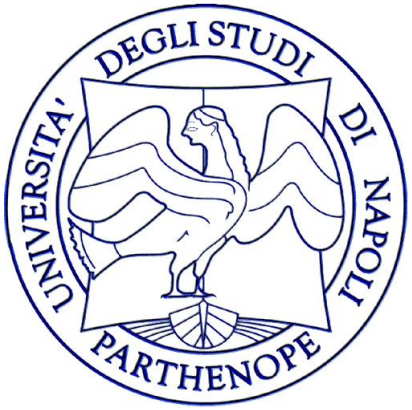
Universitá degli Studi di Napoli "Parthenope"
Scuola interdipartimentale delle Scienze, dell'Ingegneria e della Salute
Dipartimento di Scienze e Tecnologie
Corso di laurea in Informatica



Tesi di laurea triennale

Analisi delle interazioni molecolari e sviluppo di un software per l'automatizzazione del docking molecolare

Preparazione dei ligandi e dei recettori, docking ed estrazione dei legami

Relatore Ch.mo
Prof. Angelo Ciaramella

Correlatore
Dott. Ferdinando Febbraio

Laureando
Alfredo Mungari
Matr. 0124002134

Anno Accademico 2021-2022

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

Le api recano importanti benefici e servizi ecologici per la società. Con l'impollinazione le api svolgono una funzione strategica per la conservazione della flora, contribuendo al miglioramento ed al mantenimento della biodiversità.
In botanica, l'impollinazione è quel processo che consiste nel trasporto dei pollini dalla parte maschile e quella femminile dell'apparato riproduttivo delle piante. Grazie ad agenti atmosferici e sopratutto al lavoro incessante degli insetti impollinatori, soprattutto le api, il polline viene trasportato da una pianta all'altra rendendo possibile la fecondazione di un'essenza vegetale della stessa specie e la conseguente produzione di semi e frutti. Una diminuzione delle api può quindi rappresentare una importante minaccia per gli ecosistemi naturali in cui esse vivono. L'agricoltura, d'altro canto, ha un enorme interesse a mantenere le api quali efficaci agenti impollinatori. La Food and Agriculture Organization - FAO ha informato la comunità internazionale dell'allarmante riduzione a livello mondiale di insetti impollinatori, tra cui Apis mellifera, le api da miele. Circa l'84% delle specie di piante e l'80% della

produzione alimentare in Europa dipendono in larga misura dall'impollinazione ad opera delle api ed altri insetti pronubi [**bellucciapi**]. Pertanto, il valore economico del servizio di impollinazione offerto dalle api risulta fino a dieci volte maggiore rispetto al valore del miele prodotto.
Da un rapporto dell'Unione Internazionale per la Conservazione della Natura (LUCN) risulta che il 10% delle specie selvatiche di api (apis mellifera) sarebbe in via di estinzione e un altro 5% sarebbe a rischio. Una delle principali cause sono i pesticidi, i quali influenzano l'apprendimento, la capacità riproduttiva, i comportamenti sociali di questi insetti e l'orientamento.
La mortalità delle api (Apis mellifera) è un fenomeno che si acuisce soprattutto in primavera e che rischia di compromettere la fondamentale funzione ecologica di questi insetti impollinatori per l'intero ecosistema.
Un'indagine di campo del Centro di referenza nazionale per l'apicoltura dell'Istituto Zooprofilattico Sperimentale delle Venezie nell'ambito di alcune morie riscontrate ha rilevato la presenza, in campioni di api morte, di residui di pesticidi e di alcuni virus delle api. Le infezioni virali potrebbero peggiorare l'impatto già negativo dei pesticidi sulla salute delle api, mettendo ulteriormente in pericolo la sopravvivenza delle colonie. Lo studio è stato effettuato su 94 campioni, provenienti dal Nord-est dell'Italia e raccolti durante la primavera 2014, prendendo in considerazione 150 principi attivi e 3 virus delle api. Lo studio è pubblicato su Journal of Apicultural Research. I ricercatori hanno riscontrato la presenza di almeno un princi-

pio attivo nel 72,2% dei campioni (api morte). Gli insetticidi sono i più abbondanti (59,4%), principalmente quelli appartenenti alla classe dei neonicotinoidi (41,8%), seguiti da fungicidi (40,6%) e acaricidi (24,1%). Gli insetticidi più frequentemente rilevati sono rappresentati da imidacloprid, chlorpyrifos, tau-fluvalinate e cyprodinil.
La presenza di una possibile relazione tra la mortalità primaverile delle api e l'impiego di trattamenti antiparassitari in agricoltura potrebbe contribuire a comprendere meglio fenomeni complessi come la moria delle api e lo spopolamento degli alveari, che negli ultimi dieci anni hanno colpito questo settore [**martinello2017spring**].
Lo scopo della presente tesi è quello di illustrare la progettazione di un software che visualizza come le molecole di specifici pesticidi si dispongono, in maniera spaziale, quando sono legate ai recettori delle api, questo processo viene definito **docking molecolare**, e successivamente il software estrae i legami che si vengono a formare.

Figura (1.1): Esemplare adulto di Apis Mellifera

1.1 Docking Molecolare

Il docking molecolare è una tecnica computazionale che mira a determinare le migliori conformazioni adottate da una molecola per legarsi ad un'altra al fine di formare un complesso stabile. Il docking molecolare viene fondamentalmente utilizzato per la valutazione delle interazioni ligando-target. A partire quindi da un ligando e dalla struttura nota del suo target, il docking molecolare permette di generare una serie di conformazioni possibili del ligando stesso localizzato all'interno del sito attivo della proteina. Esse sono denominate *"binding poses"*e sono valutate da particolari funzioni chiamate *"scoring functions"*, che creano quindi un vero e proprio ranking. Le *migliori poses* rappresentano quella che viene identificata come la miglior soluzione proposta dall'algoritmo per l'interazione tra il ligando e il target[**meng2011molecular**].
Gli algoritmi di docking sono formati da due componenti fondamentali: l'algoritmo di ricerca (o *"search algorithm"*) e la *"scoring function"*. Il primo si occupa di generare un insieme di 12 conformazioni del ligando all'interno del sito designato del target, mentre la seconda valuta le *poses* generate, assegnando a ciascuna di esse un punteggio (detto *"score"*) in base a parametri di tipo geometrico ed energetico. Le migliori conformazioni in uscita da questa valutazione sono passate nuovamente all'algoritmo di ricerca, che andrà a creare una nuova generazione di conformazioni partendo dalle migliori soluzioni della run precedente. Il funzionamento iterativo del *search algorithm* e della *scoring function* permettono di otte-

nere, alla fine di un determinato numero di cicli, un insieme di *poses* che vengono fornite come output all'utente e che sono ritenute essere le migliori soluzioni per il *binding* delle molecole in esame da parte del programma di docking utilizzato.
In generale, il docking molecolare è eseguibile in tre differenti condizioni, che si differenziano l'una dall'altra per i gradi di libertà tenuti in considerazione dall'algoritmo durante il calcolo:

- docking a corpo rigido, che approssima sia il ligando che la proteina come strutture rigide
- docking semi-flessibile, che considera il target come rigido, tendendo però in considerazione i gradi di libertà conformazionale del ligando
- docking flessibile, in cui vengono considerati i gradi di libertà sia del ligando che dei residui del target nel sito attivo.

Intuitivamente, passando da un approccio a corpo rigido fino ad uno flessibile, la complessità di calcolo aumenta, e, proporzionalmente, anche il tempo di esecuzione.
Ad oggi sono disponibili diversi protocolli di docking, e ognuno sfrutta una particolare coppia algoritmo di *ricerca-scoring function*.
Il docking molecolare consiste in tre obiettivi principali collegati tra loro: predizione della posa, screening virtuale e stima dell'affinità di legame. Una metodologia di docking di successo deve essere in grado di prevedere correttamente la posa nativa del ligando all'interno del sito di legame del recettore

(cioè di trovare la geometria sperimentale del ligando entro un certo limite di tolleranza) e le interazioni fisico-chimiche molecolari associate. Inoltre, quando si analizzano librerie di composti di grandi dimensioni, il metodo deve essere in grado di distinguere con successo le molecole che si legano da quelle che non si legano e di classificare correttamente questi ligandi tra i migliori composti del database. Un algoritmo di ricerca e una funzione di score energetico sono gli strumenti di base di una metodologia di docking per generare e valutare le conformazioni dei ligandi. La capacità di gestire con successo la flessibilità molecolare intrinseca di un sistema e di descrivere correttamente l'energia delle interazioni recettore-ligando è cruciale per lo sviluppo di metodologie di docking predittivo che sono utili negli studi prospettici di progettazione di farmaci [**guedes2014receptor**].

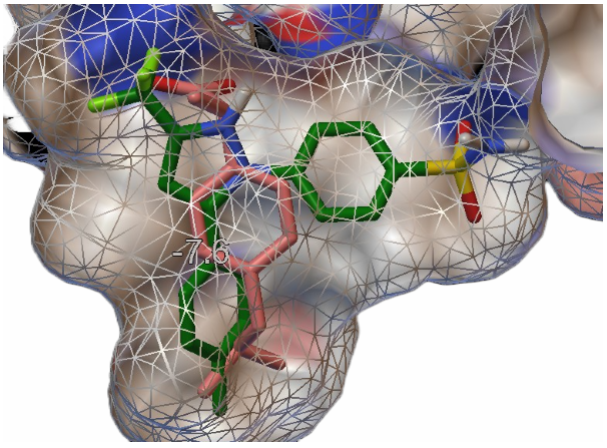


Figura (1.2): Rappresentazione dei due ligandi ibuprofene a sinistra e celecoxib a destra che hanno effettuato il docking con un enzima di COX2

1.2 Simulazione

La simulazione di un processo di docking è un processo molto più che complicato. In tale approccio, la proteina e il ligando sono separati fisicamente da una certa distanza, e il ligando trova la sua posizione nel sito attivo della proteina dopo aver compiuto diversi movimenti nello spazio. Tali movimenti includono rotazioni, traslazioni e torsione di alcuni angoli di rotazione degli atomi. Ognuno di questi movimenti ha un determinato costo energetico nel sistema, quindi dopo ogni mossa viene ricalcolata l'energia totale del sistema. Questo approccio modellizza molto precisamente quello che accade nella realtà. Di contro, il costo richiesto in termini di tempo e prestazioni è molto elevato.

1.3 Software per il docking molecolare

I programmi di docking molecolare eseguono un algoritmo di ricerca in cui la conformazione del ligando viene valutata ricorsivamente fino a raggiungere la convergenza all'energia minima. Infine, una funzione di punteggio di affinità, $\triangle$G (Energia potenziale totale in kcal/mol), viene impiegata per classificare le pose candidate come la somma delle energie elettrostatiche e di van der Waals. Le forze trainanti per queste specifiche interazioni nei sistemi biologici mirano alla complementarità tra la forma e l'elettrostatica delle superfici del sito di legame e del ligando o del substrato.
Negli ultimi vent'anni, sono stati sviluppati più di 60 diversi

strumenti e programmi di docking sia per uso accademico e commerciali, come DOCK (Venkatachalam et al. 2003) AutoDock (Österberg et al. 2002), FlexX (Rarey et al. 1996), Surflex (Jain 2003), GOLD (Jones et al. 1997), ICM (Schapira et al. 2003), Glide (Friesner et al. 2004), Cdocker, LigandFit (Venkatachalam et al. 2003), MCDock, FRED (McGann et al. 2003), MOE-Dock (Corbeil et al. 2012), LeDock (Zhao e Caflisch 2013), AutoDock Vina (Trott e Olson 2010), rDock (Ruiz-Carmona et al. 2014), UCSF Dock (Allen et al. 2015) e molti altri.

Tra questi programmi, AutoDock Vina, GOLD e MOE-Dock hanno predetto le pose migliori con gli score migliori. AutoDock e LeDock sono stati in grado di identificare i corretti legami dei ligandi nelle pose. Sia Glide (XP) che AutoDock hanno previsto le pose con un'accuratezza del 90,0% (Wang et al. 2016). È stato dimostrato che AutoDock ha prodotto fattori di arricchimento più rispetto a Glide in uno studio di screening virtuale contro il Fattore Xa, mentre Glide ha superato AutoDock contro lo stesso bersaglio in un analogo studio di screening virtuale. Nel complesso, è stato riportato recentemente che questi programmi di docking sono in grado di predire pose sperimentali con deviazioni al quadrato della radice (RMSD) in media (RMSD) in media da 1,5 a 2 Å [**pagadala2017software**].

Come mostrato nella Figura 1.3, il software di docking molecolare può aiutarci a individuare la conformazione e l'orientamento ottimali in base alla complementarità e alla preorganizzazione con un algoritmo specifico, quindi ad applicare

una funzione di scoring per prevedere l'affinità del legame e ad analizzare la modalità interattiva [**fan2019progress**].

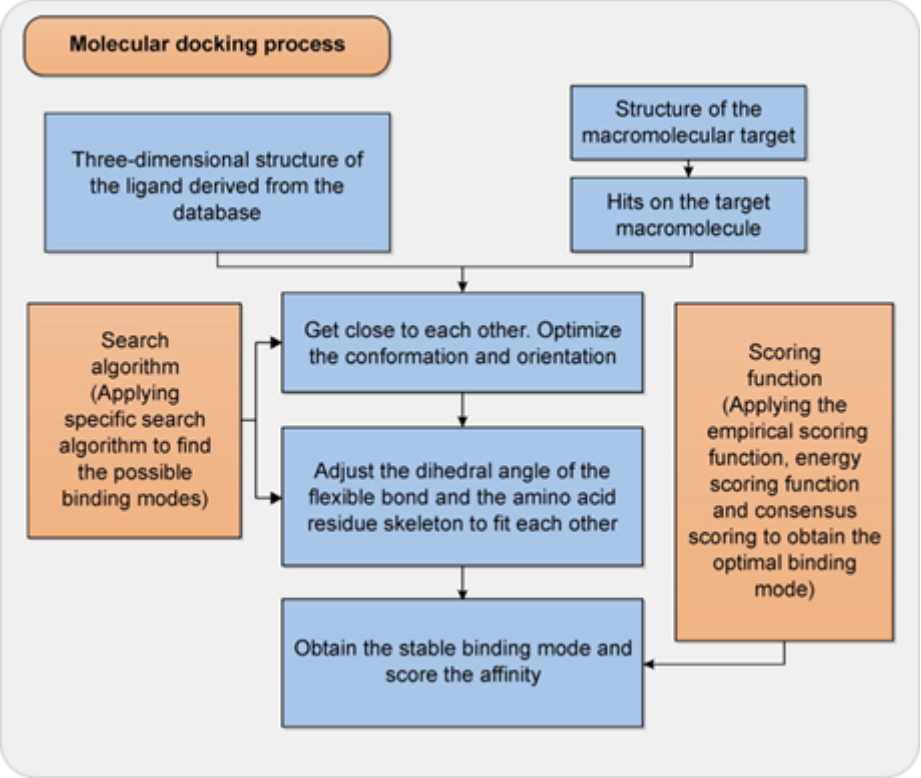


Figura (1.3): Processo del docking molecolare

1.4 Idea e sviluppo

L'**idea** nasce dall'attività di tirocinio svolta presso il "Centro nazionale di ricerca" di Napoli, per un totale di 300 ore (12 CFU), sotto la supervisione del Responsabile del laboratorio professore Angelo Ciaramella e del dottore Ferdinando Febbraio del CNR di Napoli. Il lavoro effettuato è consistito nella realizzazione di un software, del tutto preliminare al progetto di tesi proposto. Lo **sviluppo** è avvenuto attraverso diverse

fasi nelle quali sono stati utilizzati ed implementati i seguenti tools:

- software per l'esecuzione del docking
- funzioni di bioinformatica per la preparazione degli input necessari
- software per l'analisi dei risultati dell'intero processo.

Sono state determinate le componenti software ideali per automatizzare il processo di docking conseguendo risultati efficienti per quanto riguarda l'output e l'analisi dello stesso, offrendo una buona usabilità del prodotto realizzato mediante una semplice ed intuitiva interfaccia grafica.

1.5 Contenuto della tesi

La tesi è divisa in tre moduli:

- nella prima parte verranno discusse le tecnologie, le piattaforme scelte per la realizzazione del software, i linguaggi di programmazione e gli strumenti di bioinformatica utilizzati
- nella seconda verrà illustrata l'applicazione realizzata, dalla preparazione dei ligandi e recettori, passando per il docking, finendo con l'estrazione dei legami dall'output ottenuto
- nell'ultima parte verranno tratte le conclusioni e saranno indicati gli sviluppi futuri del software realizzato.

Capitolo 2

Tecnologie e piattaforme

In questo capitolo verranno trattate le tecnologie utilizzate, a partire dai linguaggi, i tools utilizzate e le piattaforme hardware di lavoro. È stato fondamentale l'utilizzo di macchine in remoto per eseguire le operazioni di training del modello. In questo caso le macchine messe a disposizione dall'Università Parthenope e dal CNR sono state fondamentali in quanto hanno garantito stabilità ed efficienza.

2.1 Linguaggi e tools

La scelta del linguaggio di programmazione è stata importante in quanto ci sono tantissimi linguaggi con librerie adatte allo scopo, solo pochi ricevono costante supporto e sono anche utilizzati nel mondo del lavoro.

2.1.1 Python

Python è un linguaggio di programmazione dinamico orientato agli oggetti utilizzabile per molti tipi di sviluppo software.

Offre un forte supporto all'integrazione con altri linguaggi e programmi, è fornito di una estesa libreria standard e può essere imparato in pochi giorni. Python inoltre è fornito delle librerie di bioinformatica adatte allo scopo del software realizzato. Di seguito le principali utilizzate:

- **pubchempy**, per le ricerche chimiche per nome, sottostruttura e somiglianza, standardizzazione chimica, conversione tra formati di file chimici, rappresentazione e recupero delle proprietà chimiche
- **prody**, per l'analisi della dinamica strutturale delle proteine
- **pandas**, per la manipolazione ed analisi dei dati
- **customTkinter**, per la realizzazione dell'interfaccia grafica
- **MolKit**, pacchetto che fornisce classi per leggere le molecole in diversi formati di file (PDB, Mol2...) e costruire una struttura gerarchica ad albero che riproduce la struttura interna della molecola
- **Matplotlib**, per la creazione di visualizzazioni statiche, animate e interattive
- **customTkinter**, per la realizzazione dell'interfaccia grafica
- **numpy**, per la gestione delle strutture dati
- **os**, per la gestione dei files

La versione di Python utilizzata è la **2.7** in quanto maggiormente compatibile con le librerie e i pacchetti utilizzati.

2.1.2 ADFRsuite

AutoDockFR (o **ADFR** in breve) è un programma per il **docking tra proteine e ligandi** sviluppato nel laboratorio Sanner dello Scripps Research sotto l'egida di *AutoDock*.
Utilizza:

- la funzione di scoring di *AutoDock4*, implementata in una libreria C++ ed inglobata in Python
- un proprio algoritmo genetico in grado di evolvere e ottimizzare più soluzioni simultaneamente, di gestire un gran numero di legami ruotabili e di terminare le iterazioni al momento della convergenza, cioè potenzialmente senza esaurire il numero assegnato di valutazioni della funzione energia
- una rappresentazione generica di flessibilità molecolare, chiamata *Albero di Flessibilità*, che consente di incorporare una varietà di movimenti molecolari sia nel ligando che nel recettore.

Pur supportando le modalità di docking disponibili in *AutoDock4* e *Vina*, è stato progettato specificamente per includere la **flessibilità del recettore** e supporta anche il **docking covalente**. Il suo algoritmo genetico personalizzato consente il docking di ligandi con più legami ruotabili rispetto ad *AutoDock4*.

Viene distribuito come parte della suite del software *ADFR*, che fornisce strumenti aggiuntivi per facilitare il docking automatizzato.
ADFR implementa diverse caratteristiche che aiutano a semplificare la procedura di docking e a supportare la gestione e la riproducibilità degli esperimenti di docking attraverso la prova dei dati. Vengono utilizzati file autodocumentati per memorizzare:

- la rappresentazione del sito di binding (cioè **i file .trg target**)
- i risultati del docking (file **.dro Docking Results Object**)
- I metadati memorizzati in questi file non solo supportano la riproducibilità, ma riducono anche i rischi di errori dell'operatore.

ADFR legge i **ligandi** preparati per il docking con AutoDock, cioè nel formato PDBQT e organizza la flessibilità del ligando in base ai legami ruotabili. Un file PDBQT può essere generato da un file .pdb di un ligando utilizzando il comando *prepar_ligand*. Il **recettore** è specificato come file target, cioè un singolo file che descrive il recettore. I file target possono essere calcolati per un recettore nel formato PDBQT dal comando utilizzando il programma **agfr** o l'interfaccia grafica **agfrgui**. Un file PDBQT può essere generato da un file pdb di un recettore utilizzando il comando *prepare_receptor* della suite *ADFR*.

ADFR è implementato nei moderni linguaggi di programmazione orientati agli oggetti e si basa su componenti software riutilizzabili. I componenti critici per le prestazioni sono implementati in C e C++ (ad esempio ADFRcc), mentre altri sono implementati in Python (ADFR, AutoSite, MolKit2, prody). *ADFR* è rilasciato sotto la licenza open source LGPL v2. In particolare per l'applicazione trattata sono stati utilizzati i due script: **prepare_ligand4** e **prepare_receptors4**, rispettivamente per la preparazione dei ligandi e dei recettori e per effettuare la loro conversione dal formato .pdb al formato .pdbqt necessario per la procedura di docking.

2.1.3 MGLTools

La suite software **MGLTools** è stata sviluppata nel laboratorio Sanner presso il Center for Computational Structural Biology (CCRB) precedentemente noto come Molecular Graphics Laboratory (MGL) dello Scripps Research Institute per la visualizzazione e l'analisi delle strutture molecolari. MGLTools comprende:

- **Python Molecular Viewer (PMV)**, un visualizzatore molecolare di uso generale
- **AutoDockTools (ADT)**, un insieme di comandi PMV sviluppati specificamente per supportare gli utenti di AutoDock
- **Vision**, un ambiente di programmazione visuale.

Questi strumenti software sono altamente integrati e basati su componenti software riutilizzabili implementati in Python e C++ (con binding Python). Il kit di strumenti grafici sottostante è Tk (Tkinter).L'ultima versione di MGLtools è la 1.5.7 che fornisce:

- un widget della dashboard ridisegnato
- un widget per la visualizzazione delle sequenze
- ottimizzazioni delle prestazioni
- nuovi comandi per i calcoli di sovrapposizione e RMSD

2.1.4 Open Babel

Open Babel è un tool per applicazioni di chimica progettato per interpretare i molteplici formati dei dati chimici e per cercare, convertire, analizzare o archiviare dati da modellistica molecolare, chimica, materiali a stato solido, biochimica o aree correlate.
La versione di OpenBabel 2.3 converte fino a 110 formati di file chimici.
I database sono ampiamente utilizzati per memorizzare le informazioni chimiche soprattutto nell'industria farmaceutica. Un requisito fondamentale di un database di questo tipo è la capacità di indicizzare le strutture strutture chimiche in modo che possano essere recuperate rapidamente, data una quary di ricerca. Open Babel offre questa funzionalità utilizzando un'indicizzazione basata sul percorso. Questa indicizzazione, FP2 in Open Babel, identifica tutte le sottostrutture lineari

e ad anello della molecola di lunghezza da 1 a 7 (escluse le sottostrutture a 1 atomo C e N) e le mappa in una stringa di bit di lunghezza su una stringa di bit di lunghezza 1024 utilizzando una funzione di hash. Se una molecola interrogata è una sottostruttura di una molecola di destinazione, tutti i bit molecola di destinazione, allora tutti i bit impostati nella molecola di query saranno impostati anche nella molecola di destinazione. Le indicizzazioni di due molecole possono anche essere utilizzate per calcolare la somiglianza strutturale utilizzando il coefficiente di Tanimoto, il numero di bit in comune diviso per tutti i bit dell'insieme. Chiaramente, la ricerca iterativa dello stesso insieme di molecole comporterà l'uso ripetuto dello stesso insieme di indicizzazioni. Per evitare la necessità di ricalcolare le indicizzazioni per un particolare file multi-molecola (come un file SDF), Open Babel fornisce un formato fastindex che memorizza esclusivamente un'indicizzazione insieme a un indice nel file originale. Questo indice porta a un rapido aumento della velocità di ricerca di corrispondenze a a fronte di una quary: insiemi di dati con diversi milioni di molecole sono facilmente consultabili in modo interattivo. In questo in questo modo, un file multi-molecola può essere utilizzato come un'alternativa efficace a un sistema di database chimico [**o2011open**].

2.2 Database

PubChem è il più grande database al mondo di informazioni chimiche liberamente accessibili, attraverso il quale è possibile cercare le sostanze chimiche per nome, formula molecolare, struttura e altri identificatori. Inoltre è possibile trovare proprietà chimiche e fisiche, attività biologiche, informazioni sulla sicurezza e sulla tossicità, brevetti, citazioni bibliografiche e altro ancora. L'interfaccia tra l'applicazione ed il database è realizzata mediante la libreria di Python **pubchempy**.
PubChem è un database di molecole chimiche, gestito dal centro nazionale per l'Informazione biotecnologica statunitense (NCBI), parte della biblioteca nazionale di medicina (NLM) dell'istituto nazionale della sanità americano (NIH). L'accesso al database PubChem può essere eseguito liberamente attraverso un sito web e possono essere scaricati dati riguardanti milioni di strutture di composti e dati descrittivi tramite il protocollo FTP.PubChem possiede descrizioni di molecole con meno di 1000 atomi e 1000 legami.
PubChem gestisce i dati in tre database interconnessi: **Substance, Compound e BioAssay**. Il database Substance archivia le descrizioni delle sostanze chimiche fornite dai depositanti. Il database Compound archivia le strutture chimiche uniche estratte dal database Substance attraverso la standardizzazione delle strutture. Il database BioAssay contiene la descrizione e i risultati degli esperimenti di analisi biologica.

2.3 Autodock

AutoDock è un programma di docking che utilizza un algoritmo genetico, il *Lamarckian Genetic Algorithm*, per il calcolo della pose migliore che interagisce con il sito attivo della proteina. Dopo aver calcolato inizialmente una popolazione di possibili soluzioni, l'algoritmo ne selezionerà una parte in base alle funzioni di scoring e darà origine a una nuova popolazione di soluzioni figlie, da cui avrà inizio un secondo ciclo di generazione e così via. In questo modo il "genotipo", ovvero la stringa binaria a cui corrisponde ciascun ligando, verrà influenzato da fattori esterni, esattamente come nell'ipotesi lamarckiana.
Le popolazioni di soluzioni sono ottenute tramite operatori genetici (mutazioni, crossover e migrazioni) che imitano quelli biologici. I gradi di libertà sono codificati in geni o stringhe binarie, e a geni e cromosomi è assegnato un valore basato sulla fitness della scoring function. Le operazioni di mutazione causano cambiamenti nel valore di un gene, mentre il crossover muove un set di geni da un cromosoma "genitore" ad un altro; la migrazione invece muove singoli geni da una sottopopolazione ad un'altra.
L'interazione tra ligando e recettore è valutata in due fasi, calcolando la variazione di energia intramolecolare del passaggio dalla forma libera a quella legata e la variazione di energia libera intermolecolare implicata nello stesso passaggio.
Per effettuare il docking è necessario per prima cosa preparare le coordinate di ligando e recettore. La preparazione delle

coordinate è la fase più importante nella procedura, poiché in esse sono inclusi parametri fondamentali come: idrogeni polari, atom – type e cariche parziali. Le coordinate del ligando originale e della macromolecola sono trattate separatamente ed i loro file sono in un formato particolare, il PDBQT.

2.4 Autodock Vina

AutoDock Vina è un nuovo programma per il docking molecolare e lo screening virtuale. Vina rappresenta la nuova versione di AutoDock, ed infatti presenta molte similitudini con il suo predecessore ma, allo stesso tempo, anche molte differenze. Una differenza importante consiste nella velocità di calcolo, dato che Vina è molto poco dispendioso sotto questo punto di vista; altra differenza fondamentale è rappresentata dal fatto che al momento del calcolo della griglia, Vina calcola internamente ed automaticamente le "grid maps" impiegate in AutoDock. Questo costituisce un grande vantaggio in termini di facilità e velocità di esecuzione. Inoltre, le funzioni di scoring e gli algoritmi utilizzati in questo tipo di analisi risultano essere completamente diversi rispetto al suo predecessore, cosa che porta a considerare Vina quasi come un software a sé stante. Vina migliora al tempo stesso in modo significativo l'accuratezza delle previsioni delle modalità di legame. Un'ulteriore miglioramento è ottenuta grazie al parallelismo, che utilizza il multithreading su macchine multicore. AutoDock Vina calcola automaticamente le mappe della griglia e raggruppa i risultati in modo trasparente per l'utente. Auto-

dock vina all'interno del software prende in input i ligandi ed i recettori in formato .pdbqt.

2.4.1 Funzione di scoring

La funzione di scoring di AutoDock Vina (qui indicata come Vina) può essere rappresentata attraverso la seguente formula:

$$c = \sum_{i<j} f_{t_i t_j}(r_{ij}) \tag{2.1}$$

dove la sommatoria è su tutte le coppie di atomi che possono muoversi l'uno rispetto all'altro, escludendo normalmente le interazioni 1-4, ovvero gli atomi separati da tre legami covalenti consecutivi. Ad ogni atomo i viene assegnato un tipo t_i e un insieme simmetrico di funzioni di interazione $f_{t_i t_j}$ della distanza interatomica r_{ij} da definire.
Questo valore può essere visto come una somma di contributi intermolecolari e intramolecolari:

$$c = c_{inter} + c_{intra} \tag{2.2}$$

L'algoritmo di ottimizzazione, descritto in seguito, cerca di trovare il minimo globale di c e di altre conformazioni a basso punteggio, che poi classifica.
L'energia libera di legame prevista viene calcolata a partire dalla parte intermolecolare della conformazione con il punteggio più basso, designata come:

$$s_1 = g(c_1 - c_{intra1}) = g(c_{inter1}) \tag{2.3}$$

dove la funzione g può essere una funzione arbitraria strettamente crescente possibilmente non lineare.
Nell'output le altre conformazioni a basso punteggio vengono formalmente restituite dai valori di s, ma per preservare il ranking, si utilizza c_{intra} come migliore modalità di legame:

$$s_i = g(c_i - c_{intra1}) \tag{2.4}$$

Per ragioni di modularità, gran parte del programma non fa riferimento ad alcuna forma funzionale delle interazioni $f_{t_i t_j}$ o g. Essenzialmente, queste funzioni vengono passate come parametro per il resto del codice. Inoltre, il programma è stato progettato in modo tale da poter utilizzare schemi di tipizzazione degli atomi come la tipizzazione degli atomi di AutoDock4 o SYBIL.

Pesi	Termini
-0.0356	$gauss_1$
-0.00516	$gauss_2$
0.840	$repulsion$
-0.0351	$hydrophobic$
-0.587	$hydrogenbonding$
0.0585	N_{rot}

Tabella (2.1): Funzione di scoring pesi e termini

La particolare implementazione della funzione di scoring che verrà presentata è stata ispirata principalmente da X-score e come tale è stato messo a punto utilizzando PDBbind. Tuttavia, alcuni termini sono diversi da X-score e, nel mettere a punto la funzione di scoring, si è andati oltre la regressione lineare. Inoltre, va notato che Vina classifica le conformazioni secondo l'eq. (2.2) o, equivalentemente, eq. (2.4), mentre X-score conta solo i contributi intermolecolari. Per quanto ne sappiamo, X-score non è stato implementato in un programma di docking, ignorare i vincoli interni potrebbe portare l'algoritmo di ottimizzazione a ricercare strutture corrotte all'interno. La derivazione della nostra funzione di scoring combina alcuni vantaggi tra quelli potenzialmente conosciuti e le funzioni di scoring empiriche: estrae informazioni empiriche da entrambe le preferenze conformazionali del sia dei complessi recettore-ligando sia dalle misure sperimentali affini.
Lo schema di tipizzazione degli atomi segue quello di X-score. Gli atomi di idrogeno non sono considerati esplicitamente, se

non per la tipizzazione degli atomi, e sono omessi dall'eq. (2.1).
Le funzioni di interazione $f_{t_i t_j}$ sono definite rispetto alla distanza di superficie $d_{ij} = r_{ij} - R_{ti} - Rtj$:

$$f_{t_i t_j}(r_{ij}) \equiv h_{t_i t_j}(d_{ij}) \tag{2.5}$$

dove R_t è il raggio di van der Waals dell'atomo di tipo t.
Nella nostra funzione di scoring, $h_{t_i t_j}$ è una somma ponderata di interazioni steriche (i primi tre termini nella tabella 2.1), identica per tutte le coppie di atomi, interazione idrofobiche tra atomi idrofobici e, dove possibile, legami a idrogeno. I pesi sono mostrati nella tabella 2.1. I termini sterici sono i seguenti:

$$gauss_1(d) = e^{-(d/0.5\mathring{A})^2} \tag{2.6}$$

$$gauss_2(d) = e^{-((d/0.5\mathring{A})/2\mathring{A})^2} \tag{2.7}$$

$$repulsion(d) = \begin{cases} d^2, se & d < 0 \\ 0, se & d \geq 0 \end{cases} \tag{2.8}$$

Il termine idrofobico è uguale a 1, quando $d < 0,5\mathring{A}$; 0, quando $d > 1,5\mathring{A}$, ed è interpolato linearmente tra questi valori. Il termine di legame a idrogeno è uguale a 1, quando $d < -0,7\mathring{A}$; 0, quando $d < 0,5\mathring{A}$, e viene interpolato linearmente tra questi valori. Seguendo Xscore, trattiamo formalmente i metalli come donatori di legami a idrogeno. In questa in questa implementazione, tutte le funzioni di interazione $f_{t_i t_j}$ sono tagliate a $r_{ij} = 8\mathring{A}$.

La Figura 1 mostra i termini sterici ponderati da soli o combinati con i termini idrofobici o H con le interazioni idrofobiche o di legame H[**trott2010autodock**].

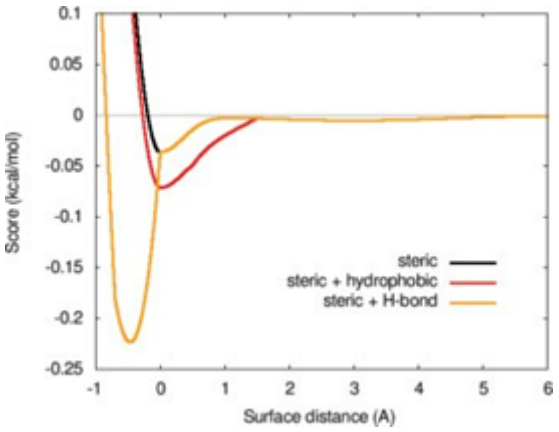


Figura (2.1): Funzione di scoring pesata

Capitolo 3

Applicazione realizzata

Il progetto di tesi proposto ha come focus principale la realizzazione del docking tra i ligandi contenuti in specifici pesticidi e i recettori dell'apis mellifera e l'estrazione dei legami che si vengono a formare. Il software realizzato può essere utilizzato in due modalità: mediante script python da terminale o mediante interfaccia grafica.

3.1 Dati in input

I dati in input all'applicazione trattata sono: ligandi e proteine. La lista dei ligandi è fornita in input tramite foglio calcolo (.xlsx, .xls) o mediante file di testo (.txt), in entrambi i casi ogni riga corrisponde al nome di un ligando. Essendo l'applicazione incentrata sullo studio degli effetti dei ligandi dei pesticidi sull'apis mellifera, come dati di esempio sono state utilizzati i ligandi le cui molecole costituiscono i pesticidi maggiormente diffusi sul mercato per un totale di 297 ligandi. La lista è disponibile nell'appendice **??**.
I recettori vengono selezionati mediante una web view aperta

sulla pagina di ricerca del sito di **PubChem**, l'utente digiterà la propria query e dopo aver selezionato il tasto *research* gli verranno mostrati tutti i composti organici relativi alla query digitata, tramite il tasto *Get Query* l'utente andrà a scaricare tutti i file dei composti organici in formato .pdb. Nel caso di esempio sono state scelte tutte le proteine dei recettori dell'Apis Mellifera come mostrato nelle foto: **??** e **??**, per un totale di 60 file .pdb contenenti le strutture di determinate proteine.

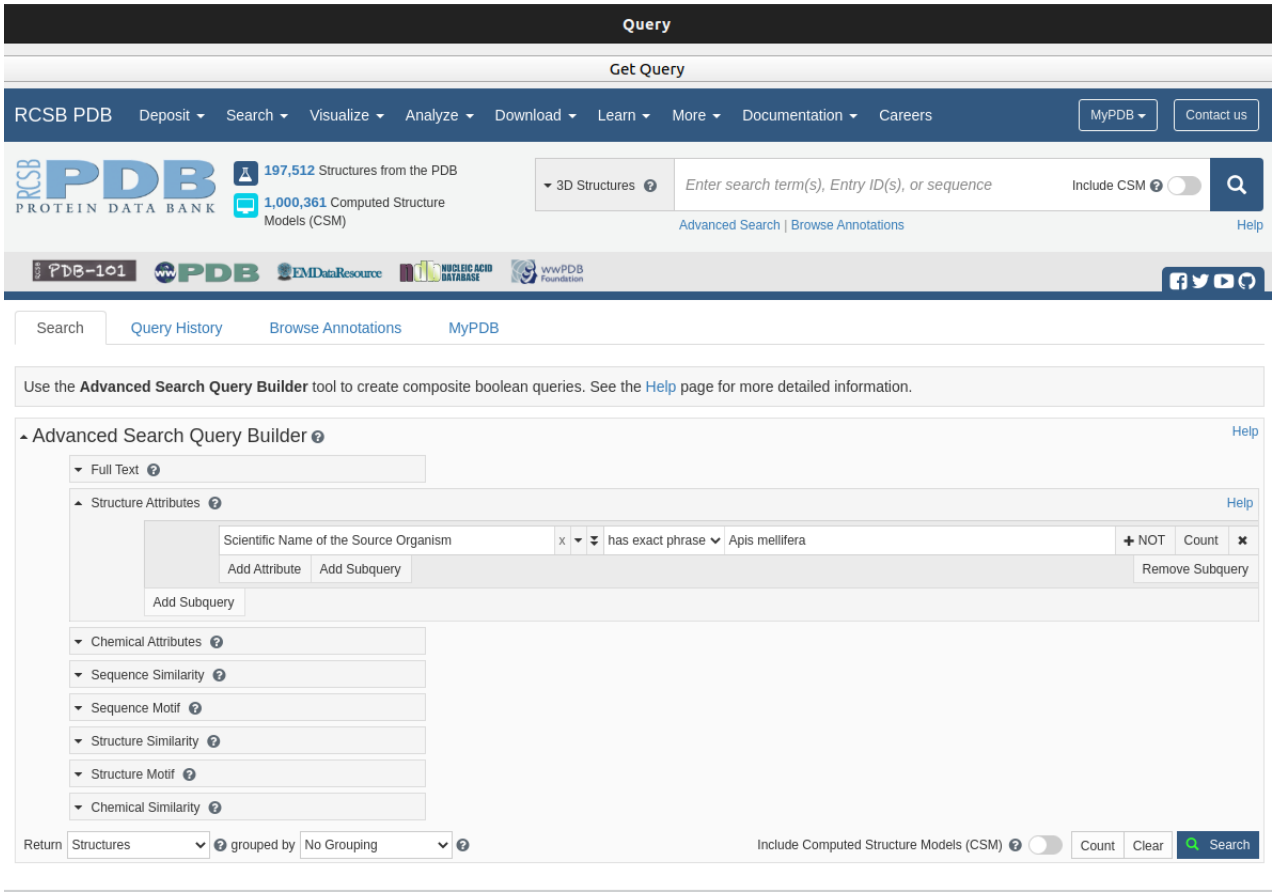


Figura (3.1): Query dei recettori

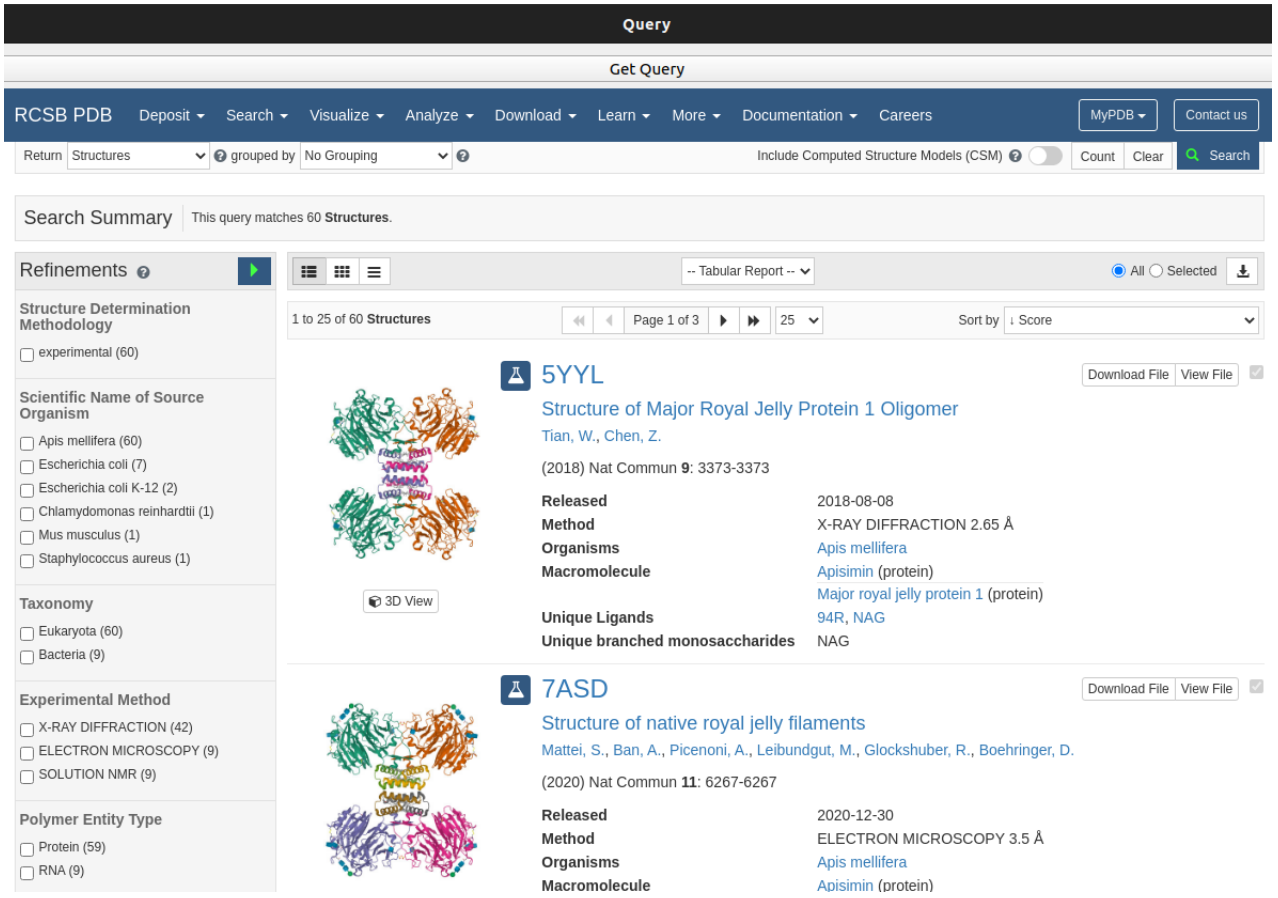


Figura (3.2): Proteine dei recettori dell'apis mellifera

3.2 Preparazione dei ligandi e dei recettori

Il primo step propedeutico per il docking è la preparazione dei ligandi e dei recettori, questa fase viene esplicitamente eseguita dal software realizzato.

3.2.1 Preparazione dei ligandi

La lista di ligandi da scaricare è fornita in input mediante un file di testo (.txt) alla nostra applicazione o mediante foglio di

calcolo (.xlsx, .xls), all'interno della lista sono presenti i nomi dei ligandi uno sotto all'altro. Tramite la funzione

Conclusioni

In questo capitolo vanno le conclusioni del progetto.
Esempio:
In questo lavoro è stato introdotto . . .

Appendice A

Tabella dei ligandi

Tabella (A.1): Tabella dei ligandi in input

Ligando	Ligando
(E,E)-7,9-Dodecadien-1-yl acetate	Gibberellins
(E,E)-8,10-Dodecadien-1-ol	Glyphosate
(3E,8Z,11Z)-Tetradeca-3,8,11-trienyl acetate	Halauxifen-methyl
3E,8Z-Tetradecadienyl acetate	Halosulfuron-methyl
(E)-11-Tetradecen-1-yl acetate	Hexythiazox
(E)-5-Decen-1-ol	Hymexazol
(E)-5-Decen-1-yl acetate	Imazalil
(E)-8-Dodecen-1-yl acetate	Imazamox
(Z,E)-7,11-Hexadecadien-1-yl acetate	Indolylbutyric acid
(Z,E)-Tetradeca-9,11-dienyl acetate	Indoxacarb
(Z,E)-9,12-Tetradecadien-1-yl acetate	Iodosulfuron
(Z)-11-Hexadecen-1-ol	Ipconazole
(Z)-11-Hexadecen-1-yl acetate	Iprovalicarb

Continua nella pagina successiva

Tabella A.1

Ligando	Ligando
(Z)-11-Hexadecenal	Isofetamid
(Z)-11-Tetradecen-1-yl acetate	Isopyrazam
(Z)-13-Octadecenal	Isoxaben
(Z)-7-Tetradecenal	Isoxaflutole
(Z)-8-Dodecen-1-ol	Kresoxim-methyl
(Z)-8-Dodecen-1-yl acetate	L-Ascorbic acid
(Z)-8-Tetradecen-1-ol	lambda-Cyhalothrin
z-8-Tetradecenyl acetate	Laminarin
(Z)-9-Dodecen-1-yl acetate	Lauric acid
(Z)-9-Hexadecenal	Lavandulyl senecioate
(Z)-9-Tetradecen-1-yl acetate	Lenacil
1-Decanol	Malathion
1-methylcyclopropene	Maleic hydrazide
1-Naphthylacetamide	Mandestrobin
1-Naphthylacetic acid	Mandipropamid
1,4-Dimethylnaphthalene	MCPA
2-Phenylphenol	MCPB
2,4-D	Mecoprop-P
Methyl 2,5-dichlorobenzoate	Mefentrifluconazole
24-Epibrassinolide	Mepanipyrim
4-(2,4-Dichlorophenoxy)butanoic acid	Mepiquat
6-Benzyladenine	Meptyldinocap
8-Hydroxyquinoline	Mesosulfuron

Continua nella pagina successiva

Tabella A.1

Ligando	Ligando
Acequinocyl	Mesotrione
Acetamiprid	Metaflumizone
Acetic acid	Metalaxyl
Acibenzolar-S-methyl	Metalaxyl-M
Aclonifen	Metaldehyde
Acrinathrin	Metam
Ametoctradin	Metamitron
Amidosulfuron	Metazachlor
Aminopyralid	Metconazole
Amisulbrom	Methoxyfenozide
Azimsulfuron	Methyl decanoate
Azoxystrobin	Methyl octanoate
Beflubutamid	Zineb
Benalaxyl-M	Metobromuron
Benfluralin	Metrafenone
Bensulfuron	Metribuzin
Bentazone	Metsulfuron-methyl
Benthiavalicarb	Milbemectin
Benzoic acid	Tetradecyl acetate
Benzovindiflupyr	Napropamide
Bifenazate	Nicosulfuron
Bifenox	Oleic acid
Bispyribac	Orange oil

Continua nella pagina successiva

Tabella A.1

Ligando	Ligando
Bixafen	Oxamyl
Boscalid	Oxathiapiprolin
Bromuconazole	Oxyfluorfen
Bupirimate	Paclobutrazol
Buprofezin	Pelargonic acid
Capric acid	Penconazole
Caprylic acid	Pendimethalin
Captan	Penflufen
Carfentrazone-ethyl	Penoxsulam
Carvone	Penthiopyrad
Chlorantraniliprole	Pethoxamid
Chlormequat	Phenmedipham
Chlorotoluron	Phosmet
Chromafenozide	Phosphane
Clethodim	Picloram
Clodinafop	Picolinafen
Clofentezine	Pinoxaden
Clomazone	Pirimicarb
Clopyralid	Pirimiphos-methyl
Cyantraniliprole	Potassium bicarbonate
Cyazofamid	Prochloraz
Cycloxydim	Prohexadione
Cyflufenamid	Propamocarb

Continua nella pagina successiva

Tabella A.1

Ligando	Ligando
Cyflumetofen	Propaquizafop
Cyhalofop-butyl	Propoxycarbazone
Cymoxanil	Propyzamide
Cypermethrin	Proquinazid
Cyprodinil	Prosulfocarb
Daminozide	Prosulfuron
Dazomet	Prothioconazole
Deltamethrin	Pyraclostrobin
Dicamba	Pyraflufen-ethyl
Dichlorprop-P	Pyridaben
Diclofop	Pyridalyl
Difenoconazole	Pyridate
Diflufenican	Pyrimethanil
Dimethachlor	Pyriofenone
Dimethenamid-P	Pyriproxyfen
Dimethomorph	Pyroxsulam
Dimoxystrobin	Quinmerac
Dithianon	Quizalofop-P
Dodecan-1-ol	Quizalofop-P-ethyl
Dodecyl acetate	Quizalofop-P-tefuryl
Dodemorph	Rescalure
Dodine	Rimsulfuron
Ethephon	S-Metolachlor

Continua nella pagina successiva

Tabella A.1

Ligando	Ligando
Ethofumesate	Sedaxane
Etofenprox	Sodium 2-methoxy-5-nitrophenolate
Etoxazole	Sodium 2-nitrophenolate
Eugenol	Sodium 4-nitrophenolate
Fenazaquin	Spiromesifen
Fenhexamid	Spirotetramat
Fenoxaprop-P	Spiroxamine
Fenpicoxamid	Sulcotrione
Fenpropidin	Sulfosulfuron
Fenpyrazamine	Sulfoxaflor
Fenpyroximate	Sulfuryl fluoride
Flazasulfuron	tau-Fluvalinate
Flonicamid	Tebuconazole
Florasulam	Tebufenozide
Florpyrauxifen-benzyl	Tebufenpyrad
Fluazifop-P	Tefluthrin
Fluazinam	Tembotrione
Flubendiamide	Terbuthylazine
Fludioxonil	Tetraconazole
Flufenacet	Tetradecan-1-ol
Flumetralin	Thiabendazole
Flumioxazin	Thiencarbazone-methyl
Fluometuron	Thifensulfuron-methyl

Continua nella pagina successiva

Tabella A.1

Ligando	Ligando
Fluopicolide	Thymol
Fluopyram	Tolclofos-methyl
Fluoxastrobin	Tri-allate
Flupyradifurone	Tribenuron
Fluquinconazole	Triclopyr
Flurochloridone	Trifloxystrobin
Fluroxypyr	Triflusulfuron
Flutianil	Trinexapac
Flutolanil	Triticonazole
Fluxapyroxad	Tritosulfuron
Folpet	Urea
Foramsulfuron	Valifenalate
Forchlorfenuron	Ziram
Formetanate	Zoxamide
Fosetyl	Quartz sand
Fosthiazate	Silthiofam
Gamma-cyhalothrin	Esfenvalerate
Garlic extract	Ethylene
Geraniol	Gibberellic acid
(2Z,4E)-5-[(1S)-1-Hydroxy-2,6,6-trimethyl-4-oxocyclohex-2-en-1-yl]-3-methylpenta-2,4-dienoic acid	
1-(4-Chlorophenyl)-5-(2-methoxyethoxy)-4-oxo-1,4-dihydrocinnoline-3-carboxylic acid	

Appendice B

Tabella dei recettori

5YYL	3BFA	3D78	6LQK
7ASD	3BFB	3FE6	6O4M
1BH1	3BFH	3FE8	7OXF
1CCV	3BJH	3FE9	7ZS6
1FCQ	3CAB	3R72	2J88
1FCU	3CDN	3RZS	3QRX
1FCV	3CYZ	3S0A	6GXN
1POC	3CZ0	3S0B	6GXP
1TER	3CZ1	3S0D	4E81
1TUJ	3CZ2	3S0E	6GXO
2H8V	3D73	3S0F	6GWT
2LIC	3D74	3S0G	6YST
2MLT	3D75	5OHX	6YSS
2MW6	3D76	5XZ3	6YSU
2N8V	3D77	6DST	5O2R