti vari controlli:

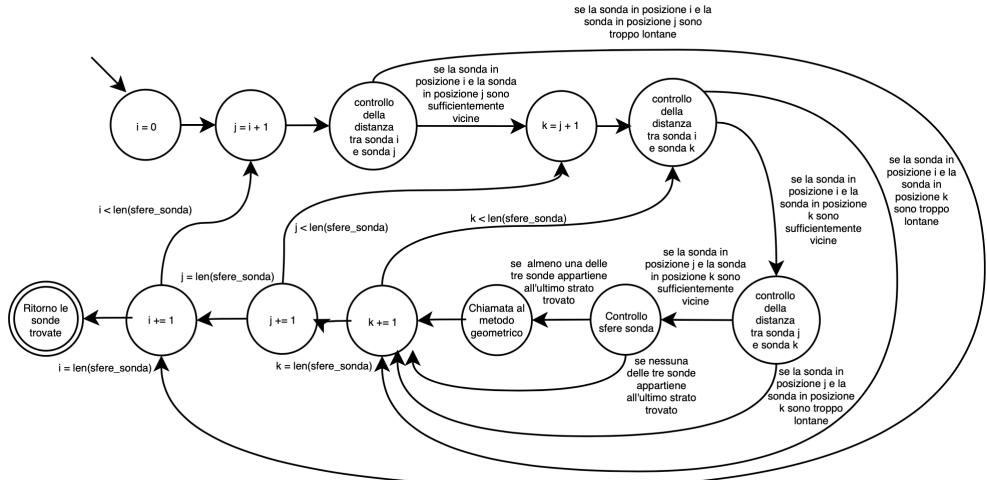


Figura 5: Controlli eseguiti nella funzione che itera le varie triplett di sfere sonda

Per implementare il metodo geometrico è stata utilizzata la libreria NumPy e in particolare sono stati usati i metodi `np.linalg.norm` per trovare la norma di un vettore, `np.dot` per calcolare il prodotto scalare tra due vettori e `np.random.rand(3)` per generare un vettore random di dimensione 3. Quest'ultima funzione è servita per ottenere il vettore  $z$ , perpendicolare a due vettori dati ( $x$  e  $y$ ). Per ottenere  $z$  veniva abbinata questa funzione all'ortogonalizzazione di Gram-Schmidt.

**Filtri** Le sfere trovate al passo precedente vengono filtrate. I filtri sono descritti al paragrafo 2.2. Il terzo filtro calcola il "conteggio di sepoltura" di ciascuna sonda, ovvero trova il numero di atomi di proteina presenti in un raggio di 8 Å dal centro della sonda. La documentazione di PASS non indica con precisione se debbano essere contati solamente gli atomi il cui centro è compreso nei 8 Å, ma si è deciso, a seguito di vari test, di considerare questa alternativa. L'ultimo filtro controlla che le sfere del nuovo strato abbiano una distanza centro-centro di almeno 1 Å e, in caso questa condizione non sia verificata, mantiene solo la sfera con "conteggio di sepoltura" maggiore.

**ASP** Per identificare gli ASP viene inizialmente calcolato il probe weight (PW) di ciascuna sonda. Per farlo viene utilizzata la seguente formula, dove  $BC(j)$  indica il conteggio di sepoltura della sonda in posizione  $j$  e  $R_0$  e  $D_0$  sono parametri forniti dalla documentazione di PASS:

$$PW(i) = \sum_{j=1}^{N_{sferesonda}} BC(j) * \exp(-(|r_i - r_j| - R_0)^2 / D_0^2)$$

**Riga di comando** All'utente è richiesto inserire tre path, tutti con estensione .pdb: il path del file di input, il path del file in cui inserire le sfere sonda individuate e il path del file in cui inserire gli ASP.

Inoltre, è possibile inserire ”-all” nel caso in cui non si volesse eseguire il filtraggio finale, descritto al paragrafo 2.2. In mancanza del comando ”-all” quest’ultimo filtro viene eseguito di default.

## 4 Valutazione dei metodi

Di seguito vengono esposte le principali caratteristiche dei due algoritmi e vengono confrontati i risultati ottenuti.

### 4.1 POCASA

Nell’algoritmo di POCASA, le cavità individuate dipendono dai diversi parametri inseriti in input, descritti al paragrafo 3.2.1. Di seguito si cerca di spiegare come mai questi valori influenzino la scelta delle cavità in questo programma. Il lato della griglia può essere di lunghezza 1 $\text{\AA}$  oppure 0.5 $\text{\AA}$ . Con lato 0.5 $\text{\AA}$  vengono trovati buchi di dimensione minore rispetto alla soluzione con lato 1 $\text{\AA}$ , poiché la griglia con lato 0.5 $\text{\AA}$  è più precisa nel definire la proteina. Il raggio della sfera di prova è il parametro di ingresso più significativo, per quanto riguarda l’identificazione delle cavità. Le cavità molto strette vengono trovate con qualsiasi raggio, mentre le cavità più larghe necessitano di un raggio sufficientemente grande. Questo perché una sfera di prova con un raggio piccolo può essere posizionata all’interno di una cavità larga senza che essa si sovrapponga alla proteina. Serve quindi scegliere opportunamente il raggio della sfera di prova in base alla forma della proteina. Dopo diversi test si può concludere che un raggio con dimensione di 4 $\text{\AA}$  può essere la scelta migliore da fare a priori senza conoscere la forma della proteina. Per quanto riguarda il parametro SPF (descritto al paragrafo 2.1) è generalmente consigliato utilizzare il valore 16.

Di seguito si riportano i risultati dei test effettuati con la proteina 1biw. Come parametri si è scelto: lato della griglia = 1 $\text{\AA}$ , SPF=16, Top N=10, ordinamento tramite la distanza di Manhattan. Le due immagini differiscono solamente per il raggio della sfera di prova. Dalle due immagini si può notare che le cavità individuate con un raggio di 4 $\text{\AA}$  hanno un volume maggiore e sono anche di numero maggiore.

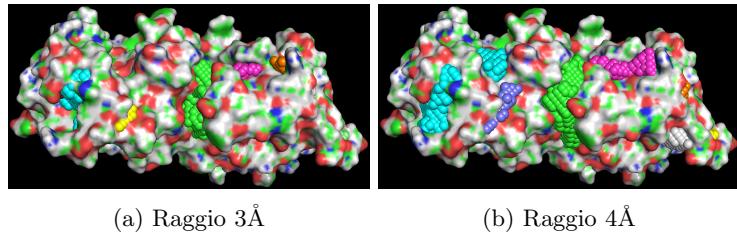


Figura 6: Risultati algoritmo POCASA con proteina 1biw

Nelle immagini seguenti si può vedere la differenza nell’utilizzo di un lato di 1 $\text{\AA}$  o di 0.5 $\text{\AA}$ . Il raggio utilizzato è di 4 $\text{\AA}$ .

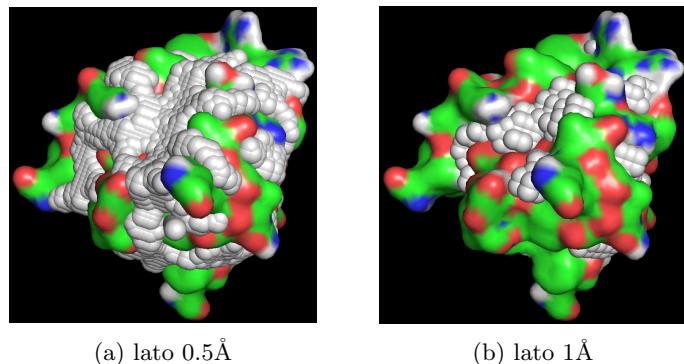


Figura 7: Risultati algoritmo POCASA con proteina 2w41

## 4.2 PASS

Nell'algoritmo di PASS le cavità della proteina vengono correttamente individuate. Inoltre, il ligando fornito viene coperto parzialmente o totalmente dalle sfere sonda. Di seguito viene mostrato un esempio. Il ligando, in questo caso specifico, non è interamente ricoperto in quanto si estende oltre la superficie della proteina (figura 8d). Questo è dovuto al filtraggio che avviene per il conteggio di sepoltura, nel quale le sfere sonda troppo esterne e superficiali vengono scartate. Di conseguenza, quando i ligandi si estendono allontanandosi dalla superficie della proteina, questi non vengono del tutto ricoperti dalle sfere sonda individuate da PASS. Anche gli ASP vengono trovati correttamente, in quanto sono sempre posizionati in prossimità del ligando fornito. In questo caso, l'ASP si trova proprio al centro del ligando (figura 8e).

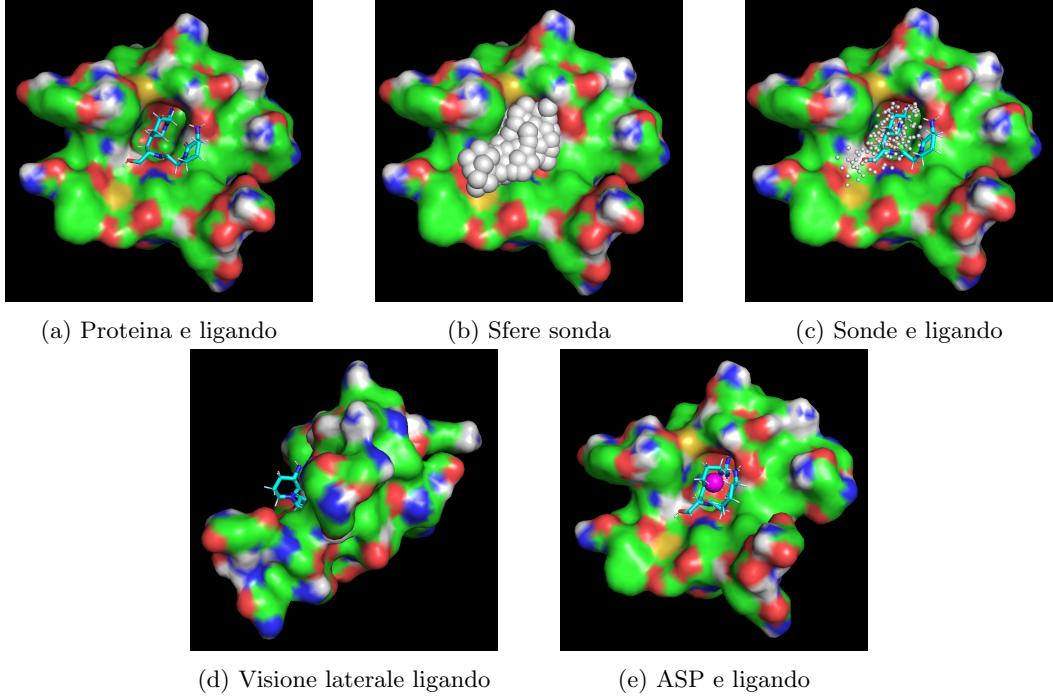


Figura 8: Risultati algoritmo PASS con proteina 1zzz

Nel seguente esempio, dove è stata utilizzata la proteina 1a0t, il ligando si trova completamente all'interno della cavità principale e dunque viene interamente coperto dalle sfere sonda individuate dall'algoritmo. Inoltre, gli ASP si trovano in prossimità del ligando.

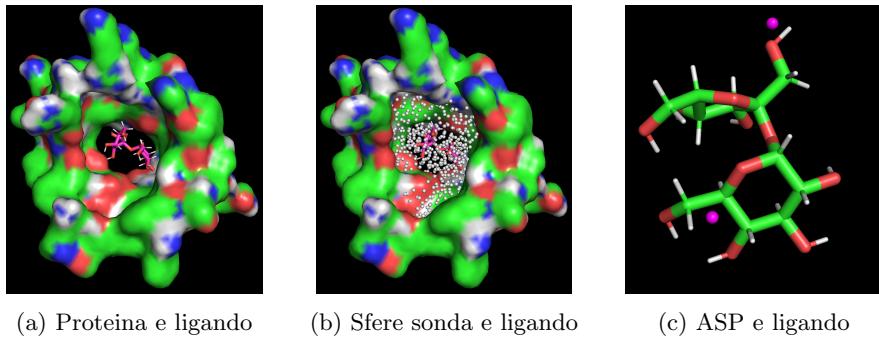


Figura 9: Risultati algoritmo PASS con proteina 1a0t

## 4.3 Confronto tra i due algoritmi

### 4.3.1 Confronto visivo

Entrambi gli algoritmi si basano su un metodo geometrico. PASS riesce ad individuare tutte le cavità sufficientemente evidenti. POCASA, dovendo dipendere dalla scelta del raggio della sfera di prova, può non individuare tutte le cavità presenti. Infatti, un raggio troppo piccolo può portare alla non identificazione di cavità nelle quali una sfera di quel raggio può facilmente entrare. Nell'esempio seguente, che riguarda la proteina 1a0t, si può notare questa caratteristica appena descritta. I risultati per PASS sono riportati nella figura 9, mentre i risultati di POCASA sono nella figura successiva. Mentre PASS trova la cavità principale, il risultato di POCASA dipende dal raggio scelto. Se si sceglie un raggio di 3 $\text{\AA}$ , la cavità centrale non viene individuata, mentre con un raggio di 4 $\text{\AA}$  questa viene trovata. Questo succede in quanto una sfera con raggio uguale o inferiore a 3 $\text{\AA}$  può essere posizionata senza sovrapporsi alla proteina all'interno della cavità.

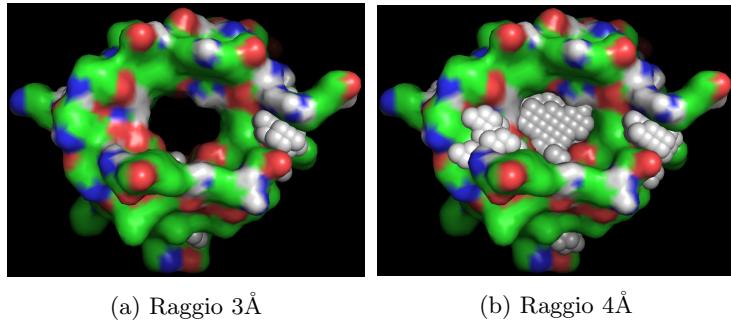


Figura 10: Confronto risultati di POCASA con diversi raggi sulla proteina 1a0t

Nelle seguenti immagini si possono confrontare i risultati ottenuti dai due algoritmi. Utilizzando PASS e POCASA con lato 0.5 $\text{\AA}$  si ottengono esiti simili, infatti entrambi identificano le stesse cavità con quasi la stessa estensione. Al contrario, POCASA con lato 1 $\text{\AA}$  trova le stesse cavità dei due precedenti, ma in maniera meno accentuata.

Si può concludere che, anche se in maniera più o meno accentuata, le cavità principali vengono identificate da entrambi gli algoritmi.

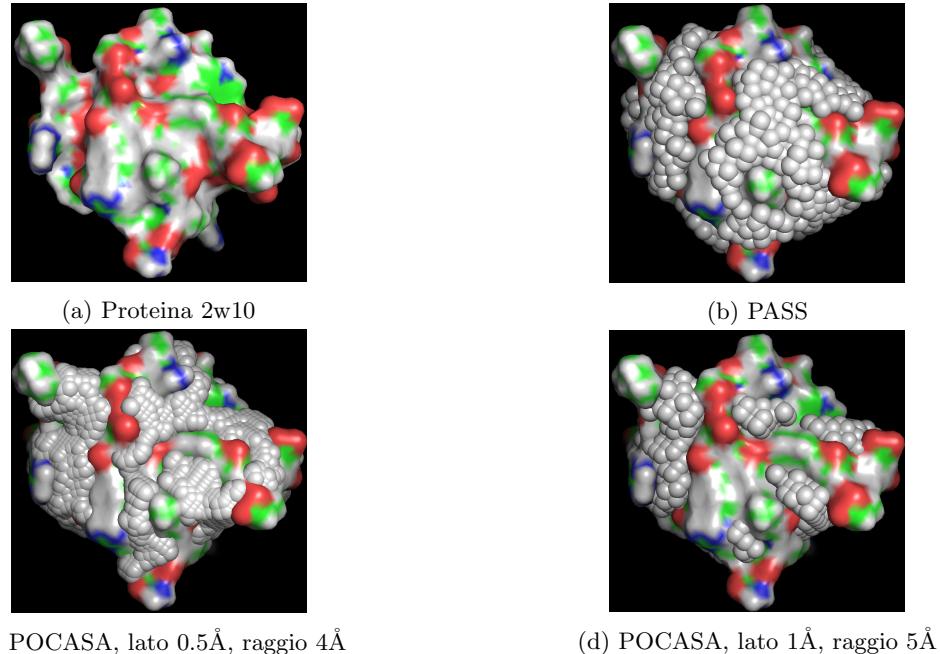


Figura 11: Confronto tra PASS e POCASA con proteina 2w10

### 4.3.2 Confronto algoritmico

Per ottenere una misura della differenza tra i risultati dei due metodi sono stati scritti due algoritmi. Il primo calcola la percentuale di sfere sonda, ottenute dal metodo di PASS, che sono sovrapposte al risultato di POCASA e viceversa. Il secondo calcola, per ciascun metodo, la percentuale di ligando coperta dalle sfere trovate.

A questo link è possibile trovare una tabella con il confronto tra i due metodi: <https://github.com/viols-code/ingegneria-informatica-project/blob/master/Comparison/Results.md>

Per quanto riguarda la copertura delle cavità, il risultato migliore si ottiene quando si confrontano i risultati di PASS con i risultati di POCASA con un raggio di 4Å. Infatti, come detto in precedenza, è consigliato utilizzare un raggio di 4Å se non si conosce la struttura della proteina. La cavità principale individuata da POCASA viene quasi interamente coperta da PASS, infatti per un raggio di POCASA di 3Å abbiamo una percentuale media di 71.47%, mentre per un raggio di 4Å abbiamo una media di 88.12%.

Per quanto riguarda la copertura del ligando le percentuali variano di molto. Se si visualizza, attraverso PYMOL, le casistiche in cui la copertura è molto bassa, si nota che il ligando si allontana dalla superficie della proteina e dunque, per i vari filtri presenti nei due metodi, non è possibile ottenere una copertura maggiore.

## 5 Conclusioni

L'obiettivo principale, ovvero quello di predire i siti di legame trovando le cavità delle proteine, è stato raggiunto da entrambi gli algoritmi, con alcune differenze. PASS individua non solo le cavità, ma anche i punti di siti attivi (ASP). Questa informazione aiuta a scegliere i ligandi da utilizzare e la loro posizione nella cavità. Inoltre, PASS non richiede all'utente di inserire alcun parametro e questo è un vantaggio, perché anche utenti con poca dimestichezza con l'algoritmo e il suo funzionamento sono in grado di utilizzarlo correttamente. POCASA, al contrario, richiede all'utente di prendere inizialmente delle decisioni che ne influenzano il risultato finale, e, come esposto precedentemente, la scelta del raggio può modificare sia il numero che il volume delle cavità individuate dal programma. D'altra parte POCASA ha un tempo di esecuzione inferiore rispetto a PASS avendo complessità  $O(N + dim * \log(dim))$  dove  $N$  è il numero di atomi della proteina e  $dim$  è la dimensione della griglia calcolata come  $(max_x - min_x) * (max_y - min_y) * (max_z - min_z)$  considerando  $max_i$  e  $min_i$  come i valori massimi e minimi per ogni coordinata. Mentre per PASS la complessità è  $O((N * M^3))$ , dove  $M$  dipende dal numero di sfere sonda individuate in ciascuna iterazione e ed è dunque difficile da determinare.

A questo link è possibile trovare l'eseguibile di entrambi gli algoritmi:

GitHub repository - <https://github.com/viols-code/ingegneria-informatica-project>.

A questo link è possibile trovare risultati ottenuti da entrambi gli algoritmi su diverse proteine:

Risultati - <https://www.dropbox.com/scl/fo/ng6dsajxyeu9fhtgvhu3z/h?dl=0&rlkey=1htrggqx7f678j39kagtmozz4>

## 6 Sitografia

- [1] *World Wide Protein Data Bank*, <https://www.wwpdb.org>
- [2] *RCSB PDB*, <https://www.rcsb.org>
- [3] *BioPandas*, <http://rasbt.github.io/biopandas/>
- [4] *POCASA: an automatic ligand-binding-site prediction program*, [http://altair.sci.hokudai.ac.jp/g6/Research/POCASA\\_e.html](http://altair.sci.hokudai.ac.jp/g6/Research/POCASA_e.html)
- [5] *Roll: a new algorithm for the detection of protein pockets and cavities with a rolling probe sphere*, <https://academic.oup.com/bioinformatics/article/26/1/46/181182>
- [6] G. Patrick Brady, Jr.\* and Pieter F.W. Stouten, *Fast Prediction and Visualization of Protein Binding Pockets with PASS*, [http://www.ccl.net/cca/software/UNIX/pass/pass\\_jcamd.html](http://www.ccl.net/cca/software/UNIX/pass/pass_jcamd.html)