UNIVERSITÀ DI PAVIA

Dipartimento di Ingegneria Industriale e dell'Informazione
Corso di Laurea Magistrale in Bioingegneria

Integrating Molecular Dynamics Simulations with Machine Learning and Deep Learning to Predict Nanobody Binding Modes

Supervisori

C.mo Prof. Luca F. Pavarino

C.mo Prof. Giorgio Colombo

Co-supervisori

Gauthier Trèves

Ivan Cucchi

Candidato

Mattia Lai

Matricola

534749

Anno Accademico 2024/2025

Abstract

Lo sviluppo di nuove molecole per applicazioni terapeutiche e diagnostiche rappresenta una delle sfide centrali della ricerca traslazionale moderna. Tuttavia, indicatori di produttività e rischio restano sfavorevoli: secondo le analisi pubblicate su *Nature Reviews Drug Discovery*, la probabilità che un candidato in Fase I giunga all'approvazione è intorno al 9–10% [1, 2]. In questo contesto, i *nanobodies*, piccoli domini anticorpali leganti l'antigene derivati dalla regione variabile delle immunoglobuline a sola catena pesante di alcuni mammiferi, e oggetto di studio del lavoro, hanno suscitato un crescente interesse quale piattaforma biotecnologica ad alto potenziale.

Nel processo di *drug discovery*, l'integrazione tra approcci sperimentali e computazionali è divenuta imprescindibile per ridurre tempi, costi e incertezze associate ai soli saggi di laboratorio; malgrado i progressi, i metodi *in silico* presentano ancora limiti, in particolare nella caratterizzazione delle modalità di legame antigene–anticorpo. Per affrontare queste criticità, un approccio promettente per colmare questo *gap* consiste nell'adottare *framework* di apprendimento supervisionato per sviluppare modelli *physics-informed*, anche grazie alla disponibilità di calcolo parallelo su GPU [3].

Sulla base di questa idea, il presente lavoro di ricerca propone di integrare simulazioni di dinamica molecolare, una tecnica computazionale impiegata per simulare il movimento di atomi e molecole nel tempo, insieme a modelli di *supervised learning* per prevedere il *binding* dei *nanobodies*. Questa integrazione è sviluppata in condizioni *target-free*, ossia senza formulare assunzioni né sul sito di legame né sulla natura dell'antigene, così da poter estendere le evidenze a tutti i potenziali *partner* molecolari. In questo contesto vengono presi in esame due meccanismi di riconoscimento fondamentali: quello ortosterico, in cui il *nanobody* si lega direttamente al sito funzionale dell'antigene bloccandone meccanicamente l'attività, e quello allosterico, in cui il *nanobody* si lega a un sito secondario, modificandone indirettamente la forma e disattivandolo.

Nella prima parte della tesi, sono state condotte simulazioni di dinamica molecolare a partire da una selezione di strutture cristallografiche di alta qualità. Successivamente, dalle traiettorie ottenute sono stati estratti descrittori fisico-chimici che costituiranno la base del *set* di *features* impiegato dai classificatori. Infine, diversi modelli fra *machine learning* e *deep learning* sono stati addestrati con l'obiettivo di ottimizzare il *trade-off* tra capacità predittiva ed efficienza computazionale.

Nella seconda parte, una volta che è stato completato l'addestramento dei classificatori, è stata eseguita un'analisi dell'importanza delle *features* con lo scopo di inferire le ragioni strutturali, a livello molecolare, che contraddistinguono un *nanobody* di classe ortosterica da uno allosterico. Questo passaggio è centrale nel lavoro di ricerca perché consente di andare oltre il mero compito predittivo, trasformando il *machine learning* da strumento di classificazione a *driver* metodologico per l'indagine delle proprietà molecolari, capace di far emergere *insight* da una metodologia *physics-oriented* altrimenti difficilmente accessibili con i soli approcci computazionali *non data-driven*.

Indice

Elenco delle figure

Elenco delle tabelle

1 Introduzione

Nella prima parte del Capitolo, saranno illustrati i fondamenti dell'immunità adattiva, a partire dai meccanismi di generazione della diversità anticorpale e dall'organizzazione strutturale degli anticorpi. Un'attenzione specifica sarà dedicata alla differenza tra le immunoglobuline *standard* e gli *Heavy-Chain Antibodies* di derivazione camelide, e in particolare ai *nanobodies*, che costituiscono l'oggetto di indagine del presente lavoro di tesi.

Nella seconda parte, il Capitolo introduce il caso studio applicativo scelto, per cui viene analizzato il sistema dei complessi *nanobodies*-proteina *Spike* di SARS-CoV-2, con particolare riferimento alla classificazione binaria dei paratopi adottata per i fini computazionali.

Nella terza parte sono proposte alcune applicazioni dei *nanobodies* in ambito biotecnologico.

Infine, la quarta e ultima parte presenta una sintesi degli approcci computazionali precedentemente trattati dal gruppo di ricerca che ha preso parte al progetto, i cui esiti hanno costituito la base di partenza del lavoro.

1.1 Aspetti biologici e molecolari degli anticorpi e dei *nanobodies*

1.1.1 Sistema immunitario adattivo

Il sistema immunitario adattativo, noto anche come immunità specifica o acquisita, rappresenta la seconda linea di difesa dell'organismo, più sofisticata e specializzata rispetto all'immunità innata. A differenza di quest'ultima, che è immediata ma aspecifica, l'immunità adattativa richiede più tempo per svilupparsi, generalmente alcuni giorni, ma possiede due caratteristiche fondamentali: la specificità e la memoria immunologica. La specificità consente di riconoscere e dirigere la risposta contro specifiche molecole estranee, chiamate antigeni.

La seconda caratteristica è la memoria immunologica. La prima esposizione a un antigene genera una risposta lenta, ma permette al sistema di ricordare il patogeno. Le esposizioni successive provocano quindi una risposta più rapida ed efficiente. Questo principio è alla base dello sviluppo dei vaccini.

Tali proprietà emergono dall'azione coordinata dei linfociti. I Linfociti B sono responsabili dell'immunità umorale, mediata da anticorpi (o immunoglobuline, Ig) che neutralizzano patogeni e tossine nei fluidi corporei, che riconoscono gli antigeni tramite il recettore BCR. I Linfociti T, invece, sono responsabili dell'immunità cellulo-mediata.

La capacità di riconoscere un vasto bacino di antigeni non è deterministica, ma viene generata casualmente durante lo sviluppo dei linfociti negli organi linfoidi primari, quali midollo osseo per i linfociti B e timo per i linfociti T. Qui, ogni linfocita produce un recettore unico (BCR o TCR) attraverso una ricombinazione genica casuale. La totalità dei recettori possibili costituisce quindi il repertorio recettoriale [4].

1.1.2 Anticorpi convenzionali

Per quanto concerne la trattazione del sistema immunitario adattativo umorale, è fondamentale analizzare le molecole effettrici che ne mediano la funzione, ovvero le immunoglobuline. Queste glicoproteine, prodotte e secrete in forma solubile dai linfociti B, differenziatisi in plasmacellulle in seguito ad attivazione antigenica, costituiscono la parte esecutiva primaria dell'immunità umorale. La loro funzione primaria è il riconoscimento altamente specifico degli antigeni, e, di conseguenza, la mediazione di una serie di eventi effettori finalizzati alla neutralizzazione e all'eliminazione della minaccia patogena. La morfologia generale delle immunoglobuline è assimilabile a una forma di "Y" (Figura 1.1), configurazione che riflette la loro duplice funzionalità: i due bracci superiori della molecola (frammenti Fab) sono deputati al riconoscimento dell'antigene, mentre la porzione inferiore, denominata frammento Fc, funge da interfaccia per il reclutamento di componenti cellulari e umorali del sistema immunitario innato [4].

Dal punto di vista strutturale, la forma canonica di un'immunoglobulina è un eterotetramero, composto da due identiche catene polipeptidiche pesanti (*heavy chains*, H) e due identiche catene leggere (*light chains*, L). Sia le catene pesanti che quelle leggere sono caratterizzate dalla presenza di domini strutturali ripetuti, ciascuno dei quali adotta un ripiegamento conservato noto come *immunoglobulin fold*. Questo ripiegamento consiste in una struttura compatta formata da due foglietti β antiparalleli disposti in una topologia a *greek key*[1], che conferisce una notevole stabilità alla molecola. Le regioni N-terminali di ciascuna catena pesante e leggera formano i domini variabili (VH e VL, rispettivamente), la cui conformazione tridimensionale definisce il sito di legame per l'antigene, detto anche paratopo. All'interno di questi domini variabili, sono presenti tre brevi segmenti ipervariabili, denominati regioni determinanti la complementarità (*Complementarity-Determining Regions*, CDR1, CDR2 e CDR3). I *loop* delle CDR, che sporgono dallo scheletro strutturale conservato (regioni *framework*), sono direttamente responsabili del contatto fisico con l'epitopo antigenico e rappresentano il principale determinante della specificità e dell'affinità di legame dell'anticorpo.

Il resto della molecola è costituito da domini costanti: le catene leggere possiedono un singolo dominio costante (CL), mentre le catene pesanti ne possiedono tre (CH1, CH2, CH3) nei comuni isotipi.

L'assemblaggio dell'eterotetramero è stabilizzato da ponti disolfuro, sia tra le due catene pesanti che tra ciascuna catena pesante e la sua corrispettiva catena leggera.

I due bracci Fab sono connessi allo stelo Fc attraverso la regione della cerniera (*hinge*), un segmento peptidico situato tra i domini CH1 e CH2 delle catene pesanti, ricco in residui di prolina e cisteina, la cui lunghezza e composizione aminoacidica variano significativamente tra le diverse sottoclassi di Ig [4]. L'isotipo predominante nel siero è l'immunoglobulina G (IgG), che rappresenta il modello strutturale di riferimento e la base per la maggior parte degli anticorpi monoclonali terapeutici [5].

[1] Il motivo strutturale a *greek key* è un arrangiamento topologico di quattro filamenti β antiparalleli, collegati da anse, che formano un β-*sandwich* caratteristico del dominio immunoglobulinico.

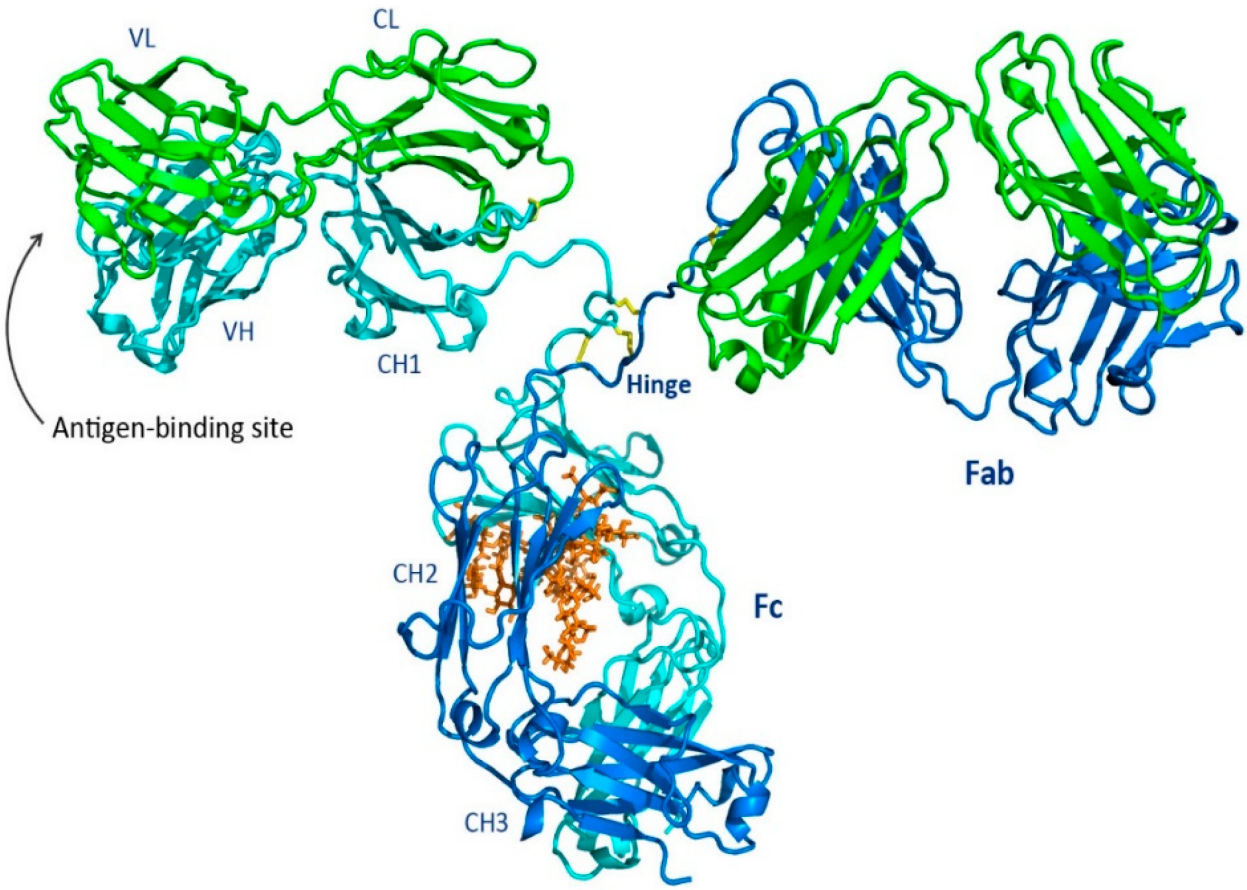

Figura 1.1: Vista tridimensionale di un'IgG intera (PDB ID:1igt) che mette in evidenza i moduli funzionali: i due bracci Fab (domini variabili VH/VL seguiti da CH1/CL; sede del paratopo e quindi del riconoscimento dell'epitopo), lo stelo Fc (domini CH2–CH3; sede dell'ingaggio delle funzioni effettrici) e la breve regione di *hinge* che collega Fab e Fc e funge da giunto flessibile. Immagine riprodotta e adattata da Chiu et al., Antibodies 2019, Fig. 1; licenza CC BY 4.0 [6].

1.1.3 *Nanobodies* di origine camelide

Un aspetto peculiare dell'immunologia dei camelidi (che includono cammelli, dromedari, lama e alpaca) è la coesistenza, accanto al repertorio convenzionale di immunoglobuline tetrameriche, di una classe distinta di anticorpi funzionali privi di catene leggere. Queste immunoglobuline, denominate anticorpi a catena pesante (*Heavy-Chain Antibodies*, HCAbs), presentano una architettura molecolare radicalmente semplificata, essendo costituite esclusivamente da dimeri di catene pesanti che mancano completamente del dominio costante CH1 e, di conseguenza, dell'intera catena leggera associata. Il dominio variabile di questi anticorpi non convenzionali, designato con il termine V_HH (*Variable domain of Heavy chain antibody*), rappresenta la più piccola unità funzionale di riconoscimento antigenico, e costituisce il componente fondamentale dei cosiddetti *nanobodies* [7]. Questo concetto è riportato nelle Figure 1.2 e 1.3.

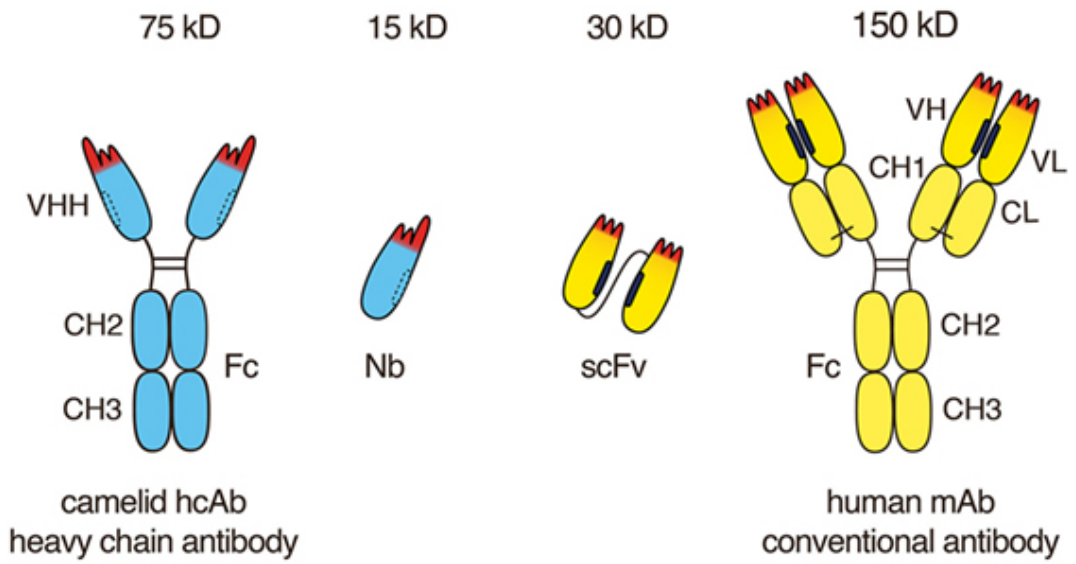

Figura 1.2: Confronto tra la struttura di un anticorpo a sola catena pesante (HCAb) tipico dei camelidi, riportato in blu, e quella di un anticorpo convenzionale (riportato in giallo). Nell'HCAb è assente il primo dominio costante (CH1) dell'omonima catena pesante e non è presente alcuna catena leggera. La funzione di riconoscimento antigenico è pertanto affidata interamente al singolo dominio variabile V_HH, o *nanobody*. Immagine riprodotta e adattata da Bannas et al., *Frontiers in Immunology* (2017), Fig. 1; licenza CC BY 4.0 [8].

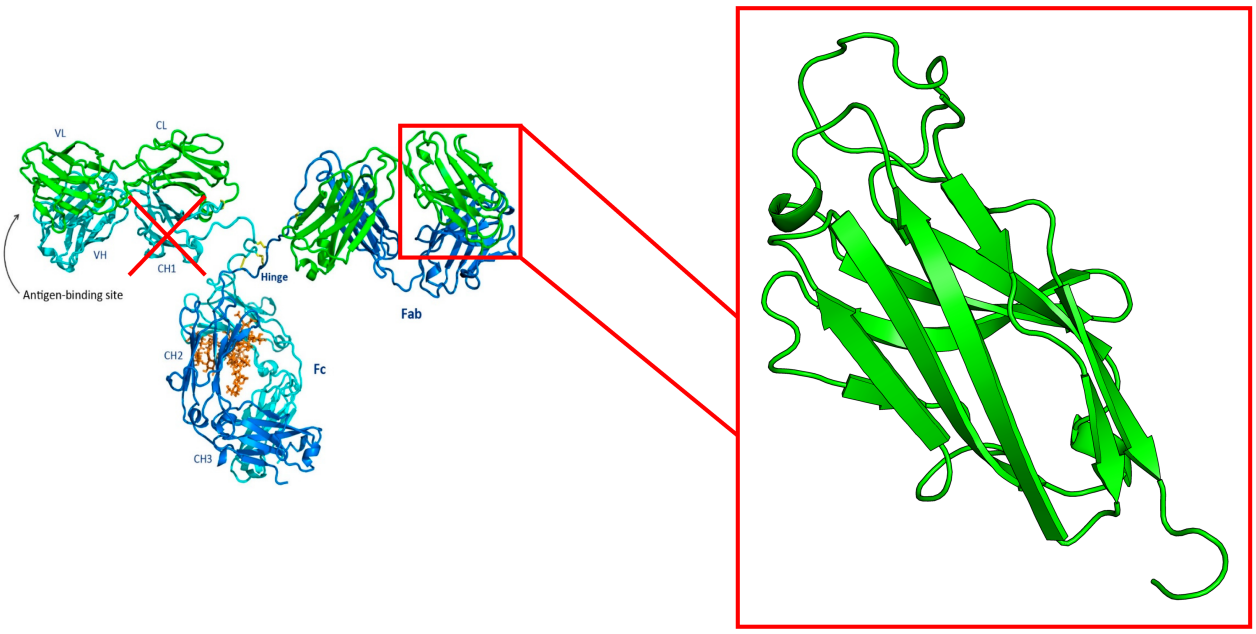

Figura 1.3: Rappresentazione *ribbon* che illustra, a sinistra, lo schema di un anticorpo convenzionale con indicazione (X rossa) del dominio CH1, assente nel repertorio anticorpale (a sola catena pesante) dei camelidi. Il riquadro di ingrandimento a destra mette a fuoco il singolo dominio variabile V_HH (*nanobody*), le cui regioni ipervariabili (CDR1–3) costituiscono il sito di legame per l'antigene.[2]

Librerie di *nanobodies* possono essere generate attraverso la diretta immunizzazione dell'animale tramite somministrazione di antigene in conformazione nativa, in presenza di adiuvante, così da indirizzare la risposta

[2]La rappresentazione *ribbon* (o *cartoon*) mostra la struttura proteica per evidenziare la struttura secondaria: le frecce indicano i foglietti β (con verso N→C), eventuali cilindri rappresentano le α-eliche e i tratti lisci sono i *loop*. Questo tipo di rappresentazione mette in risalto l'architettura a *sandwich* di foglietti β tipica dei domini immunoglobulinici, comuni nei *nanobodies*, e consente di apprezzare conformazione e lunghezza dei CDR.

verso epitopi conformazionali biologicamente rilevanti [9]. Il protocollo sperimentale prevede sei inoculazioni sottocutanee a cadenza settimanale, con 100–200 µg di antigene per dose; una settimana dopo l'ultima somministrazione si esegue il prelievo ematico (circa 100 mL nel lama).

Il dominio variabile di questi anticorpi (V_HH), sebbene mantenga la conformazione fondamentale (*immunoglobulin fold*), presenta una composizione aminoacidica differente dagli anticorpi convenzionali, compensando così l'assenza fisiologica del dominio variabile della catena leggera (VL). Dal punto di vista strutturale, questi domini V_HH mostrano la stessa architettura dei domini VH degli anticorpi, con quattro regioni *framework* conservate (FR 1/2/3/4) che circondano tre *loop* ipervariabili, chiamati regioni determinanti la complementarità (CDR 1/2/3). Le regioni *framework* (FR1, FR2, FR3 e FR4) costituiscono lo scheletro strutturale conservato del dominio immunoglobulinico, sul quale si innestano le regioni ipervariabili CDR.

Le proprietà fisico-chimiche dei V_HH sono differenti da quelle degli anticorpi convenzionali. Possiedono una massa molecolare particolarmente ridotta, pari a circa 15 kilodalton (kD)[3] , che conferisce loro una superiore capacità di diffusione tissutale rispetto agli anticorpi completi (una Ig completa è tipicamente 150 kD). Mostrano un'elevata solubilità intrinseca in mezzi acquosi e una significativa stabilità termodinamica, resistendo a temperature di denaturazione più elevate e a condizioni di pH estreme rispetto agli anticorpi convenzionali. Queste proprietà rendono i V_HH particolarmente adatti all'espressione in sistemi eterologhi, come batteri o lieviti, dove possono essere prodotti in grandi quantità con costi contenuti, senza necessitare dei complessi processi di *folding* e assemblaggio tipici degli anticorpi tetraeterici [10].

Nei *nanobodies*, il paratopo è costituito dalle CDR insieme a un numero limitato di residui dei *framework*; sebbene tutte e tre le CDR contribuiscano, il ruolo principale spetta alla CDR3, che è spesso più lunga dei *loop* delle comuni immunoglobuline, e adotta conformazioni capaci di penetrare nelle cavità degli antigeni. Ciò spiega perché gli epitopi bersaglio dei V_HH differiscono frequentemente da quelli riconosciuti dagli anticorpi convenzionali [11]. Questa conformazione risulta particolarmente adatta per legare epitopi criptici, come tasche attive enzimatiche o solchi di interazione in complessi proteici, siti spesso inaccessibili ai voluminosi paratopi formati dall'associazione VH-VL degli anticorpi convenzionali [7].

1.1.4 Meccanismi immunogenetici di diversificazione delle *Complementarity Determining Regions* (CDRs)

Sul piano immunogenetico, i camelidi esibiscono i medesimi meccanismi di diversificazione del repertorio anticorpale diversi da quelli osservati in altri mammiferi. Un tratto caratteristico risiede nell'organizzazione del *locus* genico delle catene pesanti. I camelidi possiedono un repertorio relativamente ampio di segmenti genici variabili (IGHVH) dedicati specificamente alla formazione degli HCAbs. Questi segmenti V (Variabili) subiscono processi di ricombinazione somatica con un insieme più ristretto di segmenti D (Diversità) e J (Giunzione), mediata dal complesso enzimatico RAG1/RAG2. La diversità giunzionale, che costituisce il principale determinante della regione ipervariabile CDR3, è amplificata da una maggiore frequenza di inserzioni (*insertions*) e delezioni (*deletions*), comunemente denominate *indel*, nei punti di giunzione tra i segmenti V, D e J [12].

1.2 Caso studio: analisi del complesso *nanobody* – proteina *spike* del virus SARS-CoV-2

Negli ultimi cinque anni, la ricerca e lo sviluppo biotecnologico dei *nanobodies* hanno ricevuto un impulso decisivo dalla pandemia di SARS-CoV-2, con la generazione di numerosi costrutti sia a fini diagnostici sia terapeutici. L'emergenza sanitaria ha agito da potente *driver* scientifico e tecnologico, accelerando lo sviluppo

[3] In termini quantitativi, $15\,\text{kDa} = 15{,}000\,\text{Da} \approx 15{,}000\,\text{g mol}^{-1}$. Ciò corrisponde a $\frac{15{,}000\,\text{g mol}^{-1}}{N_A} \approx 2{,}5\times10^{-20}\,\text{g} = 2{,}5\times10^{-23}\,\text{kg}$ per singola molecola; un dominio V_HH di tale massa contiene tipicamente $\sim 130-140$ amminoacidi (assumendo $\approx 110\,\text{Da}$ per residuo).

di piattaforme di selezione, ingegnerizzazione e caratterizzazione che hanno portato alla generazione di numerosi costrutti proteici. In tale contesto, la disponibilità di un bersaglio comune e altamente *standardizzato* (la proteina *Spike*) ha reso possibile una mappatura sistematica delle interazioni *nanobody*–epitopo e, soprattutto, il collegamento tra modalità di legame osservate a livello strutturale e conseguenze funzionali in termini di neutralizzazione virale.

1.2.1 Struttura e funzione della *spike*

La proteina *spike* (S) di SARS-CoV-2, che media l'ingresso del virus nelle cellule ospiti, è una glicoproteina di fusione di classe I che si organizza in un omotrimero. Ciascun trimero è composto da tre protomeri identici associati con simmetria di rotazione ternaria (per cui la struttura appare identica dopo una rotazione di 120° attorno a un proprio asse). Ogni protomero può essere scomposto in due subunità funzionalmente distinte, denominate S1 e S2, le quali svolgono ruoli sequenziali e distinti nel processo infettivo. La subunità S1 è preposta al riconoscimento e al legame con il recettore cellulare ACE2 (*Angiotensin-Converting Enzyme* 2)[4] [13]. All'interno della subunità S1 sono presenti due domini cruciali: il Dominio N-Terminale (NTD), la cui funzione può includere interazioni con corecettori, e il *Receptor-Binding Domain* (RBD), che contiene specificamente il sito di legame per ACE2 [14]. Il RBD è mobile e può alternare tra una conformazione *up* (o sollevata), ricettiva per il legame con il recettore, e una *down* (o abbassata), in cui l'interfaccia di legame è meno accessibile [15]. La porzione del RBD che contatta direttamente il recettore è definita *receptor binding site* (RBS). La subunità S2, invece, contiene il sistema molecolare necessario per la fusione delle membrane virale e cellulare [16].

1.2.2 Tassonomia dei *nanobodies*: implicazioni molecolari e computazionali

L'analisi strutturale di un ampio repertorio di complessi V_HH–RBD ha introdotto un criterio di classificazione esplicitamente centrata sui *nanobodies*, e sulle modalità di *binding* con l'epitopo. In tale contesto, lo studio fondante di Xiang et al. [17] cataloga i V_HH in cinque classi, definite da posizione geometrica, proprietà fisico–chimiche, rigidità strutturale e grado di conservazione evolutiva tra i sarbecovirus[5]. Infatti, a seguito di uno sforzo sperimentale notevole, in cui una vasta gamma di *nanobodies* leganti l'RBD sono stati prodotti e caratterizzati strutturalmente, gli Autori hanno raggruppato i *nanobodies* in 5 classi identificate sulla base dell'epitopo riconosciuto sull'RBD. Questa impostazione, stabilisce una relazione diretta tra struttura del sito di legame del *nanobody* e funzione biologica, con ricadute sulla potenza e sull'ampiezza dello spettro d'azione nei confronti delle diverse varianti virali.

I *nanobodies* di Classe I si legano specificamente al *Receptor-Binding Site* (RBS), neutralizzando il virus attraverso un meccanismo competitivo diretto che preclude fisicamente l'aggancio al recettore ACE2. Sebbene mostrino un'elevatissima potenza neutralizzante, la loro efficacia è spesso vulnerabile alle *escape mutations* che si accumulano in questa regione soggetta a intensa pressione selettiva.

Invece, *nanobodies* di Classe II riconoscono epitopi contigui ma non sovrapposti al RBS. Il loro meccanismo di neutralizzazione, di tipo indiretto, si basa prevalentemente sulla modulazione della dinamica conformazionale del trimero di *Spike*, stabilizzandolo energeticamente nella configurazione chiusa (con RBD in conformazione *down*) e riducendo così l'accessibilità del sito di legame per ACE2 [18]. Questo conferisce loro un profilo di resistenza alle varianti generalmente più favorevole.

La Figura 1.4 mostra chiaramente la differente regione di intereazione fra epitopo e paratopo.

Per le finalità applicative e computazionali di questo lavoro, la tassonomia in cinque classi viene ricondotta a una classificazione binaria. Questa scelta è motivata da precise esigenze biologiche e analitiche.

[4] ACE2 è una carbossipeptidasi di membrana espressa in vari epiteli (quali vie aeree, intestino, rene) e coinvolta nell'equilibrio del sistema renina–angiotensina; nel contesto dell'infezione da SARS-CoV-2 funge da recettore d'ingresso.

[5] Per *sarbecovirus* si intende il *pool* di betacoronavirus che include SARS-CoV, SARS-CoV-2 e virus affini isolati in diversi animali.

Dal punto di vista biologico, la dicotomia riassume il *trade-off* tra potenza neutralizzante e ampiezza del riconoscimento antigenico, che rappresenta la caratteristica distintiva più rilevante nella risposta anticorpale, e riassume primariamente lo scopo del lavoro, ovvero indagare le modalità di *binding* epitopo–paratopo.

Computazionalmente, invece, questa riduzione permette di costruire un *framework* analitico gestibile, essenziale per identificare con chiarezza i determinanti alla base dei meccanismi di riconoscimento, maggiormente distintivi e identificati nel contesto di una classificazione binaria rispetto ad una multiclasse.

La classificazione qui adottata distingue:

(i) Classe Ortosterica (RBS-competitiva), che comprende esclusivamente i *nanobodies* di Classe I, i quali legano il *Receptor-Binding Site* (RBS) in competizione diretta con il recettore ACE2.

(ii) Classe Allosterica, che raggruppa i *nanobodies* delle Classi II, i quali legano un epitopo non sovrapposto al RBD e più conservato. La classe II è stata scelta come rappresentativa del meccanismo allosterico rispetto alle altre classi (III, IV, V, che presentano comunque un meccanismo di legame allosterico) in quanto è la più rappresentata in termini di strutture cristallografiche depositate nel *Protein Data Bank* (PDB) [19].

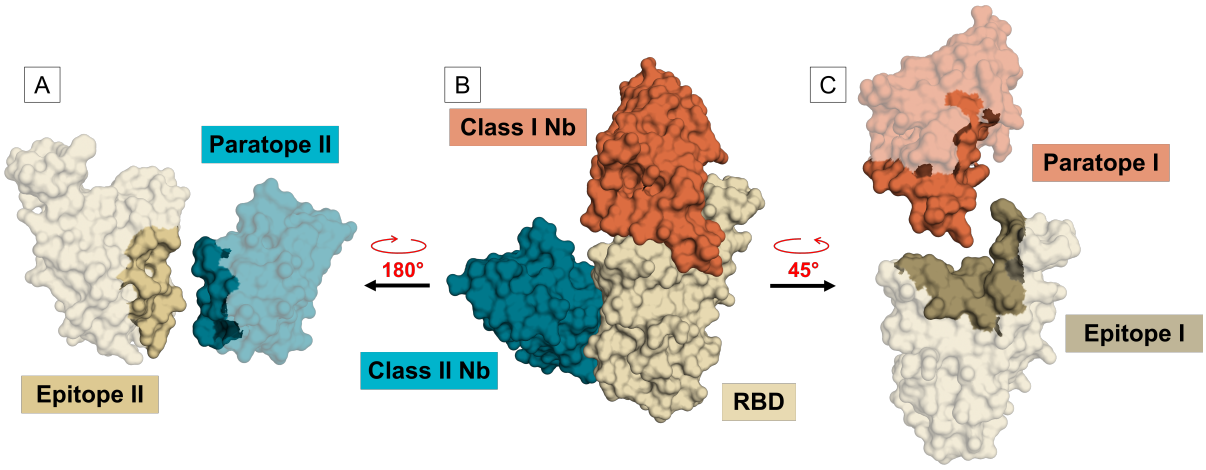

Figura 1.4: Struttura cristallografica del dominio RBD (in beige) in complesso con due *nanobodies* neutralizzanti (PDB ID: 7OLZ). Il *nanobody* di classe I Re5D06 e il *nanobody* di classe II Re9F06 sono rappresentati rispettivamente come superfici arancione e azzurro; nei pannelli A e C la tonalità più scura evidenzia i paratopi sui *nanobodies* e gli epitopi sull'RBD. Le frecce rosse indicano rotazioni di 180° e 45° tra le viste. Visualizzazione adottata: *surface*[6].

1.3 Applicazioni Terapeutiche e Biotecnologiche dei *Nanobodies*

Le proprietà strutturali e funzionali dei *nanobodies* hanno inaugurato nuove prospettive nella ricerca traslazionale e nello sviluppo biotecnologico. Di seguito sono riportate due proposte proposte applicative di particolare rilievo.

In ambito farmaceutico, il primo farmaco basato su un V_HH, e approvato per l'uso clinico, è il *caplacizumab* per il trattamento della porpora trombocitopenica trombotica [20]. La porpora trombocitopenica trombotica (TTP) è una rara e potenzialmente fatale patologia microangiopatica trombotica, caratterizzata dalla formazione di aggregati piastrinici patologici all'interno della microvascolatura sistemica. Questo processo è scatenato da una carenza dell'enzima ADAMTS13 (*A Disintegrin And Metalloproteinase with Thrombospondin type 1 Site, member 13)*, fisiologicamente responsabile della proteolisi dei multimeri del

[6]La resa *surface* mostra la superficie di accessibilità al solvente (SASA), evidenziando l'involucro molecolare e le interfacce di contatto.

fattore di von Willebrand (vWF) di dimensioni particolarmente elevate (UL-vWF). L'assenza di questa attività proteasica porta all'accumulo di UL-vWF nel plasma, con conseguente aggregazione piastrinica spontanea e la formazione di trombi che ostruiscono i vasi di piccolo calibro [21]. Il *caplacizumab*, un *nanobody* diretto contro il dominio A1 del vWF, inibisce l'interazione tra vWF e le piastrine, prevenendo la formazione di microtrombi.

Un altro ambito riguarda il loro utilizzo come domini di *targeting* in recettori chimerici antigenici (CAR) per le terapie cellulari CAR-T (*Chimeric Antigen Receptor T-cell*). Questa strategia di immunoterapia prevede il prelievo di linfociti T dal paziente, la loro ingegnerizzazione *ex vivo* per esprimere un recettore sintetico (CAR) in grado di riconoscere un antigene tumorale, e la successiva reinfusione delle cellule modificate per eliminare in maniera specifica le cellule cancerose [22]. La ridotta dimensione sterica dei V_HH, unita all'elevata specificità e stabilità, consente di generare costrutti di riconoscimento antigenico di nuova generazione con migliorate capacità di penetrazione tissutale e ridotta immunogenicità rispetto alla terapia anticorpale tradizionale. In uno studio pubblicato su *Nature* [23], gli Autori hanno sviluppato un approccio di ingegnerizzazione basato su CRISPR/Cas9 per la generazione di linfociti T esprimenti CAR con domini V_HH diretti contro antigeni espressi in tumori solidi.

1.4 *Overview* degli approcci predittivi sviluppati finora

Il presente lavoro si sviluppa a partire dal caso studio descritto nella Sezione 1.2.2.

I collaboratori del Dipartimento di Chimica dell'Università di Pavia, che hanno preso parte al progetto di tesi, hanno precedentemente impiegato e sviluppato approcci computazionali allo scopo di ottenere delle predizioni sui meccanismi di legame dei *nanobodies*.

In particolar modo, sono state adottate due differenti strategie. In entrambi i casi, gli esiti ottenuti sono risultati parzialmente inadeguati al problema; tali limiti saranno discussi più approfonditamente nel Capitolo 5.

1.4.1 Valutazione predittiva con *AlphaFold 3*-DockQ

In questa prima fase sono state generate predizioni di complessi RBD–*nanobody* utilizzando *AlphaFold 3* (AF3) a partire dalle sole sequenze aminoacidiche dell'RBD e di ciascuno dei *nanobodies* selezionati [24]. Per ogni coppia RBD–*nanobody*, è stato selezionato solo il primo modello predittivo in uscita, corrispondente alla struttura considerata più probabile per AF3. Come indice di confidenza interno al predittore è stato impiegato l'ipTM, che approssima la confidenza sul posizionamento relativo fra epitopo e paratopo. In accordo con la documentazione del *server* [25], si è adottata la seguente interpretazione delle soglie ipTM:

(i) ipTM > 0.8 predizione affidabile;
(ii) $0.6 \leq$ ipTM ≤ 0.8 predizione incerta;
(iii) ipTM < 0.6 predizione non affidabile.

Per la verifica dell'accuratezza strutturale è stato calcolato, per ciascun complesso, il DockQ *score* [26] confrontando la struttura predetta con la corrispondente struttura cristallografica depositata nel *Protein Data Bank* [19]. In pratica, DockQ fornisce una misura della qualità del *docking* (ossia dell'accuratezza del posizionamento reciproco dei *partner* e dei contatti all'interfaccia), e consente di mappare le categorie CAPRI[7] in soglie numeriche:

(i) DockQ > 0.80 (alta);
(ii) $0.49 <$ DockQ ≤ 0.80 (media);
(iii) $0.23 <$ DockQ ≤ 0.49 (accettabile);
(iv) DockQ ≤ 0.23 (non corretto).

[7] CAPRI (*Critical Assessment of PRedicted Interactions*) è un *benchmark* comunitario per la valutazione delle predizioni di complessi proteina–proteina. Le categorie qualitative (Incorrect, Acceptable, Medium, High) sono definite da combinazioni di tre metriche: frazione di contatti nativi (F_{nat}), RMSD all'interfaccia (iRMSD) e RMSD del partner mobile/ligando (LRMSD).

In linea con l'impostazione del presente lavoro (ossia una classificazione binaria secondo meccanismo ortosterico e allosterico, riportato sempre nella Sezione 1.2.2), la soglia decisionale adottata per discriminare predizioni corrette è fissata a DockQ > 0.23.

La ragione di questa scelta risiede nel significato strutturale della soglia CAPRI: un complesso con DockQ $>$ 0.23 presenta un'accurata predizione di almeno una parte dell'interfaccia epitopo-paratopo. Sebbene lontana dalla piena accuratezza strutturale, un modello al di sopra di questa soglia cattura sufficienti dettagli dell'orientamento reciproco per poter distinguere se il legame è ortosterico o allosterico. Al contrario, un modello con DockQ ≤ 0.23, classificato come incorretto, presenta un'interfaccia totalmente mal predetta.

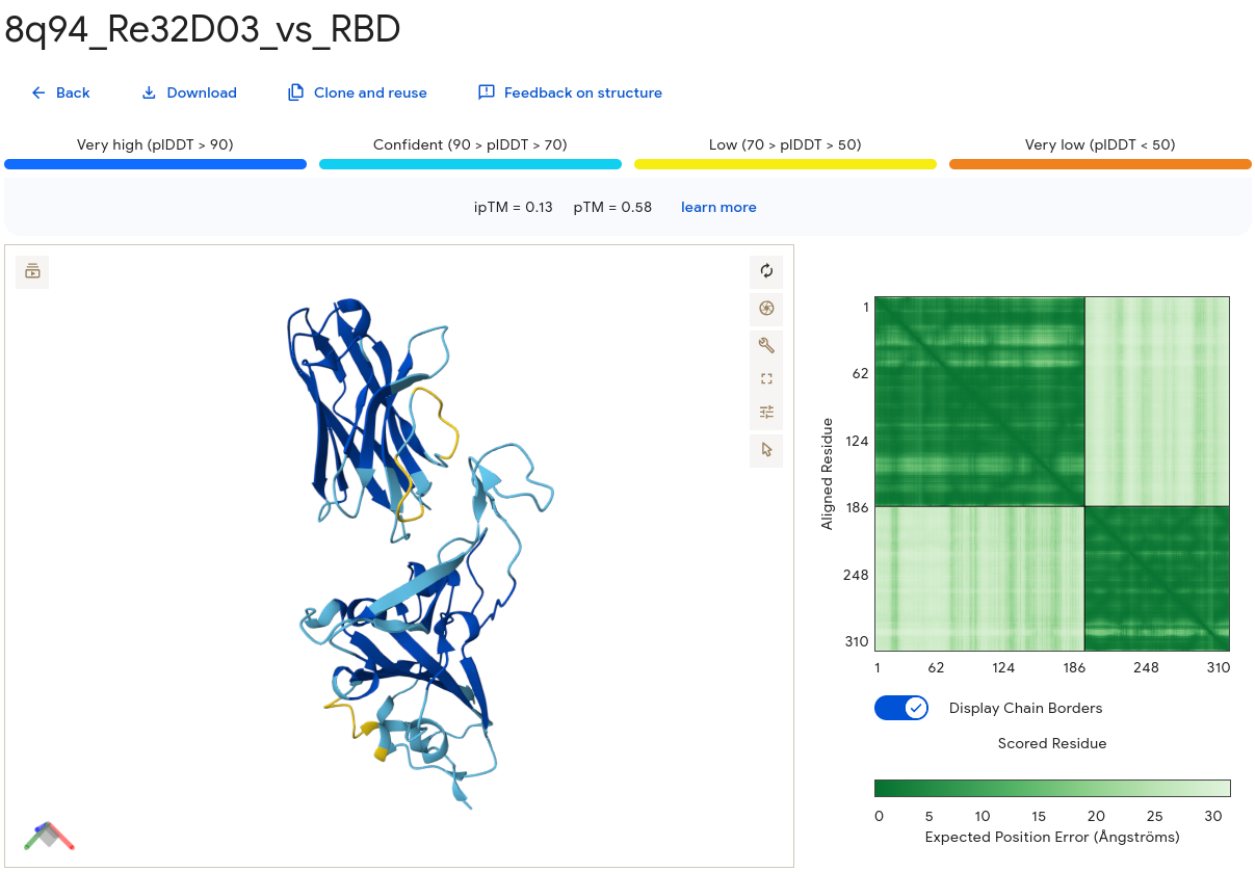

Figura 1.5: Esempio di interfaccia di *AlphaFold server* per il *nanobody* 8q94_Re32D03 (ortosterico) in complesso con il suo RBD. Il punteggio ipTM $= 0.13$ risulta inferiore alla soglia di affidabilità (ipTM < 0.6).

In Figura 1.5 è mostrato un caso di predizione errata tramite AF3; il punteggio DockQ non è mai direttamente riportato nell'interfaccia, ma è stato calcolato a posteriori tramite uno *script ad hoc* ed è pari a DockQ $= 0.053$ (ossia < 0.23), indicando che la predizione non è corretta. Infine, è stato tracciato lo *scatter plot* ipTM vs DockQ per tutti i complessi testati.

1.4.2 Valutazione predittiva con *AlphaFold 3*: *distance refinement*

Per rifinire i risultati ottenuti con AF3, è stato sviluppato un approccio geometrico basato sulle distanze tra l'interfaccia di contatto del paratopo e due epitopi predefiniti sull'RBD (Figura 1.4). L'idea è quella di quantificare, per ciascun *nanobody*, in che misura le diverse pose predette da AF3 privilegino un epitopo rispetto all'altro.

In termini formali, sia R_{RBD} l'insieme dei residui che costituiscono l'RBD e R_{Nb} l'insieme dei residui del *nanobody* considerato. Nel presente metodo sono state considerate le cinque pose predette in *output* da AF3, le quali corrispondono a cinque configurazioni diverse in cui varia la posizione spaziale dei residui, ma non la

struttura in termini di composizione. Le coordinate del carbonio alfa (Cα)[8] del residuo $i \in R_{\mathrm{Nb}}$ per ciascuna posa $p \in \{1,\dots,5\}$ sono indicate con $\mathbf{r}_{i,\mathrm{C}\alpha}^{(p)} \in \mathbb{R}^3$.

Gli epitopi definiti sull'RBD sono modellati come due insiemi fissati di residui $E_1,\ E_2 \subset R_{\mathrm{RBD}}$, comuni a tutti i *nanobodies*.

Operativamente, la procedura si articola nei seguenti passi:

(i) Definizione dei residui in contatto. Per ciascuna posa p si definisce l'insieme dei residui del *nanobody* in contatto con l'RBD come

$$C^{(p)} = \left\{ i \in R_{\mathrm{Nb}} \,\middle|\, \exists\, j \in R_{\mathrm{RBD}} \text{ t.c. } \left\| \mathbf{r}_{i,\mathrm{C}\alpha}^{(p)} - \mathbf{r}_{j,\mathrm{C}\alpha}^{(p)} \right\|_2 < 4\,\text{Å} \right\}, \tag{1.1}$$

cioè un residuo del *nanobody* è considerato in contatto se il suo Cα si trova a distanza inferiore a 4 Å dal Cα di almeno un residuo dell'RBD.

(ii) Centro di massa dei residui in contatto sul *nanobody*. Per ogni posa p, si calcola il centro di massa (COM) delle coordinate dei Cα dei soli residui in contatto:

$$\mathbf{r}_{\mathrm{COM,Nb}}^{(p)} = \frac{\displaystyle\sum_{i \in C^{(p)}} m_i\, \mathbf{r}_{i,\mathrm{C}\alpha}^{(p)}}{\displaystyle\sum_{i \in C^{(p)}} m_i}, \tag{1.2}$$

dove m_i è la massa associata al residuo i.

(iii) Centri di massa dei due epitopi sull'RBD. Per ciascun epitopo $e \in \{1,2\}$ si calcola il COM delle coordinate dei Cα dei residui che lo compongono:

$$\mathbf{r}_{\mathrm{COM,RBD}}^{(e)} = \frac{\displaystyle\sum_{j \in E_e} m_j\, \mathbf{r}_{j,\mathrm{C}\alpha}}{\displaystyle\sum_{j \in E_e} m_j}, \qquad e = 1,2, \tag{1.3}$$

dove m_j è la massa associata al residuo j sull'RBD.

(iv) Calcolo delle distanze centro–centro per ciascuna posa. La prossimità geometrica tra l'interfaccia del *nanobody* e ciascun epitopo è quantificata, per ogni posa p ed epitopo e, tramite la distanza euclidea tra i corrispondenti centri di massa:

$$d_{p,e} \;=\; \left\| \mathbf{r}_{\mathrm{COM,Nb}}^{(p)} - \mathbf{r}_{\mathrm{COM,RBD}}^{(e)} \right\|_2, \qquad p = 1,\dots,5,\ e = 1,2. \tag{1.4}$$

Per ciascun epitopo e si considera quindi la distanza minima tra le cinque pose:

$$d_e^{\min} \;=\; \min_{1 \le p \le 5} d_{p,e}, \qquad e = 1,2, \tag{1.5}$$

ottenendo, per ogni *nanobody*, un valore riassuntivo di prossimità a ciascun epitopo.

L'analisi delle distribuzioni delle distanze $d_{p,e}$ e dei corrispondenti minimi $d_e^{\min}$ ha condotto all'adozione di due soglie $T_1 = 16\,\text{Å}$ e $T_2 = 8\,\text{Å}$.

Si assume che AF3 predice l'epitopo 1 (ortosterico) se, per il *nanobody* considerato, valgono congiuntamente le seguenti condizioni:

$$d_1^{\min} < d_2^{\min}, \qquad d_1^{\min} < T_1, \qquad \left| d_1^{\min} - d_2^{\min} \right| < T_2. \tag{1.6}$$

[8] Il carbonio alfa (Cα) è l'atomo di carbonio chirale centrale nella struttura generica di un amminoacido, covalentemente legato a quattro sostituenti: il gruppo amminico (–NH_2), il gruppo carbossilico (–COOH), un atomo di idrogeno e la catena laterale (gruppo R). Nelle strutture proteiche il Cα costituisce, con gli atomi carbonilici e amminici, il *backbone*.

In modo del tutto analogo, si definisce una preferenza di AF3 per l'epitopo 2 scambiando opportunamente gli indici 1 e 2 in (1.6).

1.4.3 *Docking* proteina-proteina

In seguito, i collaboratori hanno adottato un approccio *state-of-the-art* basato sul *docking* proteina-proteina[9], utilizzando il *tool* BIOLUMINATE della *suite* MAESTRO, prodotta dall'azienda SCHRÖDINGER [27, 28]. Per ognuno dei complessi *nanobodies*-RBD selezionati, è stata eseguita la seguente *pipeline*:

(i) *Epitope-constrained docking* del *nanobody* sia sull'epitopo 1 sia sull'epitopo 2 (Figura 1.4). Per ciascun epitopo, il *workflow* di BIOLUMINATE genera tipicamente pose dell'ordine di 10^3 (numero determinato automaticamente dal *tool*). Le strutture in *output* sono annotate in BIOLUMINATE/MAESTRO con due proprietà visualizzate nella *Project Table*: *PIPER pose energy* e *PIPER pose score*. Lo *score* di interazione tra le due proteine è calcolato come:

$$E \;=\; w_1 E_{\text{rep}} \;+\; w_2 E_{\text{attr}} \;+\; w_3 E_{\text{elec}} \;+\; w_4 E_{\text{DARS}}, \tag{1.7}$$

dove w_n sono i pesi, E_{rep} ed E_{attr} rappresentano rispettivamente i contributi repulsivo e attrattivo di van der Waals, E_{elec} è il termine elettrostatico. La somma E di Eq. (1.7) è riportata come *PIPER pose energy*, mentre il quarto termine ($w_4 E_{\text{DARS}}$) è riportato come *PIPER pose score*. Il termine E_{DARS} è un potenziale *structure-based* costruito secondo l'approccio *Decoys As the Reference State* (DARS) e rappresenta prevalentemente contributi di desolvazione [28, 29].

(ii) Selezione delle pose: per ciascun epitopo, tutte le soluzioni generate vengono ordinate secondo la funzione di punteggio di Eq. (1.7) (*ranking*) e si trattengono le 50 pose con punteggio più favorevole; queste costituiscono il *set* di *docking* per i passaggi successivi.

(iii) *Re-scoring* energetico con MM/GBSA: per ciascun epitopo, le 50 pose selezionate vengono sottoposte a *re-scoring* con MM/GBSA (implementazione PRIME nella *suite* BIOLUMINATE) [30, 31]. Il metodo approssima l'energia libera di legame usando un modello di solvente implicito.
L'energia libera di legame MM/GBSA è quindi calcolata come:

$$\Delta G_{\text{bind}}(\text{MM/GBSA}) \;=\; G_{\text{complex}} \;-\; (G_{\text{RBD}} \;+\; G_{\text{Nb}}), \tag{1.8}$$

dove G_{complex} rappresenta l'energia libera del complesso RBD-*nanobody*. Ogni contributo energetico è ottenuto combinando termini di *force field* con il modello di solvatazione implicito [31]. Se il contributo della 1.8 è negativo significa che G_{complex} è minore della somma delle energie libere dei singoli RBD e *nanobody*, il che, dal punto di vista fisico-chimico, significa che la formazione del complesso è favorita energeticamente. Per questa ragione, nella valutazione dei risultati, se all'interno delle 50 pose sono presenti ΔG_{bind} positivi, questi sono stati esclusi dalla valutazione.

(iv) *Success criterion*: per ciascun epitopo, si costruisce l'istogramma dei valori residui di ΔG_{bind} (dopo l'esclusione delle pose con $\Delta G_{\text{bind}} > 0$), con ΔG_{bind} sull'asse delle ascisse e la frequenza (conteggio delle pose) sull'asse delle ordinate. Si calcola quindi il primo quartile ($Q1$, 25° percentile) di ciascuna distribuzione:

$$Q1_{(e)} \;=\; \text{perc}_{25}\big(\{\Delta G^{(e)}_{\text{bind}}\}\big), \qquad e \in \{\text{epitopo 1}, \text{epitopo 2}\}. \tag{1.9}$$

L'epitopo predetto è quello con quartile più favorevole (più negativo):

$$\widehat{e} \;=\; \arg\min_{e}\; Q1_{(e)}. \tag{1.10}$$

[9] Per *docking* proteina–proteina si intende l'insieme di metodi computazionali che, a partire dalle strutture tridimensionali separate delle due proteine interagenti (in questo caso antigene-anticorpo), predicono la geometria del complesso (posa di legame), esplorando tutti i possibili gradi di libertà rotazionali/traslazionali. Le soluzioni generate vengono valutate con funzioni di *scoring* e raggruppate in *cluster* per identificare i modelli più probabili.

In termini grafici, la distribuzione dei valori di ΔG_{bind} dell'epitopo predetto come favorevole risulta complessivamente spostata verso valori più negativi (coda più estesa sul lato dei valori negativi).
La predizione mediante *docking* è considerata corretta se l'epitopo individuato dal criterio (1.10) coincide con l'epitopo effettivo (ortosterico oppure allosterico).

2 Scopo dello Studio e Obiettivi

A partire da queste premesse, il presente lavoro di ricerca combina simulazioni di dinamica molecolare con modelli di apprendimento supervisionato per sviluppare un approccio integrato volto alla predizione e caratterizzazione dei meccanismi di *binding* dei *nanobodies*.

Le simulazioni di dinamica molecolare sono state essenziali per superare i limiti di una descrizione statica delle interazioni *nanobody*-antigene, consentendo di caratterizzare il comportamento dinamico e la plasticità conformazionale di queste proteine, con maggiore attenzione ai *loop* CDR, e alla CDR3 in particolare. Questa metodologia ha permesso di generare le traiettorie temporali, rappresentative delle possibili conformazioni spaziali del paratopo, da cui sono stati estratti descrittori strutturali e fisico-chimici dipendenti dal tempo, successivamente impiegati come *features* per l'addestramento dei modelli predittivi.

Su tale fondamento, si è optato per l'integrazione di *machine learning* e *deep learning*, capaci di valorizzare i descrittori dinamici e di colmare alcune lacune prestazionali osservate con approcci quali AF3 e il *docking* rigido *proteina-proteina* condotto con BIOLUMINATE (Sezione 1.4).

Alla luce di tali motivazioni, la tesi è divisibile in due parti fondamentali, le quali, corrispondono, anche agli obiettivi del presente studio:

(i) Classificazione e predizione del meccanismo di legame. A partire da una selezione di alcune delle biomolecole prodotte da Xiang et al. [17], suddivise in base al meccanismo di legame con l'epitopo (ortosterico e allosterico), sono state eseguite simulazioni di dinamica molecolare in condizioni *target-free*, comprendendo quindi il solo *nanobody*, svincolato dal complesso con il suo *target* (RBD/*Spike* di SARS-CoV-2.). La scelta metodologica adottata è motivata dall'obiettivo di garantire la massima generalizzabilità dell'indagine, estendendo la caratterizzazione a tutte le possibili modalità di riconoscimento da parte di queste biomolecole. Dalle simulazioni dinamiche sono stati estratti descrittori quantitativi appartenenti a diverse categorie, complementari fra di loro, ciascuna delle quali contribuisce a fornire una prospettiva sul comportamento del sistema. Infine, sono stati addestrati diversi modelli di apprendimento supervisionato che integrano tali descrittori, al fine di cercare il modello *physics-informed* che meglio si adatta al problema. L'intero processo è stato replicato su due *dataset* costruiti dal medesimo *pool* di biomolecole selezionate e organizzati a complessità crescente, al fine di validare la robustezza e la stabilità dei risultati. L'obiettivo ultimo di questa fase è il confronto con altri metodi computazionali per validare l'idea di integrazione delle predizioni con un approccio *physics-based* come strategia promettente per una accurata predizione dei meccanismi di legame.

(ii) Inferenza dei determinanti chimico-fisici. Il lavoro non si limita al solo problema di classificazione binario, ma, attraverso un approccio *data-driven*, cerca di indagare le ragioni chimiche e strutturali che caratterizzano i diversi meccanismi di legame, con lo scopo di identificare i fattori di natura chimico-fisica che ne permettono la classificazione. A questo scopo, una volta addestrati i modelli, è stata eseguita un'analisi dell'importanza delle *features*, supportata anche da evidenze statistiche, con lo scopo di indagare quali descrittori sono maggiormente predominanti per la classificazione. In questo modo, è possibile concludere sulle ragioni che determinano un *nanobody* a legarsi su un epitopo in maniera ortosterica rispetto ad una allosterica.

3 Metodi di Dinamica Molecolare

Nella prima parte del Capitolo sarà fornita una rassegna teorica dei metodi computazionali impiegati, con particolare *focus* sulla dinamica molecolare (MD) come strumento per la caratterizzazione dei *nanobodies*. La seconda parte delinea invece il *setup* operativo sviluppato per lo studio molecolare *in silico*.

3.1 Dinamica Molecolare

I metodi di Meccanica Molecolare (MM), che rappresentano la base teorica sul quale si fonda la trattazione della seguente Sezione, costituiscono approcci empirici finalizzati a descrivere l'energia potenziale di un sistema molecolare in funzione delle sue coordinate geometriche [32]. In tale formalismo, il livello descrittivo più elementare è rappresentato dagli atomi, considerati come le unità fondamentali. Al contrario, gli approcci di Meccanica Quantistica (QM) considerano esplicitamente il comportamento degli elettroni, garantendo una precisione superiore nella caratterizzazione delle proprietà elettroniche del sistema, sebbene con un notevole incremento dei costi computazionali. I metodi MM, al contempo, risultano particolarmente vantaggiosi ed esaustivi per la caratterizzazione di sistemi biologici macromolecolari e di elevata complessità, come proteine, lipidi o acidi nucleici.

La dinamica molecolare (MD) è stata selezionata come metodologia centrale per la caratterizzazione dei *nanobodies* in quanto permette di osservare, su scala atomistica e con risoluzione temporale nell'ordine del femtosecondo ($1fs = 10^{-15}s$), il comportamento del sistema biologico in un ambiente che approssima le condizioni fisiologiche di soluzione acquosa. A differenza di una struttura cristallografica (statica), che fornisce un'unica descrizione istantanea (spesso mediata e talvolta influenzata dalle condizioni sperimentali), una simulazione di MD consente di caratterizzare in maniera qualitativa e, successivamente, quantitativa, le proprietà dinamiche e conformazionali del complesso proteico in esame. In pratica, la MD è una tecnica computazionale che permette di simulare l'evoluzione temporale di biomolecole, assimilate ad un sistema di particelle (atomi) interagenti, dove le traiettorie degli atomi sono ottenute integrando numericamente le equazioni del moto della meccanica classica.

Si consideri un sistema composto da N atomi, ciascuno trattato come un punto materiale caratterizzato da una massa m_i, secondo l'approssimazione di Born-Oppenheimer [33]. La posizione di ogni atomo nello spazio è descritta dal vettore $\mathbf{r}_i = (x_i, y_i, z_i)$. L'insieme di tutte le coordinate atomiche è raccolto in un unico vettore di dimensione $3N$: $\mathbf{r} \equiv (\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N)^T$. Analogamente, le velocità di tutti gli atomi sono raccolte nel vettore $3N$-dimensionale $\mathbf{v} \equiv (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N)^T$. Le masse degli atomi sono organizzate in una matrice diagonale di dimensione $3N \times 3N$, indicata con $\mathbf{M}$, dove ciascuna massa m_i è ripetuta tre volte lungo la diagonale (per corrispondere alle tre dimensioni spaziali).

La forza che agisce su ciascun atomo i è definita come il gradiente negativo dell'energia potenziale del sistema $V(\mathbf{r})$ rispetto alle coordinate dell'atomo:

$$\mathbf{F}_i(\mathbf{r}) = -\nabla_{\mathbf{r}_i} V(\mathbf{r}). \tag{3.1}$$

L'energia potenziale $V(\mathbf{r})$ è una funzione scalare che mappa una data configurazione atomica $\mathbf{r}$ in un valore di energia (tipicamente espresso in kJ/mol o kcal/mol). Il suo gradiente fornisce la forza che spinge il sistema verso stati a minore energia, i quali corrispondono, dal punto di vista fisico, a quelli più stabili per il sistema multi-atomico.

Per l'intero sistema vale la seconda legge di Newton, che può essere scritta in una forma compatta e vettoriale:

$$\mathbf{M}\ddot{\mathbf{r}}(t) = \mathbf{F}_i(\mathbf{r}) = -\nabla_{\mathbf{r_i}} V(\mathbf{r}), \tag{3.2}$$

dove $\ddot{\mathbf{r}}$ rappresenta il vettore delle accelerazioni (il *double-dot* indica la derivata seconda rispetto al tempo

delle coordinate $\mathbf{r}$) e $\nabla_{\mathbf{r}_i} V(\mathbf{r})$ è il gradiente dell'energia potenziale rispetto a tutte le $3N$ coordinate.
Per integrare numericamente queste equazioni, è conveniente riformularle il modello differenziale del secondo ordine (3.2) come un sistema di equazioni differenziali del primo ordine, introducendo esplicitamente le velocità come variabili:

$$\frac{d\mathbf{r}}{dt} = \mathbf{v}, \qquad \frac{d\mathbf{v}}{dt} = \mathbf{M}^{-1}\mathbf{F}_i(\mathbf{r}) = \mathbf{A}(\mathbf{r}), \tag{3.3}$$

dove $\mathbf{A}(\mathbf{r})$ è il vettore delle accelerazioni. Il sistema è quindi completamente definito dalle $6N$ variabili incognite: le $3N$ coordinate di posizione e le $3N$ componenti di velocità.
Nel caso delle simulazioni di MD applicate a sistemi biologici, il potenziale $V(\mathbf{r})$ è definito da una equazione comprendente una serie di contributi [34, 35]. La forma funzionale generale è:

$$V(\mathbf{r}) = V_{\text{legami}} + V_{\text{angoli}} + V_{\text{torsioni}} + V_{\text{torsioni improprie}} + V_{\text{non covalenti}}. \tag{3.4}$$

I primi quattro termini, definiti *bonded*, descrivono le interazioni covalenti all'interno di una molecola. L'ultimo termine ($V_{\text{non covalenti}}$) raccoglie le interazioni non covalenti (*non-bonded*), che a loro volta si scompongono in due contributi fondamentali: le interazioni di van der Waals e quelle elettrostatiche. Queste descrivono le interazioni tra atomi che non sono legati covalentemente, sia all'interno della stessa molecola (per atomi separati da più di tre legami) che tra molecole diverse.

(i) Energia dei legami (V_{legami}): modella l'energia associata allo stiramento o compressione di un legame covalente rispetto alla sua lunghezza di equilibrio.
Questo termine è definito da un potenziale armonico:

$$V_{\text{legami}} = \sum_{\text{legami}} \frac{1}{2} k_b (b - b_0)^2, \tag{3.5}$$

dove la somma è estesa a tutti i legami covalenti, k_b è la costante di forza del legame (valore che riflette la rigidità del legame covalente), b è la sua lunghezza istantanea e b_0 è la sua lunghezza di equilibrio. Il modello matematico che meglio approssima questo termine è l'oscillatore armonico; in meccanica classica, esso ha una frequenza di vibrazione naturale ν, data da:

$$\nu = \frac{1}{2\pi}\sqrt{\frac{k}{\mu}}, \tag{3.6}$$

dove k è la costante di forza e μ è la massa ridotta[1] del sistema dei due atomi. I legami covalenti, soprattutto quelli che coinvolgono atomi leggeri come l'idrogeno (ad esempio, il legame O-H o N-H, in generale X-H), sono caratterizzati da valori di k_b molto alti (che li rendono molto rigidi) e da masse ridotte molto piccole. Il risultato di questa combinazione è una frequenza di vibrazione elevata (dell'ordine dei fs^{-1}), il che significa che un legame covalente completa un ciclo completo di stiramento e compressione in una finestra temporale estremamente breve. Questa proprietà fisica ha una diretta implicazione per le simulazioni di dinamica molecolare. Per integrare accuratamente le equazioni del moto, e catturare correttamente queste oscillazioni senza che l'algoritmo numerico diventi instabile, è necessario utilizzare un passo di integrazione temporale (Δt) estremamente piccolo, tipicamente di 1-2 fs. Questo piccolo passo di integrazione, è il principale *bottleneck* della MD, in quanto è necessario compiere un numero considerevole di passi per simulare intervalli di tempo biologicamente rilevanti, tipicamente nell'ordine dei microsecondi ($1\mu s = 10^{-6}s$), o al più millisecondi ($1ms = 10^{-3}s$). Per mitigare questo problema e consentire passi temporali più lunghi (ad esempio, 2-4 fs), vengono applicati algoritmi di vincolo che bloccano le lunghezze di questi legami più rigidi al loro valore di equilibrio, rimuovendo efficacemente le loro vibrazioni ultra-veloci dal calcolo numerico.

[1] La massa ridotta μ è definita come la massa efficace di un sistema a due corpi secondo la relazione $\mu = \frac{m_1 m_2}{m_1 + m_2}$.

(ii) Energia degli angoli (V_{angoli}): contributo energetico associato alla flessione di un angolo di valenza rispetto al suo valore di equilibrio. L'angolo di valenza è l'angolo formato da due legami covalenti adiacenti che condividono un atomo centrale. Un esempio classico è l'angolo H-O-H in una molecola d'acqua. Anch'esso è modellato da un potenziale armonico:

$$V_{\text{angoli}} = \sum_{\text{angoli}} \frac{1}{2} k_\theta (\theta - \theta_0)^2, \tag{3.7}$$

dove k_θ è la costante di forza angolare, θ è l'angolo di valenza istantaneo e θ_0 è il suo valore di equilibrio ideale, determinato dall'ibridazione dell'atomo centrale (ad esempio, un valore di equilibrio di circa 104.5° per una molecola d'acqua, 109.5° per un carbonio sp^3, 120° per un carbonio sp^2, 180° per un carbonio sp)[2]. La costante di forza k_θ è elevata, sebbene generalmente inferiore a quella dei legami covalenti (k_b), rendendo anche questo un termine ad alta frequenza, seppur leggermente più bassa rispetto alle vibrazioni di legame della Eq. (3.5).

(iii) Energia delle torsioni proprie (V_{torsioni}): termine noto anche come energia dei diedri propri, modella l'energia associata alla libera rotazione attorno a un legame covalente singolo, un movimento che definisce la conformazione tridimensionale di una molecola. A differenza dei termini armonici, questo potenziale è periodico, e descrive le barriere energetiche che una molecola deve superare per ruotare da una conformazione stabile a un'altra. La sua forma funzionale è data da una serie di Fourier:

$$V_{\text{torsioni}} = \sum_{\text{torsioni}} \sum_{n} \frac{k_{\phi,n}}{2} [1 + \cos(n\phi - \delta_n)], \tag{3.8}$$

dove ϕ rappresenta l'angolo di torsione istantaneo, definito come l'angolo tra i piani formati dalle triplette di atomi (i, j, k) e (j, k, l) per una sequenza di quattro atomi consecutivi *i-j-k-l* legati covalentemente. Il parametro n (un intero positivo) indica la periodicità, ovvero il numero di minimi energetici che il potenziale presenta in un giro completo di 360°, ed è determinato dalla simmetria degli atomi coinvolti nella rotazione. Il parametro δ_n (un angolo di sfasamento, spesso 0° o 180°) determina il valore angolare in corrispondenza del quale il potenziale raggiunge il suo minimo, definendo così la conformazione più stabile. Infine, $k_{\phi,n}$ rappresenta l'ampiezza della barriera energetica associata a quella specifica periodicità, ovvero l'energia necessaria per superare il blocco rotazionale tra due conformazioni stabili. In ambito biologico, questo termine risulta di grande importanza per definire le conformazioni dello scheletro peptidico nelle proteine, modellando la formazione di strutture secondarie come eliche α e foglietti β, nonché per determinare la flessibilità e le modalità di interazione delle catene laterali degli amminoacidi.

(iv) Energia delle torsioni improprie ($V_{\text{torsioni improprie}}$): non rappresenta una vera rotazione conformazionale, bensì viene utilizzato come vincolo per imporre una geometria molecolare specifica che non è catturata dagli altri termini del potenziale. La sua funzione principale è preservare la planarità di gruppi strutturali come anelli aromatici o legami carbonilici (dove l'ibridazione sp^2 richiede atomi complanari), nonché mantenere la corretta chiralità attorno a centri stereogenici, impedendo l'inversione stereochimica (ad esempio, il passaggio da una configurazione R a S in un carbonio asimmetrico). A differenza del potenziale periodico delle torsioni proprie, questo termine è tipicamente modellato da un potenziale armonico che penalizza gli scostamenti dal valore angolare ideale:

$$V_{\text{torsioni improprie}} = \sum_{\text{improprie}} \frac{1}{2} k_\xi (\xi - \xi_0)^2, \tag{3.9}$$

[2]Con "sp" si indica uno stato di ibridazione atomica in cui un orbitale s e uno p si combinano per formare due orbitali ibridi diretti a 180° (geometria lineare). Per completezza sono di seguito riportate le altre specifiche di ibridazione: sp^2 $(1s + 2p) \rightarrow$ geometria trigonalare planare, $\sim 120°$; sp^3 $(1s + 3p) \rightarrow$ geometria tetraedrica, $\sim 109.5°$.

dove ξ rappresenta l'angolo improprio istantaneo, calcolato come l'angolo tra il legame *i-j* e il piano formato dagli atomi *j-k-l* in una quadrupla di atomi specificamente definita. Il parametro ξ_0 definisce il valore angolare di equilibrio. La costante di forza k_ξ controlla la rigidità di questo vincolo.

(v) Energia delle interazioni non covalenti ($V_{\text{non covalenti}}$): governa le interazioni tra atomi non legati da interazioni covalenti, descrivendo sia le forze attrattive che repulsive che determinano il riconoscimento molecolare, la stabilità conformazionale e i processi di legame. L'energia totale non covalente è data dalla somma di due contributi fisici distinti: l'interazione di van der Waals e l'interazione elettrostatica. L'interazione di van der Waals (V_{vdW}) descrive le forze di dispersione di London (attrazione a medio raggio dovuta a fluttuazioni dipolari transienti) e la repulsione sterica a corto raggio dovuta al principio di esclusione di Pauli[3]. Questo contributo è modellato dal potenziale di Lennard-Jones, che fornisce una rappresentazione di queste due contributi:

$$V_{\text{vdW}} = \sum_{i<j} 4\varepsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{6} \right], \tag{3.10}$$

dove r_{ij} è la distanza tra gli atomi i e j, ε_{ij} rappresenta la profondità della buca di potenziale (che determina l'intensità dell'attrazione di van der Waals), e σ_{ij} è la distanza alla quale il potenziale tra le due particelle è nullo, definendo così la dimensione sterica effettiva degli atomi.
Il secondo contributo, l'interazione elettrostatica ($V_{\text{elettrostatica}}$), descrive l'attrazione o repulsione coulombiana tra atomi dotati di cariche:

$$V_{\text{elettrostatica}} = \sum_{i<j} \frac{1}{4\pi\varepsilon_0} \frac{q_i q_j}{\varepsilon_r r_{ij}}, \tag{3.11}$$

dove q_i e q_j sono le cariche atomiche determinate dal campo di forze, ε_0 è la permittività del vuoto, ε_r è la costante dielettrica relativa del mezzo, e r_{ij} è la distanza interatomica.

Combinando tutti i contributi descritti, il potenziale di forza totale $V(\mathbf{r})$ si esprime come:

$$\begin{aligned} V(\mathbf{r}) \;=\; & \sum_{\text{legami}} \frac{1}{2} k_b (b - b_0)^2 \;+\; \sum_{\text{angoli}} \frac{1}{2} k_\theta (\theta - \theta_0)^2 \\ & + \sum_{\text{torsioni}} \sum_{n} \frac{k_{\phi,n}}{2} [1 + \cos(n\phi - \delta_n)] \;+\; \sum_{\text{improprie}} \frac{1}{2} k_\xi (\xi - \xi_0)^2 \\ & + \sum_{i<j} 4\varepsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^{6} \right] \;+\; \sum_{i<j} \frac{1}{4\pi\varepsilon_0} \frac{q_i q_j}{\varepsilon_r r_{ij}} . \end{aligned} \tag{3.12}$$

3.1.1 Risoluzione numerica

La risoluzione numerica del sistema di equazioni differenziali ordinarie espresso nella (3.2) richiede l'impiego di tecniche di integrazione a passo discreto. In tale contesto, note le approssimazioni della posizione $\mathbf{r}_n \approx \mathbf{r}(t_n)$ e della velocità $\mathbf{v}_n \approx \mathbf{v}(t_n)$ all'istante t_n, si procede al calcolo delle stesse grandezze all'istante successivo $t_{n+1} = t_n + \Delta t$, dove Δt denota il passo temporale.
Per simulazioni di dinamica molecolare, è preferibile adottare particolari metodi numerici che rispettino le seguenti proprietà: simplettici e tempo reversibili [36]. Nel presente lavoro è stato scelto l'impiego del metodo di Verlet [37]. Al fine di illustrare il metodo, si consideri inizialmente il caso unidimensionale di una particella di massa m soggetta a una forza $F(r)$; le estensioni al caso vettoriale sono immediate. Il metodo di Verlet (noto anche come metodo di Størmer–Verlet) deriva dall'espansione in serie di Taylor della posizione $r(t)$

[3] Le forze di dispersione di London sono forze attrattive dovute a dipoli transienti che si generano a causa delle fluttuazioni della densità elettronica in atomi e molecole vicini. La repulsione a corto raggio è, invece, l'interazione di natura repulsiva che si manifesta a distanze molto ravvicinate a causa del principio di esclusione di Pauli, quando gli orbitali elettronici si sovrappongono.

attorno al tempo t. In particolare, le espansioni in avanti e indietro sono date da:

$$r(t+\Delta t) = r(t) + \dot{r}(t)\,\Delta t + \frac{1}{2}\ddot{r}(t)\,\Delta t^2 + \frac{1}{6}r^{(3)}(t)\,\Delta t^3 + \frac{1}{24}r^{(4)}(t)\,\Delta t^4 + O(\Delta t^5), \tag{3.13}$$

$$r(t-\Delta t) = r(t) - \dot{r}(t)\,\Delta t + \frac{1}{2}\ddot{r}(t)\,\Delta t^2 - \frac{1}{6}r^{(3)}(t)\,\Delta t^3 + \frac{1}{24}r^{(4)}(t)\,\Delta t^4 + O(\Delta t^5). \tag{3.14}$$

Sommando le due espansioni, si osserva la cancellazione dei termini dispari, ottenendo:

$$r(t+\Delta t) + r(t-\Delta t) = 2r(t) + \ddot{r}(t)\,\Delta t^2 + \frac{1}{12}r^{(4)}(t)\,\Delta t^4 + O(\Delta t^6). \tag{3.15}$$

Isolando $r(t+\Delta t)$ si ottiene la forma continua del metodo di Verlet:

$$r(t+\Delta t) = 2r(t) - r(t-\Delta t) + \ddot{r}(t)\,\Delta t^2 + \frac{1}{12}r^{(4)}(t)\,\Delta t^4 + O(\Delta t^6). \tag{3.16}$$

Il termine $\frac{1}{12}r^{(4)}(t)\,\Delta t^4 + O(\Delta t^6)$ rappresenta l'errore di troncamento, indicando che lo schema è accurato fino a termini di ordine $O(\Delta t^4)$ per la posizione.

La corrispondente forma discreta dello schema si ottiene sostituendo l'accelerazione $\ddot{r}(t_n)$ con $a(r_n) = F(r_n)/m$:

$$r_{n+1} = 2r_n - r_{n-1} + a(r_n)\,\Delta t^2. \tag{3.17}$$

Per analizzare l'errore di troncamento, si consideri l'approssimazione alle differenze finite della derivata seconda:

$$\frac{r(t_{n+1}) - 2r(t_n) + r(t_{n-1})}{\Delta t^2} = \ddot{r}(t_n) + \frac{1}{12}r^{(4)}(t_n)\,\Delta t^2 + O(\Delta t^4). \tag{3.18}$$

Sostituendo nella (3.18) la soluzione esatta $r(t_n)$, si ottiene proprio l'errore di troncamento dello schema. Da ciò segue che l'errore locale di troncamento è $O(\Delta t^2)$. Sotto opportune ipotesi di regolarità della funzione $a(r)$, e grazie alla zero-stabilità dello schema, l'errore globale risulta essere $O(\Delta t^2)$.

L'estensione al caso vettoriale, rilevante per la dinamica molecolare, è immediata. Denotando con $\mathbf{r}_n \in \mathbb{R}^{3N}$ il vettore delle posizioni di tutte le particelle (atomi) al tempo t_n, e con $\mathbf{a}(\mathbf{r}) = \mathbf{M}^{-1}\mathbf{F}_i(\mathbf{r})$ l'accelerazione, lo schema diventa:

$$\mathbf{r}_{n+1} = 2\mathbf{r}_n - \mathbf{r}_{n-1} + \Delta t^2\,\mathbf{a}(\mathbf{r}_n). \tag{3.19}$$

Sebbene lo schema (3.19) non coinvolga esplicitamente le velocità, queste possono essere ricostruite a posteriori mediante un'approssimazione alle differenze finite centrate:

$$\mathbf{v}_n \approx \frac{\mathbf{r}_{n+1} - \mathbf{r}_{n-1}}{2\Delta t}. \tag{3.20}$$

Tale approssimazione è consistente con l'ordine dello schema, presentando un errore di troncamento $O(\Delta t^2)$. Tuttavia, in contesti dove è richiesta una valutazione precisa delle velocità (ad esempio per il calcolo della temperatura istantanea), la ricostruzione differenziale può introdurre errori significativi o amplificare il rumore numerico. Ciò motiva l'adozione di varianti dello schema di Verlet che includono esplicitamente le velocità.

Tra le varianti più diffuse in MD vi è la *velocity–Verlet*, che fornisce posizioni e velocità sincronizzate allo stesso istante, mantenendo le proprietà geometriche della famiglia. Con la valutazione esplicita delle forze, lo

schema numerico adottato nel lavoro è il seguente:

$$\mathbf{F}_n = -\nabla_{\mathbf{r}} V(\mathbf{r}_n), \qquad \mathbf{a}_n = \mathbf{M}^{-1}\mathbf{F}_n, \tag{3.21}$$

$$\mathbf{r}_{n+1} = \mathbf{r}_n + \Delta t\,\mathbf{v}_n + \tfrac{1}{2}\Delta t^2\,\mathbf{a}_n, \tag{3.22}$$

$$\mathbf{F}_{n+1} = -\nabla_{\mathbf{r}} V(\mathbf{r}_{n+1}), \qquad \mathbf{a}_{n+1} = \mathbf{M}^{-1}\mathbf{F}_{n+1}, \tag{3.23}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \tfrac{1}{2}\Delta t(\mathbf{a}_n + \mathbf{a}_{n+1}). \tag{3.24}$$

In (3.22) si predice la nuova posizione usando l'accelerazione corrente; in (3.23) si ricalcola il gradiente del potenziale e l'accelerazione alla posizione aggiornata; infine (3.24) corregge la velocità con la media trapezoidale delle accelerazioni ai due estremi del passo. Tutte le quantità sono riferite all'atomo i-esimo della biomolecola.

3.1.2 Controllo degli *Ensemble* Termodinamici

La trattazione fin qui discussa è valida per il microcanonico (*ensemble* NVE), ovvero un sistema isolato caratterizzato dalla conservazione di tre quantità fondamentali: il numero di particelle N (nessuno scambio di materia con l'esterno), il volume V (nessun lavoro di espansione o compressione) e l'energia totale E (nessuno scambio di calore o lavoro).

Più frequentemente, i sistemi biologici sono in contatto con un ambiente che mantiene costanti la temperatura (T) e, in molti casi, anche la pressione (P), con valori caratteristici di circa $T \approx 300\,\mathrm{K}$ e $P \approx 1\,\mathrm{bar}$.

Per confrontare direttamente le simulazioni con i dati sperimentali è necessario campionare diversi *ensemble* statistici: l'*ensemble* canonico (NVT), in cui il numero di particelle (N), il volume (V) e la temperatura (T) restano costanti; e l'*ensemble* isotermo-isobarico (NPT), in cui sono costanti N, P e T. Quest'ultimo è generalmente il più rappresentativo per sistemi biologici in soluzione, perché riproduce le condizioni di laboratorio (temperatura e pressione costanti).

Il fondamento concettuale che giustifica l'utilizzo delle simulazioni di MD per il calcolo di proprietà macroscopiche risiede in un principio della meccanica statistica, noto come ipotesi ergodica [38, 39]. In pratica, questo principio stabilisce che, per un sistema in equilibrio, la media temporale di una qualsiasi osservabile fisica (ad esempio, l'energia potenziale, la pressione o un parametro geometrico) calcolata lungo una traiettoria sufficientemente lunga coincide con la sua media d'insieme, cioè con il valore atteso calcolato sulla distribuzione di probabilità dell'*ensemble* termodinamico di riferimento. Questo si esprime matematicamente come:

$$Q^{\mathrm{obs}}_{\mathrm{ens}} = \langle A \rangle_{\mathrm{ens}} = \langle A \rangle_t = \lim_{t\to\infty} \frac{1}{t} \int_0^t Q(\mathbf{p},\mathbf{r})\mathrm{d}t \approx \frac{1}{\tau_{\mathrm{tot}}} \sum_{r=1}^{\tau_{\mathrm{tot}}} Q(\mathbf{p}_r,\mathbf{r}_r) \tag{3.25}$$

dove Q è una grandezza dipendente dal tempo (funzione di posizioni, velocità e di loro combinazioni), τ_{tot} è il numero totale di *steps* della simulazione, e l'uguaglianza tra media d'insieme e media temporale è garantita dall'ipotesi ergodica[4]. Invece di dover calcolare un valore medio ponderato su tutte le possibili configurazioni microscopiche che il sistema potrebbe assumere a una data temperatura e pressione (computazionalmente proibitiva per sistemi complessi), è sufficiente osservare l'evoluzione del sistema nel tempo e mediare il valore dell'osservabile lungo questa singola, lunga traiettoria. La simulazione diventa così un esperimento *in silico*. Di conseguenza, una proprietà calcolata come media temporale in una simulazione NVT, rappresenta la stima più probabile del valore che si misurerebbe in un esperimento reale condotto alla stessa temperatura.

Il processo di campionamento consiste nella generazione di una serie di configurazioni rappresentative della distribuzione di probabilità dell'*ensemble* scelto. Negli ensemble NVT e NPT, il campionamento corretto è reso possibile dall'impiego di algoritmi di controllo detti, rispettivamente, termostati e barostati. Molte

[4]Nel presente lavoro, qualora venga presentato, il simbolo $\langle . \rangle$ indica l'operazione di media temporale.

proprietà fondamentali dei sistemi biomolecolari mostrano una dipendenza critica dalla temperatura e dalla pressione. Tra queste si segnalano: la densità e la comprimibilità del solvente, la struttura di solvatazione, la costante dielettrica efficace, la stabilità conformazionale, la cinetica di legame, e la mobilità ionica. In una simulazione NVE, queste variabili non sono controllate direttamente, ma emergono dalle condizioni iniziali imposte al sistema. Ne consegue che i loro valori effettivi possono discostarsi in modo significativo dalle condizioni sperimentali desiderate. Per ovviare a questa limitazione, nella pratica simulativa si ricorre a algoritmi di controllo che impongono attivamente i valori desiderati di temperatura e pressione.

Nel presente lavoro di tesi, il controllo della temperatura è stato realizzato mediante il termostato di Langevin, mentre il controllo della pressione è stato attuato tramite il barostato di Berendsen in modalità isotropa.

Termostato di Langevin

Il termostato di Langevin è un formalismo matematico che permette di simulare l'effetto di un *thermal bath* (bagno termico) su un sistema molecolare. Concettualmente, il sistema non è più considerato isolato ma in contatto con un ambiente esterno infinitamente più grande, caratterizzato da una temperatura costante T_0. Questo bagno termico, avendo un numero di gradi di libertà molto superiore a quello del sistema, agisce come un serbatoio di calore infinito, in grado di scambiare energia con il sistema senza che la sua temperatura vari in modo apprezzabile. L'interazione con questo serbatoio avviene attraverso due meccanismi fisici complementari:

(i) una forza dissipativa (o di attrito), che rimuove energia dal sistema;
(ii) una forza stocastica (o rumore), che immette energia nel sistema in modo casuale.

L'equazione fondamentale che incorpora questi effetti è una modifica dell'equazione del moto. Per ogni atomo i (di massa m_i, posizione $\mathbf{r}_i$ e velocità $\mathbf{v}_i$) si ha:

$$m_i \dot{\mathbf{v}}_i \;=\; \mathbf{F}_i(\mathbf{r}) \;-\; \underbrace{m_i \gamma \mathbf{v}_i}_{\text{Attrito}} \;+\; \underbrace{\mathbf{R}_i(t)}_{\text{Rumore}}, \tag{3.26}$$

dove $\mathbf{F}_i(\mathbf{r}) = -\nabla_{\mathbf{r}_i} V(\mathbf{r})$ rappresenta la forza derivante dal potenziale $V(\mathbf{r})$ (Eq. 3.1). Il termine $-m_i \gamma \mathbf{v}_i$ introduce una forza di attrito viscoso proporzionale e opposta alla velocità dell'atomo. Il coefficiente γ (espresso in ps^{-1}) determina l'intensità dell'accoppiamento con il *thermal bath* e fisicamente rappresenta la frequenza media con cui una particella urta contro le particelle del bagno termico. Il termine $\mathbf{R}_i(t)$ rappresenta gli urti casuali dovuti all'interazione con le particelle del *thermal bath*, e viene modellizzato come un processo stocastico gaussiano.

Questi termini, influenzano direttamente la temperatura istantanea del sistema, monitorata attraverso la temperatura cinetica, definita come:

$$T_{\text{kin}}(t) \;=\; \frac{2K(t)}{f\,k_B}, \quad \text{con} \quad K(t) = \frac{1}{2}\sum_i m_i \mathbf{v}_i^2, \tag{3.27}$$

dove k_B è la costante di Boltzmann e f rappresenta il numero di gradi di libertà effettivi del sistema. Il valore di f dipende dai vincoli applicati al sistema e tipicamente è pari a $3N - N_{\text{vincoli}}$. All'equilibrio, il valor medio dell'energia cinetica soddisfa il teorema di equipartizione dell'energia [40], che in formule:

$$\langle K \rangle = \frac{1}{2} f k_B T_0 \quad \Rightarrow \quad \langle T_{\text{kin}} \rangle = T_0, \tag{3.28}$$

dimostrando che il termostato mantiene efficacemente la temperatura desiderata.

La scelta del parametro γ rappresenta un *trade-off* tra l'efficienza del termostato e la perturbazione della dinamica del sistema.

Barostato di Berendsen

Il controllo della pressione in dinamica molecolare viene realizzato mediante l'utilizzo di algoritmi detti barostati. Tra questi, il barostato di Berendsen rappresenta uno schema di *weak coupling* (accoppiamento debole) che regola il volume del sistema in risposta alla differenza tra la pressione istantanea P_{inst} e un valore target P_0 desiderato [41]. La pressione istantanea in una simulazione molecolare è una quantità microscopica che fluttua rapidamente nel tempo. Il suo valore viene calcolato a partire dal teorema del viriale [35], che per un sistema periodico assume la forma:

$$P_{\text{inst}} = \frac{2K}{3V} + \frac{1}{3V}\sum_{i=1}^{N} \mathbf{r}_i \cdot \mathbf{F}_i, \tag{3.29}$$

dove:

(i) $K = \frac{1}{2}\sum_i m_i \mathbf{v}_i^2$ è l'energia cinetica totale del sistema;
(ii) V è il volume istantaneo della cella di simulazione;
(iii) $\mathbf{F}_i$ è la forza totale agente sull'atomo i.

Si noti come la somma in (3.29) si estende su tutti gli N atomi del sistema. Il barostato di Berendsen si basa sulla risposta termodinamica del volume a variazioni di pressione.
La grandezza di maggiore rilievo è la comprimibilità isoterma κ_T, definita come:

$$\kappa_T \equiv -\frac{1}{V}\left(\frac{\partial V}{\partial P}\right)_T, \tag{3.30}$$

che quantifica la variazione relativa di volume in risposta a una variazione di pressione a temperatura costante. Nel caso applicativo in esame, κ_T può essere considerata approssimativamente costante per piccole variazioni di pressione. Di conseguenza l'Equazione (3.30) può essere integrata per ottenere:

$$\frac{\Delta V}{V} \approx -\kappa_T \, \Delta P. \tag{3.31}$$

Il barostato di Berendsen impone un rilassamento esponenziale della pressione del sistema verso il valore target P_0. L'equazione differenziale che governa questo processo è:

$$\frac{dP}{dt} = -\frac{1}{\tau_P}\left(P - P_0\right), \tag{3.32}$$

dove τ_P è il tempo caratteristico di rilassamento della pressione. Combinando l'Equazione (3.32) con la relazione termodinamica (3.31), e notando che $\Delta P = P_{\text{inst}} - P_0$, si ottiene l'equazione per la dinamica del volume:

$$\frac{dV}{dt} = \frac{\kappa_T}{\tau_P}\left(P_{\text{inst}} - P_0\right) V. \tag{3.33}$$

Questa equazione differenziale ordinaria esprime la logica di controllo: se $P_{\text{inst}} > P_0$, il volume aumenta (espansione) per ridurre la pressione; se $P_{\text{inst}} < P_0$, il volume diminuisce (compressione) per aumentare la pressione. La costante $\frac{\kappa_T}{\tau_P}$ determina la rapidità della risposta.
Per l'implementazione in MD, l'Equazione (3.33) viene discretizzata con un passo temporale Δt. Assumendo che la variazione relativa di volume per passo sia piccola ($|\frac{\Delta V}{V}| \ll 1$), si ottiene:

$$\frac{V(t+\Delta t) - V(t)}{V(t)} \approx \frac{\kappa_T \, \Delta t}{\tau_P}\left(P_{\text{inst}} - P_0\right). \tag{3.34}$$

Riorganizzando i termini, si arriva alla forma:

$$\frac{V(t+\Delta t)}{V(t)} \approx 1 + \underbrace{\frac{\kappa_T \Delta t}{\tau_P}}_{\alpha_P} \left(P_{\text{inst}} - P_0\right), \tag{3.35}$$

dove α_P è un parametro adimensionale che controlla l'intensità della correzione di volume per passo.

Nella sua forma isotropa, il barostato di Berendsen applica una riscalatura uniforme di tutte le dimensioni del sistema. Il fattore di scalatura s è calcolato come:

$$s = \left(\frac{V(t+\Delta t)}{V(t)}\right)^{1/3} = \left[1 + \alpha_P \left(P_{\text{inst}} - P_0\right)\right]^{1/3}.$$

Tutte le coordinate atomiche $\mathbf{r}_i$ e i vettori della cella periodica $\{\mathbf{a}_k\}$ vengono quindi riscalati omogeneamente:

$$\mathbf{r}_i \leftarrow s\,\mathbf{r}_i, \qquad \mathbf{a}_k \leftarrow s\,\mathbf{a}_k. \tag{3.36}$$

Questa trasformazione preserva la forma e le proporzioni della cella di simulazione mentre ne modifica il volume.

3.2 *Setup* delle simulazioni di Dinamica Molecolare

Per ciascun *nanobody* sono state condotte simulazioni di dinamica molecolare (MD) in ambiente LINUX, utilizzando la *suite* AMBER24 (*Assisted Model Building with Energy Refinement*). AMBER costituisce un pacchetto *software* completo, comprendente sia i *force field*[5] sia i moduli per l'esecuzione delle simulazioni (ad esempio `pmemd`) e per l'analisi dei risultati (come `cpptraj` di `AmberTools`) [42]. Le simulazioni sono state avviate mediante *script* BASH, adattati per automatizzare l'intera *pipeline* computazionale, eseguite sfruttando le risorse computazionali del *cluster* del Dipartimento di Chimica dell'Università di Pavia, equipaggiato con 2 processori Intel Xeon Gold 5318Y [43], per un totale di 48 core, e 2 GPU NVIDIA Tesla A100 (80 GB ciascuna) con il *toolkit* CUDA versione 12.2 per la parallelizzazione dei calcoli [44, 45].

Prima di avviare la fase di produzione, corrispondente all'integrazione numerica delle equazioni del moto, è necessario implementare una serie di procedure preliminari finalizzate alla preparazione del sistema. Tali procedure, articolate in stadi successivi, hanno l'obiettivo di garantire la stabilità numerica della simulazione e l'adeguato rilassamento del sistema verso condizioni termodinamiche di equilibrio. Il protocollo preparatorio si articola nelle seguenti fasi sequenziali:

- *(i)* Preparazione delle strutture: selezione dei *nanobodies*, ispezione visuale, estrazione e ottimizzazione delle biomolecole, assegnazione degli stati di protonazione, solvatazione e neutralizzazione della carica;
- *(ii)* Minimizzazione dell'energia: rilassamento della geometria attraverso algoritmi di ottimizzazione per eliminare sovrapposizioni atomiche e tensioni conformazionali residue;
- *(iii)* Riscaldamento graduale: incremento controllato della temperatura nell'*ensemble* NVT;
- *(iv)* Equilibrazione: raggiungimento dell'equilibrio termodinamico completo in *ensemble* NPT mediante applicazione combinata di termostato e barostato;
- *(v)* Produzione: dallo stato NPT equilibrato si avvia la fase di produzione (MD in NPT) e di generazione di repliche indipendenti.

Completata la produzione, si procede finalmente all'analisi delle traiettorie, e all'estrazione dei descrittori.

[5] Con il termine *force field* si indica l'insieme delle funzioni analitiche e dei parametri che approssimano l'energia potenziale del sistema in funzione delle coordinate atomiche, includendo termini covalenti (legami, angoli, torsioni) e non covalenti (interazioni di van der Waals ed elettrostatiche).

Figura 3.1: *Worflow* completo adottato nel presente lavoro.

3.2.1 Preparazione delle strutture

Selezione dei *nanobodies*

L'indagine sui *nanobodies*, così come in generale lo studio sull'interazione proteina-proteina, si fonda sul presupposto secondo il quale una piena comprensione dei meccanismi strutturali e funzionali che ne determinano l'attività biologica non può prescindere dall'analisi dell'interfaccia paratopo-epitopo a livello di risoluzione atomica. Tale esigenza, il cui fondamento emerge anche dall'evidenza sperimentale in sede di saggi laboratoriali, nasce dalla constatazione che questi, sebbene indispensabili, risultano insufficienti per una caratterizzazione esaustiva del riconoscimento molecolare.

A questo proposito, la cristallografia a raggi X e la criomicroscopia elettronica a particella singola forniscono una visione strutturale ad alta risoluzione; metodi in soluzione come la diffusione di raggi X a piccolo angolo (SAXS) e la risonanza magnetica nucleare (NMR) aggiungono informazioni sulle conformazioni e sui movimenti molecolari che non sono completamente catturati dalle tecniche che richiedono lo stato cristallino [46, 47]. Nel presente lavoro, tuttavia, l'attenzione non è rivolta ai dettagli operativi di tali metodiche, bensì al loro *output* finale: la generazione e la deposizione di strutture nel *Protein Data Bank* (PDB), archivio di riferimento internazionale per le strutture di proteine e complessi biomolecolari [19].

Concettualmente, il PDB non si limita ad archiviare rappresentazioni grafiche di molecole, bensì conserva i dati sperimentali e i modelli atomistici derivati dalla loro interpretazione. Il file PDB (o il cui formato moderno è PDBx/mmCIF) costituisce una rappresentazione delle coordinate tridimensionali (attraverso una terna x, y, z) per ciascun atomo della macromolecola risolta. A queste coordinate strutturali sono associati metadati di diversa natura:

- *(i)* Ciascun atomo è univocamente identificato mediante il tipo elementare (ad es. C, N, O, S), un numero progressivo e il residuo amminoacidico di appartenenza.
- *(ii)* Specifiche dei legami chimici (tabella `CONECT`) che definiscono la topologia molecolare.
- *(iii)* Informazioni relative alla metodologia strutturale impiegata, alla risoluzione ottenuta e alle condizioni sperimentali.

Tuttavia, è fondamentale riconoscere i limiti intrinseci di questa risorsa. Le strutture depositate nel PDB rappresentano tipicamente un'istantanea statica, detta anche minimo energetico cristallografico, spesso selezionato per catturare conformazioni ordinate e stabili. Questo modello, non è in grado di rappresentare appieno la variabilità conformazionale dei complessi biomolecolari. Proprio in questo contesto, l'integrazione con la dinamica molecolare si rivela non solo utile, ma necessaria. Le simulazioni di MD, inizializzate sulle coordinate cristallografiche del PDB, consentono di superare il limite della rappresentazione cristallografica, trasformandola in una traiettoria temporale.

Il punto di partenza del *workflow* operativo di questo lavoro, in accordo con la trattazione della Sezione 1.2, è una selezione mirata di 50 *nanobodies* (V_HH) in complesso con l'RBD del virus del Covid (complesso V_HH:SARS-CoV-2), provenienti dal PDB, organizzati in due classi bilanciate, composte rispettivamente da 25 complessi di classe I (*nanobodies* ortosterici) e 25 esemplari di classe II (*nanobodies* allosterici). Le strutture cristallografiche sono state selezionate seguendo due criteri:

- *(i)* unicità (non ci sono *nanobodies* ripetuti in termini di sequenza amminoacidica);
- *(ii)* alta risoluzione atomica della struttura depositata nel PDB, sempre minore di 3 Å.

Si precisa che la selezione delle biomolecole candidate, in base ai criteri sopra indicati, ha portato a un numero di strutture cristallografiche che tiene conto sia delle esigenze computazionali e dei tempi di produzione, sia

della necessità di garantire la qualità delle dinamiche simulate.

Inoltre, la scelta di garantire un bilanciamento perfetto tra le classi risponde all'esigenza di mitigare effetti di sbilanciamento nelle successive fasi di addestramento dei modelli.

I composti selezionati sono riportati nella Tabella 3.1.

Tabella 3.1: *Nanobodies* impiegati nelle simulazioni MD suddivisi in base al tipo di legame con l'epitopo. Ogni stringa è nel formato `PDBID_NanobodyID`.

Class I Orthosteric	**Class II Allosteric**
7c8v_SR4	8cyc_Nb2-34
6yz5_H11-D4	8cy7_Nb2-38
8cya_Nb2-67	8cyb_Nb1-8
7kgj_Sb45	7klw_Sb68
7kgk_Sb16	7fat_Nb1A7
7mfu_Sb14	7nkt_NM1226
7fau_Nb1B11	7oap_C1
7oao_C5	7oay_F2
7olz_Re5D06	7olz_Re9F06
7f5g_DL4	7fbj_17F6
8gz5_VHH-P17	7fbk_20G6
7wd1_R14	7x2j_Nb70
8q7s_Ma6F06	7x2m_1-2C7
8q95_Ma16B06	7wd2_S43
8owv_H6	8hr2_Nb1B5
6zxn_Ty1	8q7s_Re21H01
7voa_aRBD5	8q95_Ma3F05
7z1c_B5	8owt_A8
7tpr_8A2	7tpr_7A3
7w1s_Nb-007	8h5u_Nb-021
7kn5_VHHE	7kn6_VHHV
7oap_H3	7my2_Nb30
7rby_Nb112	8elq_C4-255
7x7e_Nb22	8q93_Re21D01
8q94_Re32D03	8q94_Ma3B12

Ispezione Visuale delle Biomolecole

In primo luogo, per un'ispezione visiva, ciascuna struttura viene caricata in PYMOL[6] [48] (*The PyMOL Molecular Graphics System, Version 3.0 Schrödinger, LLC*), un'interfaccia grafica per la visualizzazione di biomolecole creata dall'azienda SCHRÖDINGER, volta a verificare la completezza delle catene amminoacidiche (sequenza). Eventuali molecole d'acqua presenti nelle strutture depositate nel PDB (dette anche acque cristallografiche) sono state rimosse utilizzando sempre PYMOL.

Estrazione e ottimizzazione delle biomolecole

Come precedentemente riportato nel Capitolo 2, una scelta metodologica fondamentale di questo studio, è quella di simulare tramite MD unicamente il *nanobody* in soluzione acquosa, senza il complesso formato dal V_HH e dal suo antigene bersaglio (il *Receptor-Binding Domain*, RBD), senza quindi includere l'intero

[6]Qualora non specificato diversamente, tutte le immagini delle biomolecole riportate nel presente lavoro sono state realizzate con PYMOL.

trimero della proteina *Spike* di SARS-CoV-2. Questa decisione di eseguire simulazioni *target-free* è dettata da un bilanciamento preciso tra accuratezza biologica e generalizzabilità computazionale. Da un lato, simulare il complesso *nanobody*-antigene fornirebbe un'analisi dettagliata del riconoscimento molecolare, ma vincolerebbe i risultati al solo contesto dell'interazione con l'RBD di SARS-CoV-2. Dall'altro, come precedentemente riportato nel Capitolo 2, l'obiettivo ultimo di questa ricerca non è limitato alla comprensione del solo sistema anti-SARS-CoV-2, ma ambisce a estrarre principi generali, predittivi e generalizzabili per la caratterizzazione dei *nanobodies*. Pertanto, il sistema SARS-CoV-2 funge da caso studio eccezionale e ben documentato da cui attingere strutture iniziali di alta qualità, aprendo le possibilità ad una successiva analisi di qualsiasi altro complesso V_HH-antigene di interesse futuro, anche al di fuori del contesto virologico.

Per questo motivo, i *nanobodies* sono stati estratti dal complesso con RBD, ed è stato quindi salvato un nuovo *file* PDB contenente solo la struttura del *nanobody*.

Solvatazione

La scelta del solvente ricopre un ruolo di rilievo nelle simulazioni di dinamica molecolare, in quanto influenza sia l'accuratezza dei risultati che il costo computazionale. Nel seguente lavoro le simulazioni di MD sono state effettuate in solvente (acquoso) esplicito. Per solvente esplicito si intende una descrizione in cui le molecole di solvente e gli ioni sono rappresentati individualmente come particelle con coordinate e potenziali propri. Il principale svantaggio di questo approccio è il notevole aumento del costo computazionale, poiché il numero di atomi nel sistema può aumentare di ordini di grandezza, per via degli ioni e delle molecole di acqua. Tuttavia, questo offre un'accuratezza della simulazione elevata.

L'efficienza computazionale della MD è influenzata anche dalla forma geometrica della cella periodica. Per le simulazioni di questo lavoro è stata selezionata una cella a forma di cubo, scelta per la sua semplicità di implementazione e minore complessità computazionale.

Oltre alla scelta della forma della *box*, è necessaria anche l'implementazione di condizioni periodiche al contorno (*Periodic Boundary Conditions*, PBC) [41]. Le condizioni periodiche al contorno costituiscono un artificio computazionale per simulare un ambiente di *bulk*, eliminando gli effetti di superficie che si avrebbero confinando il sistema in una scatola finita circondata dal vuoto. Concettualmente, le PBC replicano la cella di simulazione primaria (cubo) in tutte e tre le direzioni dello spazio, generando un reticolo infinito e periodico di immagini identiche del sistema originale. In questo schema, ogni particella presente nella cella primaria interagisce non solo con tutte le altre particelle all'interno della stessa cella, ma anche con le particelle-immagine delle celle adiacenti, o, più precisamente, con la copia più vicina di ogni altra particella nel reticolo periodico. Per garantire che, sotto condizioni periodiche, ciascuna coppia di atomi contribuisca una sola volta ai termini non legati a corto raggio entro il *cutoff*, si applica la *minimum-image convention* [35]. Questo principio stabilisce che, per una data coppia di atomi i e j, si consideri solo l'interazione con la copia di j più vicina a i nell'insieme formato dalla cella primaria e dalle sue immagini periodiche.

Nel lavoro è stato scelto un raggio di *cutoff* pari a $r_c = 8$Å, che definisce la distanza massima alla quale le interazioni non covalenti (Equazioni 3.10 e 3.11) vengono calcolate esplicitamente: oltre questa soglia, si assume che tali interazioni siano così deboli da poter essere trascurate senza compromettere significativamente l'accuratezza fisica del modello.

***Step* del protocollo**

Il protocollo adottato si articola come segue:

- *(i)* Le strutture dei *nanobodies* sono state convertite in un file PDB compatibile con AMBER utilizzando l'*utility* `pdb4amber` inclusa nella *suite* AMBER24: questa operazione *standardizza* la nomenclatura atomica e risolve eventuali incongruenze nel file PDB originale.
- *(ii)* Aggiunta degli atomi di idrogeno ai residui non protonabili[7] della struttura.

[7] Per "protonabili" si intendono siti (residui o gruppi funzionali) che possono acquisire o cedere un protone. Esempi includono le

(iii) Assegnazione degli stati protonici ai residui protonabili (quali ad esempio istidina e aspartato) mediante l'algoritmo PROPKA a pH=7.4 [49].
Oltre ad assegnare lo stato protonico, PROPKA corregge le conformazioni dei gruppi funzionali coinvolti per favorire interazioni non-covalenti. In particolare, l'algoritmo identifica e risolve conformazioni ambigue o subottimali di residui come asparagina (Asn), glutammina (Gln) e istidina (His), riorientandone i gruppi laterali per massimizzare l'efficienza dei legami idrogeno. L'*output* finale fornisce non solo una mappa degli stati di protonazione, ma anche una geometria ottimizzata, garantendo la coerenza elettrostatica e strutturale del sistema prima dell'avvio della simulazione.

(iv) Utilizzo dell'utility `tleap`, impiegata per generare i file di topologia (`.parm`, descrizione completa della molecolarità del sistema, quali atomi, legami, parametri di *force field*, cariche atomiche) e coordinate (`.prmtop`, le posizioni tridimensionali x, y, z di tutti gli atomi all'inizio della simulazione).

Finalmente, il protocollo specifico adottato include i seguenti parametri:

(i) Applicazione del *force field* AMBER ff14SB per la proteina e selezione del modello di solvente TIP3P per l'acqua [50, 51].

(ii) Solvatazione del sistema in un *box* cubico di molecole d'acqua, con una distanza minima di 12.0 Å tra qualsiasi atomo della proteina e i bordi del *box*.

(iii) Neutralizzazione della carica netta del sistema mediante aggiunta di un appropriato numero di ioni Na^+ (sodio) e Cl^- (cloro), rendendo il sistema elettricamente neutro. Questa procedura avviene tramite sostituzione di molecole casuali di acqua in posizioni elettrostaticamente favorevoli.

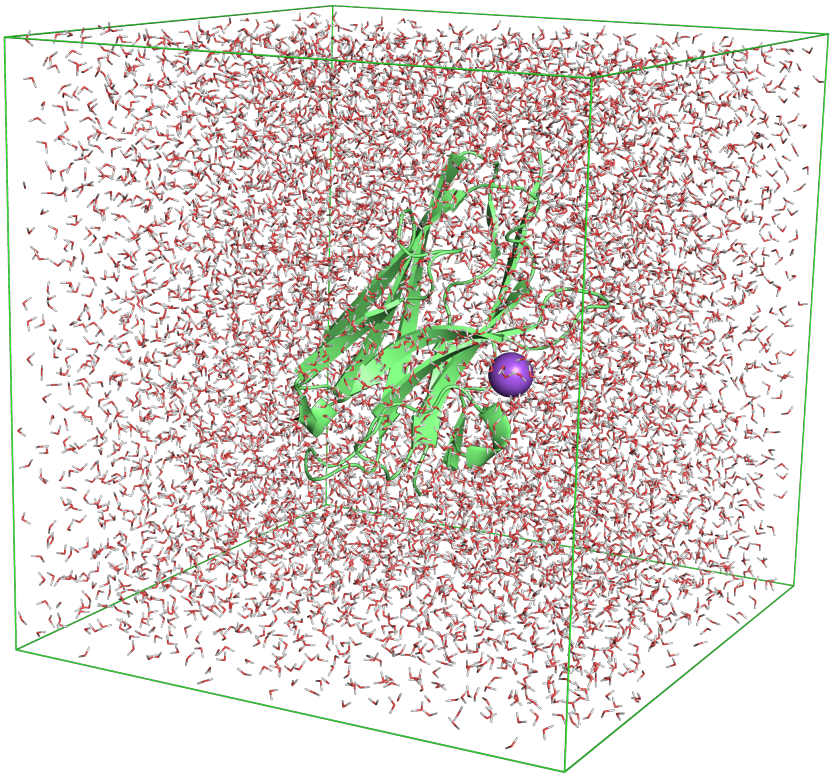

Figura 3.2: Risultato finale del protocollo di MD adottato, applicato al *nanobody* 7oap_C1 (allosterico). In figura, è possibile vedere il *nanobody* (in verde), immerso nella *box* cubica e circondato dalle molecole di solvente (acqua). In viola è riportato un atomo di Sodio (Na^+), aggiunto per neutralizzare la carica netta del sistema.

catene laterali di Asp/Glu (acide), His/Lys/Arg (basiche).

3.2.2 Minimizzazione dell'energia

La fase di minimizzazione dell'energia rappresenta un passo essenziale per preparare il sistema alla MD propriamente detta. Il suo obiettivo principale è rilassare la configurazione iniziale (il cosiddetto minimo cristallografico) rimuovendo contatti atomici troppo ravvicinati, e rifinendo geometrie locali non ottimali, senza introdurre perturbazioni eccessive sulla struttura del soluto [42]. L'approccio adottato prevede la minimizzazione dell'energia potenziale totale definita come:

$$V(\mathbf{r}; k_{\text{pos}}) \;=\; V_{\text{ff}}(\mathbf{r}) \;+\; V_{\text{pos}}(\mathbf{r}; k_{\text{pos}}, H), \tag{3.37}$$

dove V_{ff} rappresenta il contributo descritto nell'Eq. (3.12), e V_{pos} è un termine di vincolo posizionale armonico applicato agli atomi pesanti del soluto (insieme H, ovvero tutti gli atomi ad eccezione degli idrogeni):

$$V_{\text{pos}}(\mathbf{r}; k_{\text{pos}}, H) \;=\; \frac{k_{\text{pos}}}{2} \sum_{i \in H} \left\| \mathbf{r}_i - \mathbf{r}_i^{\text{ref}} \right\|^2 . \tag{3.38}$$

In questa espressione, $\mathbf{r}_i$ indica la posizione corrente dell'atomo i, $\mathbf{r}_i^{\text{ref}}$ la posizione di riferimento (derivata dalla struttura cristallografica, ovvero dal PDB), e k_{pos} la costante di forza del vincolo (in kcal mol^{-1} Å^{-2}).
La ricerca del minimo è formalizzata come:

$$\mathbf{r}^{\star} \;=\; \arg\min_{\mathbf{r}} V(\mathbf{r}; k_{\text{pos}}), \tag{3.39}$$

ed è eseguita attraverso un processo iterativo che prevede la riduzione progressiva della costante di forza permettendo al sistema di rilassarsi in modo controllato attorno alla geometria sperimentale.

La procedura di minimizzazione adottata nel lavoro è stata articolata in due stadi consecutivi, ciascuno della durata complessiva di 300 *steps*. Ogni stadio è stato ulteriormente suddiviso nell'applicazione sequenziale di due diversi algoritmi di ottimizzazione: i primi 10 cicli hanno impiegato l'algoritmo dello *steepest descent* (discesa più ripida) particolarmente robusto in presenza di regioni con alta curvatura del potenziale e di atomi ravvicinati, seguiti da 290 *steps* con l'algoritmo del *conjugate gradient* (gradiente coniugato) per un raffinamento più preciso della geometria molecolare nelle regioni vicine al minimo energetico [35]. Questa ripartizione algoritmica (10 + 290 = 300 *steps*) è stata applicata in entrambi gli stadi di minimizzazione, sebbene con differenti condizioni al contorno: nel primo stadio la minimizzazione è stata applicata selettivamente ai soli atomi di idrogeno, mantenendo il resto del sistema vincolato armonicamente con una costante di forza di 5 kcal mol^{-1} Å^{-2} (con riferimento all'Eq.ne 3.38), mentre nel secondo stadio l'intero sistema è stato minimizzato senza l'applicazione di vincoli.

Il criterio di convergenza richiede che, tra due iterazioni consecutive della minimizzazione, lo spostamento massimo di qualunque coordinata cartesiana sia inferiore ad una soglia. Formalmente:

$$\|\mathbf{r}^{(m)} - \mathbf{r}^{(m-1)}\|_{\infty} = \|\Delta\mathbf{r}\|_{\infty} < \delta, \quad \delta \approx 10^{-3}\,\text{Å}. \tag{3.40}$$

La norma infinito $\|\cdot\|_{\infty}$ seleziona la componente di spostamento di modulo maggiore. La condizione (3.40) garantisce una conformazione che preserva, entro fluttuazioni minime, le coordinate sperimentali e allo stesso tempo minimizza l'energia potenziale del sistema.

3.2.3 Equilibrazione

Equilibrazione del solvente

Completata la minimizzazione, il sistema è stato sottoposto a una fase di equilibrazione del solvente della durata di 9 ps (9.000 *steps*) in *ensemble* canonico (NVT), con passo d'integrazione di 1 fs. Le velocità

iniziali degli atomi del solvente sono state campionate dalla distribuzione di Maxwell-Boltzmann[8] [39] alla temperatura iniziale di $T = 25$ K, mentre l'*ensemble* è stato controllato mediante il termostato di Langevin (Equazione 3.26).

Per garantire la stabilità del sistema durante l'equilibrazione, tutti gli atomi non appartenenti al solvente sono stati vincolati armonicamente con costante di forza $k = 10$ kcal mol^{-1} Å^{-2} (Eq. 3.38). Il profilo termico è stato articolato in tre fasi successive: riscaldamento da 25 K a 400 K nei primi 3 ps (3.000 *steps*), mantenimento a 400 K per 3 ps (3.000 *steps*), e raffreddamento finale a 25 K negli ultimi 3 ps (3.000 *steps*).

Riscaldamento del sistema

A partire dalle fasi successive all'equilibrazione del solvente, è stato implementato l'algoritmo SHAKE [52] per vincolare tutti i legami che coinvolgono atomi di idrogeno (*involving hydrogen bonds*). Questo approccio, permette di utilizzare passi d'integrazione più lunghi (2 fs) mantenendo la stabilità numerica (in accordo con la trattazione dell'Equazione 3.6, che evidenzia come i legami X-H siano caratterizzati dalle vibrazioni più rapide del sistema).

Si è quindi proceduto al riscaldamento del sistema completo da 25 a 300 K in 20 ps (10.000 *steps*) in *ensemble* NVT. Per prevenire l'*unfolding* della proteina durante questa fase, i carboni alfa sono stati vincolati armonicamente con una costante di forza di 5 kcal mol^{-1}Å^{-2}.

La temperatura è stata controllata con il termostato di Langevin con frequenza di collisione $\gamma = 0.75$ ps^{-1}.

Equilibrazione in NPT

Segue una fase di equilibratura in *ensemble* isotermo-isobarico (NPT), mantenendo la temperatura a 300 K attraverso il termostato di Langevin e regolando la pressione mediante un barostato di Berendsen isotropo (3.33). Il passo d'integrazione è stato mantenuto pari a 2 fs.

Al fine di preservare l'integrità strutturale della proteina durante l'equilibratura, è stato implementato un protocollo di rilascio graduale dei vincoli armonici applicati ai carboni alfa:

- Primo *step* (20 ps, 10.000 *steps*): vincoli con costante di forza $k = 3.75$ kcal mol^{-1}Å^{-2};
- Secondo *step* (20 ps, 10.000 *steps*): vincoli con costante di forza $k = 1.75$ kcal mol^{-1}Å^{-2};
- Terzo *step* (1 ns, 500.000 *steps*): completa rimozione dei vincoli, che costituisce la fase di pre-produzione del sistema.

Questo, consente di controllare la stabilità del campionamento, attraverso il monitoraggio delle serie temporali di temperatura, pressione, volume, accertando l'assenza di *drift* sistematici e la compatibilità delle fluttuazioni con l'*ensemble* termodinamico.

In questa fase si sono inoltre confermate l'integrità del *box* periodico (assenza di vuoti) e la corretta conservazione delle grandezze vincolate.

3.2.4 Produzione e repliche

Le fasi iniziali di minimizzazione e di equilibrazione del solvente precedentemente discusse sono state condotte utilizzando il modulo SANDER di AMBER, ottimizzato per l'esecuzione su CPU.

Per le fasi successive, computazionalmente più onerose, come il riscaldamento del sistema e la finale equilibrazione in NPT, è stato invece impiegato il modulo PMEMD.CUDA GPU-ACCELERATED, che sfrutta il calcolo parallelo su GPU.

Raggiunta la stabilità termo-barica, si è proceduto con la fase di produzione, consistente nell'integrazione numerica delle equazioni del moto. Il controllo della temperatura ($T_0 = 300$ K) è stato gestito tramite un

[8] Per ogni particella di massa m_i, le componenti di velocità sono campionate da una distribuzione gaussiana $\sim N(0, k_B T/m_i)$; in tre dimensioni, il modulo della velocità segue la distribuzione $f(v) = 4\pi\left(\frac{m_i}{2\pi k_B T}\right)^{3/2} v^2 e^{-m_i v^2/(2k_B T)}$.

termostato di Langevin ($\gamma = 1$ ps^{-1}), mentre per il controllo della pressione ($P_0 = 1$ atm) è stato utilizzato un barostato di Berendsen isotropo con un tempo di rilassamento di 1 ps.

Le simulazioni sono state eseguite in NPT per una durata di 1000 ns ciascuna, impiegando un passo di integrazione $\Delta t = 2$ fs. Per il trattamento delle interazioni non covalenti (Equazioni 3.10 e 3.11), è stato applicato un *cutoff* di 8 Å.

Ad ogni passo della dinamica, le forze agenti sugli atomi vengono calcolate secondo il campo di forze impostato, le velocità sono aggiornate mediante l'algoritmo di Størmer–Verlet nella sua variante *velocity* (Equazioni 3.22, 3.23, 3.24) e, infine, vengono determinate le nuove posizioni atomiche.

Al fine di ottenere un campionamento statisticamente robusto e caratterizzare adeguatamente l'evoluzione conformazionale del sistema, in particolare per le regioni ipervariabili (CDR1-3) dei *nanobodies*, sono state generate quattro repliche indipendenti per ogni proteina. Ciascuna replica, pur condividendo identici parametri fisici, condizioni al contorno e durata, è stata inizializzata con un insieme distinto di velocità atomiche, campionato da una distribuzione di Maxwell-Boltzmann a 300 K utilizzando *random seeds* differenti. Questo protocollo permette di verificare la riproducibilità dei risultati, mediare il comportamento stocastico della MD e fornire una stima dell'incertezza statistica per l'estrazione dei descrittori.

Delle traiettorie prodotte, della durata complessiva di 1 μs (1000 ns) per replica, sono state salvate le coordinate atomiche con un intervallo di campionamento $\Delta t_{\text{out}} = 50$ ps (corrispondente a 25000 *steps*), per un totale di 20000 *frame* per replica e, considerando tutte le repliche, 80000 *frame* complessivi per ogni *nanobody*.

Al termine della produzione, infatti, le traiettorie delle quattro repliche indipendenti sono state concatenate in un'unica traiettoria cumulativa della durata complessiva di 4000 ns (4 μs) utilizzando il *tool* CPPTRAJ [53].

Prima delle analisi successive, le traiettorie concatenate sono state sottoposte a un'operazione di *fitting* e *centering* per rimuovere gli artefatti dovuti all'utilizzo delle condizioni periodiche al contorno (PBC) e per eliminare le rototraslazioni globali del sistema.

Tabella 3.2: Parametri globali adottati nelle simulazioni di MD.

Parametro	**Simbolo**	**Valore / Impostazione**
Ensemble impiegati	–	NVT (equilibrazione solvente, riscaldamento); NPT (equilibrazione, produzione)
Temperatura *target*	T_0	300 K (equilibrazione solvente 25 $\rightarrow$ 400 K)
Pressione	P_0	1 atm (in NPT)
Time–*step*	Δt	1 fs (equilibrazione solvente); 2 fs (riscaldamento sistema/equilibrazione/produzione)
Termostato	–	Langevin, $\gamma = 0.75$–1 ps^{-1}
Barostato	–	Berendsen, $\tau_p = 1$ ps
Vincoli SHAKE	–	Attivo sui legami con H (da riscaldamento in poi)
Cutoff interazioni non–covalenti	r_c	8 Å (Lennard–Jones e Coulomb)
Numero repliche	–	4 (*random seeds*)
Durata produzione (per replica)	–	1000 ns
Motore MD	–	SANDER (pre–produzione); PMEMD.CUDA (equilibrazione/produzione)

4 Metodi di *Machine Learning* e *Deep Learning*

La prima parte è dedicata alle analisi delle traiettorie di MD, in particolare all'estrazione di descrittori dalle traiettorie.

La seconda parte descrive il *workflow* di metodi di apprendimento supervisionato per la classificazione dei *nanobodies* sulla base di descrittori precedentemente estratti.

4.1 Estrazione dei descrittori dalla Dinamica Molecolare

Il successivo *step* del lavoro consiste nell'analisi delle traiettorie di MD per estrarre variabili quantitative. Sebbene la simulazione di dinamica molecolare fornisca in *output* le coordinate atomiche, per integrare efficacemente metodologie di apprendimento supervisionato risulta necessario trasformare la traiettoria in un *set* di variabili che condensino l'informazione dinamica, preservandone simultaneamente l'interpretabilità fisica. Tali variabili, denominate descrittori, rappresentano quindi una riduzione informativa della traiettoria molecolare.

I descrittori estratti dalle traiettorie di dinamica molecolare possono essere classificati in due categorie principali:

(i) Descrittori per residuo: calcolati come valori mediati lungo l'intera traiettoria temporale, sono rappresentati da un valore scalare per ciascun amminoacido costituente il *nanobody*;

(ii) Descrittori di *time series*: evoluzione temporale di grandezze specifiche lungo l'intera durata della simulazione.

4.1.1 Descrittori per residuo

Tra i descrittori per residuo rientrano:

(i) *Average Distance Fluctuation* (DF): si ottiene da un'analisi che costruisce la *Distance Fluctuation matrix* $N \times N$, con N pari al numero di residui della proteina, ed è eseguita tramite lo *script* PYTHON `compute_df_matrix.py` [54]. Come descritto da Morra et al. [55], ciascun elemento DF_{ij} di questa matrice rappresenta la fluttuazione quadratica media della distanza tra i residui i e j lungo l'intera traiettoria. Da questa matrice, il descrittore *"df average"* per un singolo residuo i viene calcolato come la media di tutti gli elementi DF_{ij} della riga o, in modo equivalente, della colonna, poiché la matrice è simmetrica ($DF_{ij} = DF_{ji}$), corrispondente a quel residuo, escludendo i vicini sequenziali prossimi per evitare di catturare fluttuazioni locali della catena peptidica (tipicamente si esclude la banda $|i-j| \leq k$ con $k = 3$).

Indicando con $d_{ij}(t)$ la distanza tempo-dipendente tra gli atomi Cα dei residui i e j, si ha:

$$DF_{ij} \;=\; \Big\langle \big(d_{ij}(t) - \langle d_{ij}(t)\rangle_t\big)^2 \Big\rangle_t, \qquad d_{ij}(t) \;=\; \big\|\mathbf{r}_i^{C\alpha}(t) - \mathbf{r}_j^{C\alpha}(t)\big\|. \tag{4.1}$$

Per definire formalmente la media per residuo, si introduce una maschera che azzera gli elementi sulla diagonale e nella sua banda di ampiezza $2k{+}1$:

$$M_{ij}^{(k)} \;=\; \begin{cases} 0, & \text{se } |i-j| \leq k, \\ 1, & \text{altrimenti.} \end{cases} \tag{4.2}$$

Il descrittore scalare per il residuo i diventa quindi:

$$\overline{DF}_i^{(3)} = \frac{\sum_{j=1}^{N} M_{ij}^{(3)} DF_{ij}}{\sum_{j=1}^{N} M_{ij}^{(3)}} \tag{4.3}$$

In pratica, un $\overline{DF}_i^{(3)}$ basso identifica residui che si muovono in modo più concertato con il resto della struttura (esclusi i vicini locali), mentre un $\overline{DF}_i^{(3)}$ alto è caratteristico di residui che si muovono in maniera meno concertata rispetto al resto della proteina, e che quindi fanno parte di segmenti del *nanobody* più proni al cambiamento conformazionale [55–57].

(ii) *Root Mean Square Fluctuation* (RMSF): quantifica la flessibilità locale di ciascun residuo lungo la simulazione, misurando quanto l'atomo Cα (carbonio alfa) del residuo i si discosta, in media quadratica, dalla propria posizione media. Per ogni residuo i si calcola:

$$\mathrm{RMSF}(i) = \sqrt{\frac{1}{T}\sum_{t=1}^{T} \left\| \mathbf{r}_i(t) - \langle \mathbf{r}_i \rangle \right\|^2} \tag{4.4}$$

dove $\mathbf{r}_i(t)$ è la posizione dell'atomo Cα del residuo i al frame t e $\langle \mathbf{r}_i \rangle$ è la sua posizione media sulla traiettoria di durata T. Valori elevati di RMSF indicano residui più mobili o meno vincolati; valori bassi denotano regioni più rigide o stabilizzate. L'analisi è stata eseguita con `cpptraj` (AmberTools/AMBER24).

(iii) Contributo al Primo Autovettore (REBELOT): descrittore derivante da un'analisi di decomposizione spettrale applicata alla matrice energetica M_{ij} ottenuta da calcoli MM/GBSA sulle strutture rappresentative dei *cluster* più popolati [31].

L'utilizzo strutture maggiormente rappresentative (tipicamente i centroidi, che forniscono una stima significativa dell'intero spazio conformazionale) si basa su una consolidata metodologia di analisi delle traiettorie di dinamica molecolare, dove i *frame* vengono raggruppati in *cluster* in base alla loro similarità strutturale. I *cluster* più popolati rappresentano gli stati conformazionali più stabili e statisticamente rilevanti della proteina.

In questo lavoro, il metodo di *cluster* adottato è il *clustering* gerarchico, *single linkage* e *bottom up*[1].

Il workflow operativo segue la metodologia MLCE/REBELOT [58–61] e prevede:

(a) preparazione di ciascuna proteina con `tleap` (generazione dei file `.prmtop` e `.inpcrd`);

(b) minimizzazione in solvente implicito[2] con `sander` (200 passi di *steepest descent*);

(c) valutazione della parte non–covalente del potenziale tramite decomposizione MM/GBSA con lo *script* PYTHON `MMPBSA.py`, ottenendo, per ogni proteina di N residui, una matrice simmetrica $N \times N$ di interazione:

$$M_{ij} = E_{ij}^{\mathrm{vdW}} + E_{ij}^{\mathrm{elec}}, \tag{4.5}$$

dove i contributi di solvatazione sono inclusi nel bilancio energetico complessivo ma, per costruzione, non vengono decomposti in termini espliciti residuo–residuo.

La matrice riportata in 4.5 rappresenta le interazioni energetiche residuo-residuo, e viene diagonalizzata

[1] Il *clustering* gerarchico *bottom-up* (agglomerativo) inizia considerando ogni punto dati come un *cluster* separato e successivamente fonde iterativamente i *cluster* più vicini fino a formare un unico *cluster* contenente tutti i punti. Il criterio *single linkage* (collegamento singolo) definisce la distanza tra due *cluster* come la distanza minima tra due punti appartenenti a *cluster* diversi.

[2] Per "solvente implicito" si intende un modello del solvente nel quale non si considera l'interazione fra le singole molecole d'acqua e ioni con gli atomi della proteina, come nel caso esplicito, ma si approssima il sistema come un solvente dielettrico uniforme.

secondo la relazione proposta dagli Autori:

$$M_{ij} = \sum_{\alpha=1}^{N} \lambda_\alpha v_i^\alpha v_j^\alpha, \tag{4.6}$$

dove λ_α sono gli autovalori e v^α i corrispondenti autovettori. Nella matrice M_{ij}, gli indici i e j rappresentano rispettivamente il residuo i-esimo e il residuo j-esimo della proteina. La decomposizione spettrale della matrice M_{ij} permette di scomporre l'informazione energetica della proteina in modi fondamentali di interazione, che matematicamente corrispondono proprio agli autovettori della matrice. Ciascuna componente v_i^α quantifica il grado di partecipazione del residuo i-esimo al α-esimo modo di interazione.
Ordinando gli autovalori dal più negativo al più positivo, gli Autori hanno mostrato che il primo modo (λ_1, v^1) cattura la quota principale delle interazioni stabilizzanti del sistema [58–61]. Il contributo di ciascun residuo i al primo autovettore è dato dalla componente v_i^1.
Si definisce quindi la matrice energetica approssimata dominata dal primo modo:

$$\widetilde{M}_{ij} \;=\; \lambda_1 \, v_i^1 \, v_j^1. \tag{4.7}$$

In accordo con la metodologia proposta da Capelli et al. [58], è stata adottata una soglia teorica di riferimento pari a $1/\sqrt{N}$, dove N rappresenta la lunghezza totale in termini di residui della proteina. Tale soglia permette di classificare i residui in base al loro accoppiamento energetico: i residui con valori di v_i^1 superiori a questa soglia sono considerati altamente energetici, indicando un forte accoppiamento con il *framework* proteico e un ruolo predominante nel mantenimento della stabilità strutturale. Al contrario, i residui con valori inferiori alla soglia sono considerati debolmente accoppiati, e quindi potenzialmente disponibili per interazioni con ligandi esterni. Questo approccio, consente di identificare le regioni strutturalmente critiche e quelle funzionalmente flessibili sulla base del loro accoppiamento energetico con il resto della proteina.
Dato che la struttura tridimensionale è nota, si costruisce anche la matrice di contatto C_{ij} ponendo due residui in contatto se almeno una coppia di atomi pesanti è a distanza $< 6\,\text{Å}$. Il prodotto di Hadamard[3] tra $\widetilde{M}$ e C fornisce la *Matrix of Low Coupling Energies* (MLCE):

$$M_{ij}^{\text{MLCE}} \;=\; \widetilde{M}_{ij} \odot C_{ij}, \tag{4.8}$$

che integra prossimità strutturale e intensità di accoppiamento energetico locale, consentendo di evidenziare sotto–strutture a debole accoppiamento con il resto della proteina (definite *patches*).
Per i *nanobodies*, le *patches* identificate con MLCE tendono a sovrapporsi alle CDR, coerentemente con la loro propensione a rimodellamenti conformazionali e al riconoscimento dell'antigene [59–61].
Codice e istruzioni MLCE sono disponibili su GitHub [58].

Per ciascun *nanobody* sono stati prodotti 3 *file* `.txt` contenenti, per ogni residuo, due colonne: nella prima l'indice del residuo stesso, e nella seconda il valore scalare relativo a quel descrittore.

4.1.2 Predizione delle CDRs

In conformità con quanto introdotto nella Sezione 1.1.3, riguardante la struttura dei *nanobodies* e il ruolo cruciale delle CDR nel determinare la specificità di legame, la caratterizzazione di tali regioni costituisce un prerequisito fondamentale per le analisi successive tramite metodi di apprendimento supervisionato. Oltre ai descrittori , estratti dalle simulazioni di MD, è stata pertanto condotta un'analisi computazionale

[3] Moltiplicazione elemento per elemento tra matrici della stessa dimensione, che in notazione indiciale corrisponde: $(A \odot B)_{ij} = A_{ij}B_{ij}$.

completa della sequenza amminoacidica di ciascun *nanobody*, finalizzata all'identificazione precisa dei residui costituenti i loop CDR, i quali non sono annotati nelle strutture PDB estratte (Sezione 3.2.1).

A tal fine, è stato impiegato NANOCDR-X, un modello di *deep learning* sviluppato specificamente per assegnare, in modalità *sequence-only*, ciascun residuo della sequenza proteica di un *nanobody* a una delle quattro classi funzionali: *body* (il *framework* della proteina, ovvero la regione conservata) o uno dei tre loop ipervariabili (CDR1, CDR2, CDR3). Questo approccio, descritto in dettaglio in Bagordo e Trèves et al. [62], supera i limiti dei metodi tradizionali basati su *motif* conservati o schemi di numerazione, spesso inadeguati a catturare l'elevatissima variabilità strutturale dei domini V_HH, conseguenza dei processi di ricombinazione V(D)J (Sezione 1.1.4) e della notevola lunghezza del *loop* CDR3.

Operativamente, il *workflow* adottato nel lavoro prevede:

(i) *Input* della sequenza amminoacidica del dominio V_HH in formato `FASTA` ottenuto dal PDB, nel modello NANOCDR-X.

(ii) Esecuzione dell'inferenza *residue-wise* per ottenere il *labeling* di ogni amminoacido nelle classi *body*/CDR1/CDR2/CDR3.

(iii) Estrazione degli indici di inizio e fine per ciascuna regione CDR identificata.

(iv) Integrazione di questi dati annotativi all'interno di un *file* di metadati, associato alla sequenza `FASTA` di ciascun *nanobody*.

L'*output* di questa fase è costituito dalla sequenza originale arricchita con le annotazioni delle CDR.

4.1.3 Descrittori di *time series*

Per ogni *nanobody*, questi descrittori sono calcolati sull'intera metatraiettoria per i CDR3, risultato della concatenazione di tutte le traiettorie di simulazione prodotte (Sezione 3.2.4), definiti sempre utilizzando il modello NANOCDR-X (Sezione 4.1.2).

Questa tipologia di descrittori è classificabile in due categorie: serie temporali di contenuto di struttura secondaria e serie temporali di distanza.

Struttura secondaria

Per l'analisi *frame-by-frame* del contenuto di struttura secondaria è stato impiegato PLUMED 2.9, una libreria *open-source* che si integra con i principali motori di MD [63]. I descrittori estratti sono Alpha–RMSD e Anti-Beta-RMSD, che trasformano ciascun frame t in un valore scalare basato su distanze RMSD $\mathbf{r}_i(t)$[4] tra sottosequenze candidate e un elemento ideale di struttura secondaria, rispettivamente α-elica e foglietto β antiparallelo: questi, in particolare, sono stati calcolati sullo stesso intervallo di residui appartenenti al CDR3.

Dal punto di vista biologico, la comparsa di tratti di α-elica o di foglietto β nel CDR3 modulano caratteristiche quali rigidità locale, orientamento e presentazione dei residui del paratopo, con effetti su affinità e modalità di riconoscimento dell'epitopo, motivando quindi il loro impiego per quantificare conformazioni funzionalmente rilevanti. Di conseguenza, l'analisi combinata delle serie temporali di Alpha RMSD e Beta RMSD permette di caratterizzare l'evoluzione temporale delle tendenze di *folding* locale delle regioni CDR3, rivelando eventuali transizioni conformazionali durante la traiettoria [53].

Questo viene realizzato calcolando la seguente relazione, funzione delle distanze RMSD:

$$s(t) \;=\; \sum_i \frac{1-\left(\frac{\mathbf{r}_i(t)-d_0}{r_0}\right)^n}{1-\left(\frac{\mathbf{r}_i(t)-d_0}{r_0}\right)^m}, \tag{4.9}$$

[4] La RMSD (*Root Mean Square Deviation*) misura la deviazione media tra le posizioni atomiche di due strutture dopo allineamento ottimale. La formula generale è: $\text{RMSD} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\mathbf{r}_i^{\text{struttura A}} - \mathbf{r}_i^{\text{struttura B}}\right)^2}$, dove N è il numero di atomi considerati e $\mathbf{r_i}$ sono le coordinate atomiche.

dove $s(t)$ è un indicatore scalare di similarità strutturale al tempo t, ottenuto come somma sulle finestre i di funzioni *switching* applicate alle rispettive RMSD; $\mathbf{r}_i(t)$ è la RMSD della finestra i al tempo t rispetto al modello ideale. Per i successivi calcoli sono stati mantenuti i parametri di *default* $n = 8$, $m = 12$, $d_0 = 0$ e $r_0 = 0,08\,\text{nm}$.

In pratica, valori più alti di questi descrittori indicano maggiore similarità alla struttura ideale. In dettaglio:

(i) Alpha–RMSD: misura il contenuto di α-elica considerando tutte le possibili finestre di almeno sei residui contigui all'interno dell'insieme di residui specificato (qui applicato ai CDR3). Per ogni finestra si calcola la RMSD rispetto a una α-elica idealizzata (utilizzando gli atomi di *backbone*). Il valore restituito di *default* è il conteggio continuo $s(t)$ della (4.9), che aumenta quando un numero maggiore di finestre presenta caratteristiche elicoidali. Si noti che la RMSD per finestra diminuisce all'aumentare della similarità, mentre $s(t)$ aumenta.

(ii) Anti–Beta–RMSD: misura il contenuto di β foglietti antiparalleli considerando tutte le possibili finestre di almeno otto residui contigui all'interno dell'insieme di residui specificato (sempre applicato ai CDR3). Per ciascuna coppia si valuta la RMSD rispetto a un foglietto β antiparallelo ideale e si costruisce $s(t)$ come nella (4.9). Qualora il CDR3 contasse meno di otto amminoacidi, l'intervallo è stato esteso includendo residui adiacenti fino a raggiungere gli otto richiesti da Anti-Beta-RMSD. In pratica:

- per il *nanobody* 7FBK_20G6 sono stati aggiunti i residui 95 e 96 a valle del CDR3 predetto;
- per 8H5U_Nb-021 è stato aggiunto il residuo 97 a monte e i residui 103 e 104 a valle.

In tutti gli altri casi è stata utilizzata la sequenza CDR3 predetta.

Distanza

Descrive la distanza tra il residuo del CDR3 con l'*average* DF massimo (Equazione 4.3) e il centro di massa (COM) del *nanobody* (escludendo il CDR3 stesso dal calcolo del COM), calcolata per ciascuno degli 80,000 *frame* della dinamica molecolare. Formalmente:

$$d(t) = \|\mathbf{r}_{\max}(t) - \mathbf{r}_{COM}(t)\|\,. \tag{4.10}$$

Questo descrittore, fornisce informazioni sull'evoluzione temporale della fluttuazione relativa del CDR3 rispetto al corpo principale del *nanobody* (il *framework*), dove valori elevati in *frame* specifici indicano un maggiore allontanamento istantaneo e quindi una potenziale maggiore esposizione solvente e disponibilità per il riconoscimento dell'epitopo in quegli istanti temporali.

Per ciascun *nanobody* sono stati prodotti 2 *file* `.dat` così strutturati:

1. `alpha_antibeta.dat`: contenente tre colonne, di cui la prima indica il numero del *frame* (da 1 a 80,000), la seconda colonna riporta il valore del descrittore Alpha–RMSD, la terza colonna riporta il valore del descrittore Anti–Beta–RMSD per ciascun *frame*.
2. `distance.dat`: contenente due colonne, di cui la prima colonna indica il numero del *frame* (da 1 a 80,000), la seconda colonna riporta il valore della distanza CDR3-COM per ciascun *frame*.

4.2 Descrittori estratti dalle simulazioni di Dinamica Molecolare

Di seguito sono riportati alcuni esempi dei descrittori estratti dalle traiettorie di alcuni alcuni dei *nanobodies* selezionati.

Tutte le analisi di seguito riportate, con annessi *plot*, sono state eseguite in locale su un *MacBook Air* (M4, 2025), dotato di SoC Apple M4 (CPU 10-core), GPU a 10-core; l'ambiente di lavoro adottato è PYTHON (v. 3.12.1) in VISUAL STUDIO CODE. Le librerie impiegate sono di seguito riportate: `pandas` (v2.2.3), `numpy` (v1.26.4), `matplotlib` (v3.10.1).

Le predizioni dei residui costituenti *core* proteico e CDR3 (in Figura 4.1 è riportato un esempio), sono state impiegate per definire le *features* di composizione amminoacidica, come descritto nella Sezione 4.3.1.

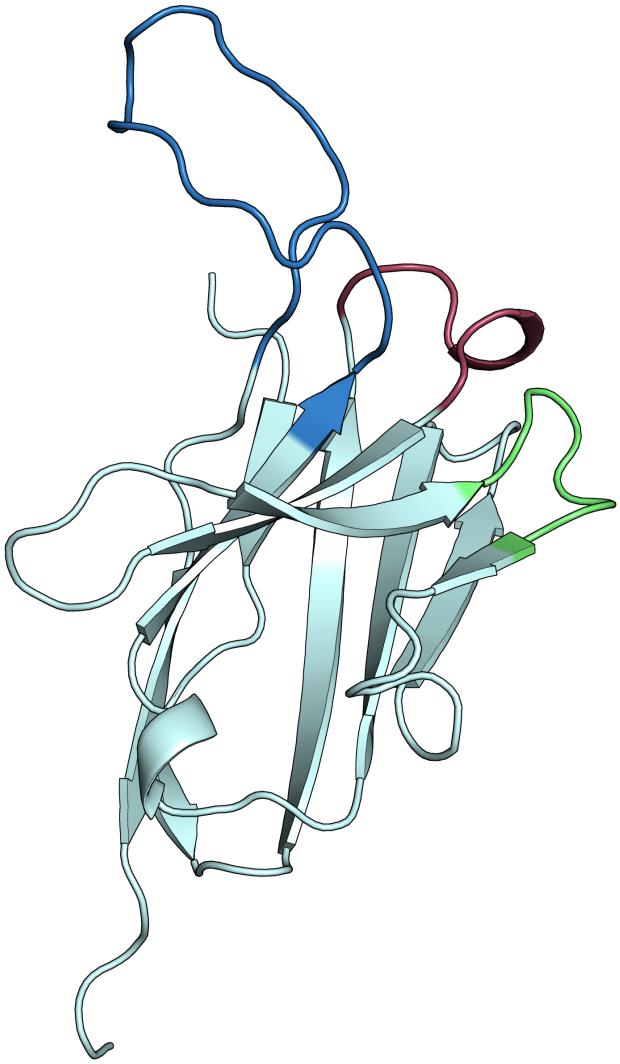

Figura 4.1: Rappresentazione *ribbon* del *nanobody* 7tpr 8a2 (ortosterico). In rosso sono evidenziati gli amminoacidi del CDR1, in verde del CDR2, e infine in blu del CDR3. Le regioni non colorate costituiscono il *framework* della proteina. Tutti questi residui sono stati predetti usando il modello NANOCDR-X [62]. L'immagine mostra una chiara estensione maggiore del *loop* del CDR3 rispetto agli altri.

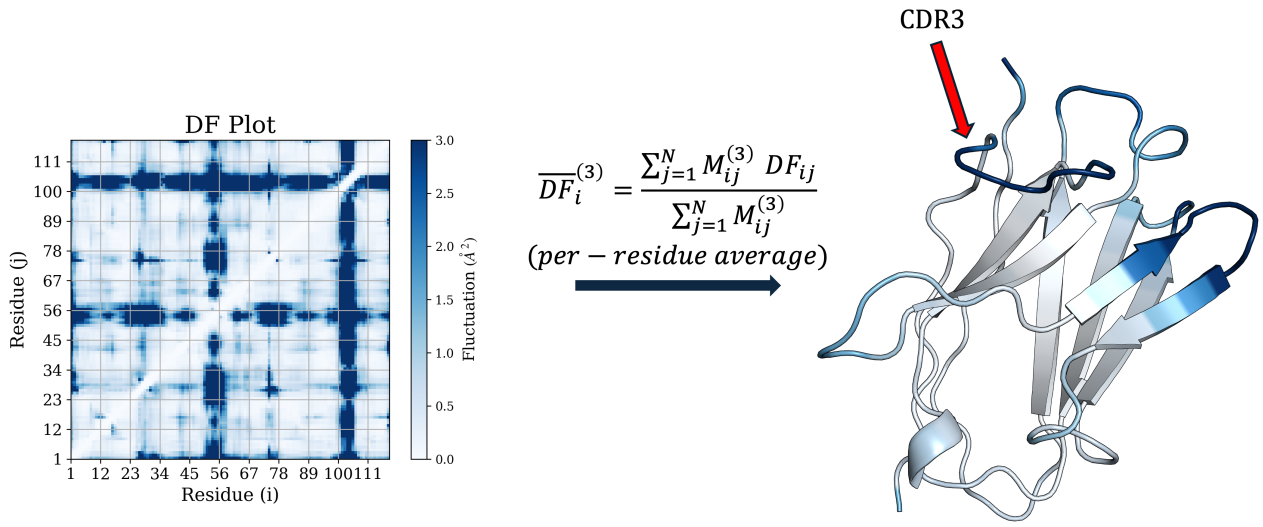

Figura 4.2: Esempio di rappresentazione del descrittore per residuo *Average DF* proiettato in scala di colore sul *nanobody* NB6zxn_Ty1 (ortosterico).

Come mostrato in Figura 4.2, il pannello di sinistra mostra la matrice DF $N \times N$ in cui ciascun elemento DF_{ij} misura la fluttuazione quadratica media della distanza Cα–Cα tra i residui i e j lungo tutta la traiettoria (Eq. 4.1); il pannello di destra mostra la proiezione di $\overline{DF}_i^{(3)}$ (Equazioni 4.2 e 4.3) sulla struttura 3D (*ribbon*) di un *nanobody*: tonalità più chiare in scala di colore indicano un $\overline{DF}_i^{(3)}$ basso, ovvero residui che si muovono in modo più concertato con il resto della struttura (esclusi i vicini locali), mentre un $\overline{DF}_i^{(3)}$ alto, indicato con un colore maggiormente scuro, è caratteristico di residui che fanno parte di segmenti del *nanobody* più predisposti al cambiamento conformazionale. Da tale proiezione emerge che i residui del *loop* CDR3 (indicati dalla freccia rossa) sono quelli che presentano un $\overline{DF}_i^{(3)}$ maggiore, come evidente dal colore scuro, a indicare che il loro moto è meno coordinato rispetto a tutti gli altri residui.

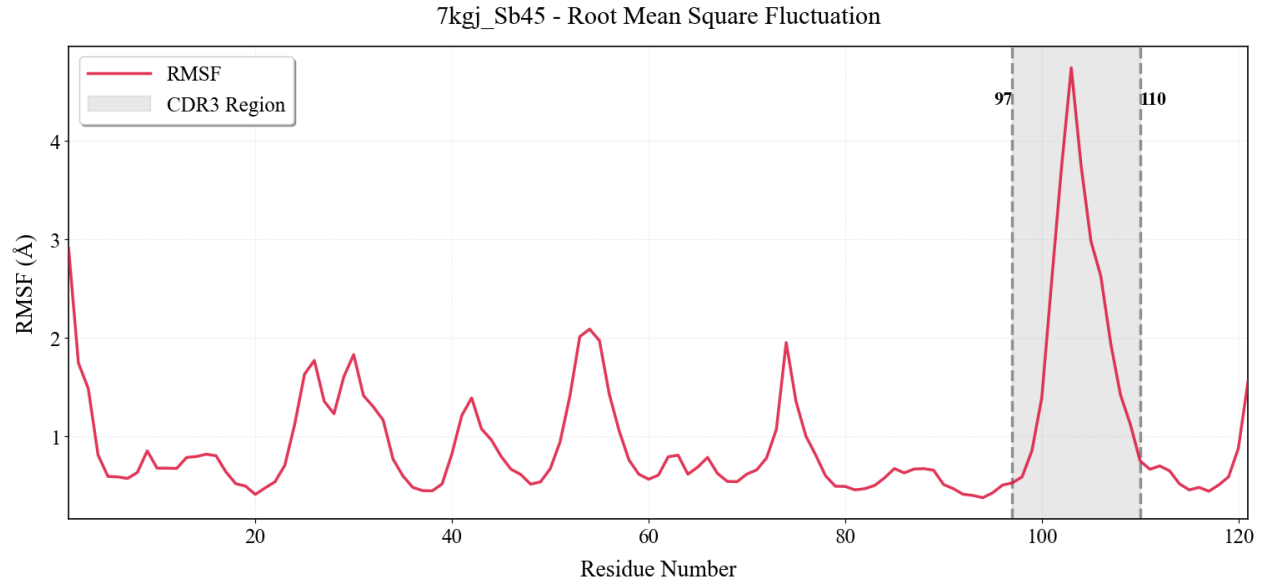

Figura 4.3: Andamento del descrittore RMSF nel caso del *nanobody* 7kgj_Sb45 (ortosterico).

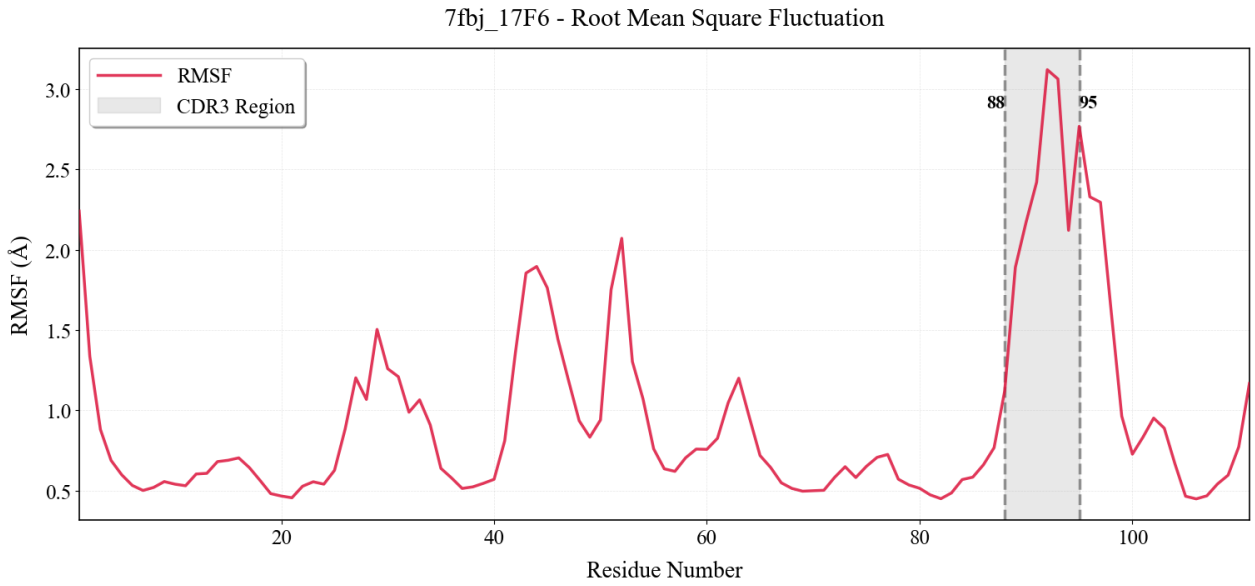

Figura 4.4: Andamento del descrittore RMSF nel caso del *nanobody* 7fbj_17F6 (allosterico).

Le Figure 4.3 e 4.4 riportano un esempio dell'andamento del descrittore RMSF, definito in (4.4). Sull'asse delle ascisse è rappresentata la posizione relativa dei residui amminoacidici, con particolare evidenza per quelli della CDR3 predetta, mentre sull'asse delle ordinate è riportata la fluttuazione quadratica media. Nel complesso, valori di RMSF più elevati denotano residui maggiormente mobili lungo l'intera traiettoria, mentre valori più bassi indicano una mobilità ridotta.

I *plot* confermano la tendenza dei residui della CDR3 a presentare valori elevati di RMSF, in virtù della loro maggiore lunghezza, confermando la loro maggiore mobilità e il minor grado di vincolo alla struttura.

Tuttavia, il confronto tra le classi evidenzia profili complessivamente confrontabili in cui non si osservano differenze significative né una separazione netta tra campioni ortosterici e allosterici.

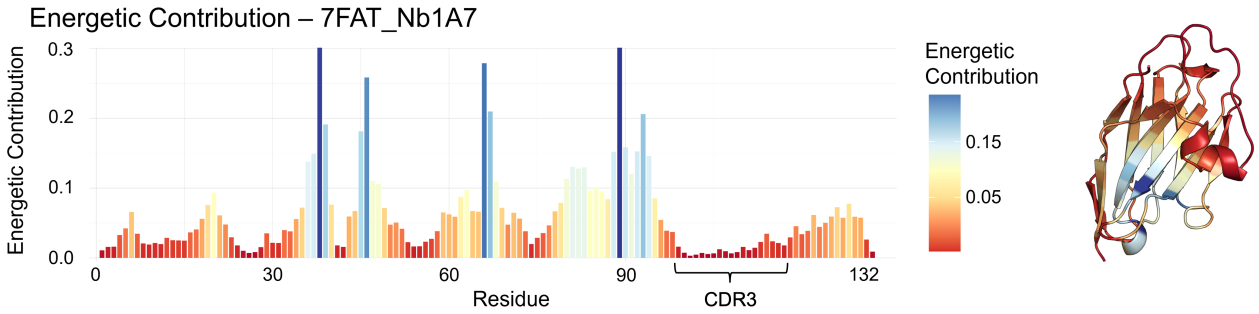

Figura 4.5: Esempio di contributo energetico per residuo al primo autovettore (*workflow* MLCE/REBELOT) proiettato in scala di colore sul *nanobody* 7fat_Nb1A7 (allosterico).

La Figura 4.5 rappresenta in ascissa la posizione relativa dei residui nel *nanobody*, e in ordinata il contributo energetico del primo autovettore derivante dalla decomposizione spettrale di REBELOT (Equazioni 4.6 e 4.7). Il grafico differenzia in blu i residui con contributo elevato, superiore alla soglia teorica $1/\sqrt{N}$ [58], e in rosso quelli a basso contributo (sotto soglia).

I risultati evidenziano come i residui con valori di v_i^1 (4.7) superiori alla soglia siano energeticamente rilevanti, denotando un forte accoppiamento con il *framework* proteico e un ruolo cruciale nel mantenimento della stabilità strutturale. Al contrario, i residui con contributi inferiori alla soglia presentano un debole accoppiamento, suggerendo una potenziale disponibilità per interazioni con ligandi esterni.

Dal *plot* si osserva come gli amminoacidi della CDR3 (evidenziati in ascisse) e, più in generale, tutte le regioni CDR presentino contributi estremamente bassi. Ciò indica il loro disaccoppiamento dal *backbone* del *nanobody*, caratteristica compatibile con una predisposizione al riconoscimento paratopo-epitopo.

La stessa scala cromatica è stata proiettata sulla struttura tridimensionale.

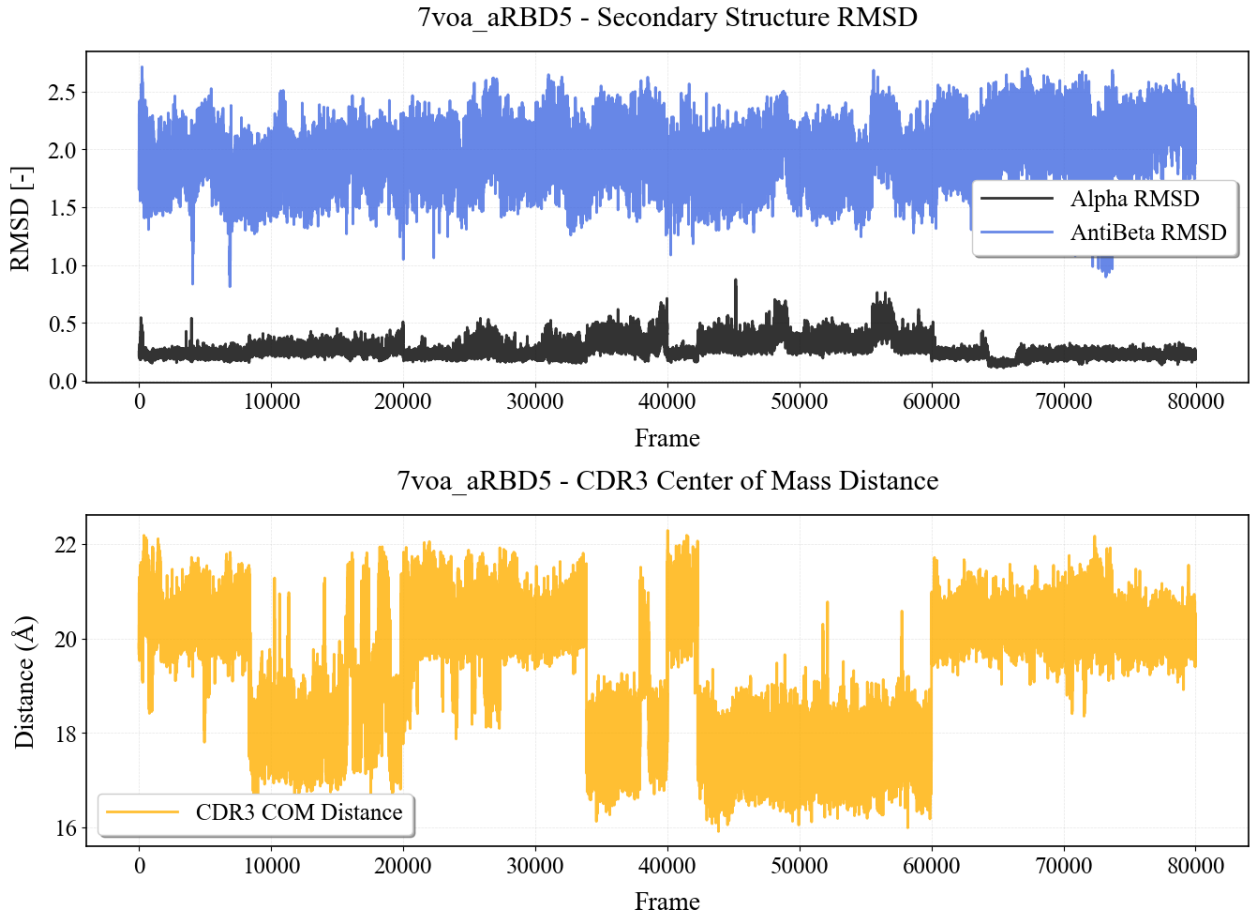

Figura 4.6: Serie temporali per il *nanobody* 7voa_aRBD5 (ortosterico).

Il pannello superiore mostra i punteggi di similarità di struttura secondaria calcolati tramite la trasformazione riportata in (4.9). Il tracciato *Alpha–RMSD* (nero) rimane sistematicamente più basso dell'*Anti–Beta–RMSD* (blu) lungo tutti gli 80,000 frame: l'andamento dell'Anti–Beta–RMSD oscilla su valori alti con soli cali transitori, mentre l'Alpha–RMSD resta vicino allo zero con rare fluttuazioni di breve durata. Questo *pattern*, specifico il *nanobody* selezionato come esempio, indica una debole propensione elicoidale (α-elica) locale del CDR3 e una tendenza ad organizzarsi in foglietti β antiparalleli. L'assenza di intersezioni fra i due segnali esclude eventi di *switch* di struttura secondaria.

Il pannello inferiore riporta la distanza istantanea $d(t)$ (4.10) tra il residuo del CDR3 con massimo *average* DF e il centro di massa del *nanobody*. La serie evidenzia un moto di *breathing* del CDR3 rispetto al *framework*, con fenomeni di avvicinamento al corpo proteico seguiti da fasi di riesposizione. Questo risultato, dimostra un comportamento dinamico particolarmente accentuato, specifico del CDR3, ed evidente in questo specifico *nanobody*, il cui moto esprime un allontanamento frequente durante la traiettoria simulata, suggerendo una maggiore esposizione al solvente, e quindi una disponibilità al riconoscimento del ligando.

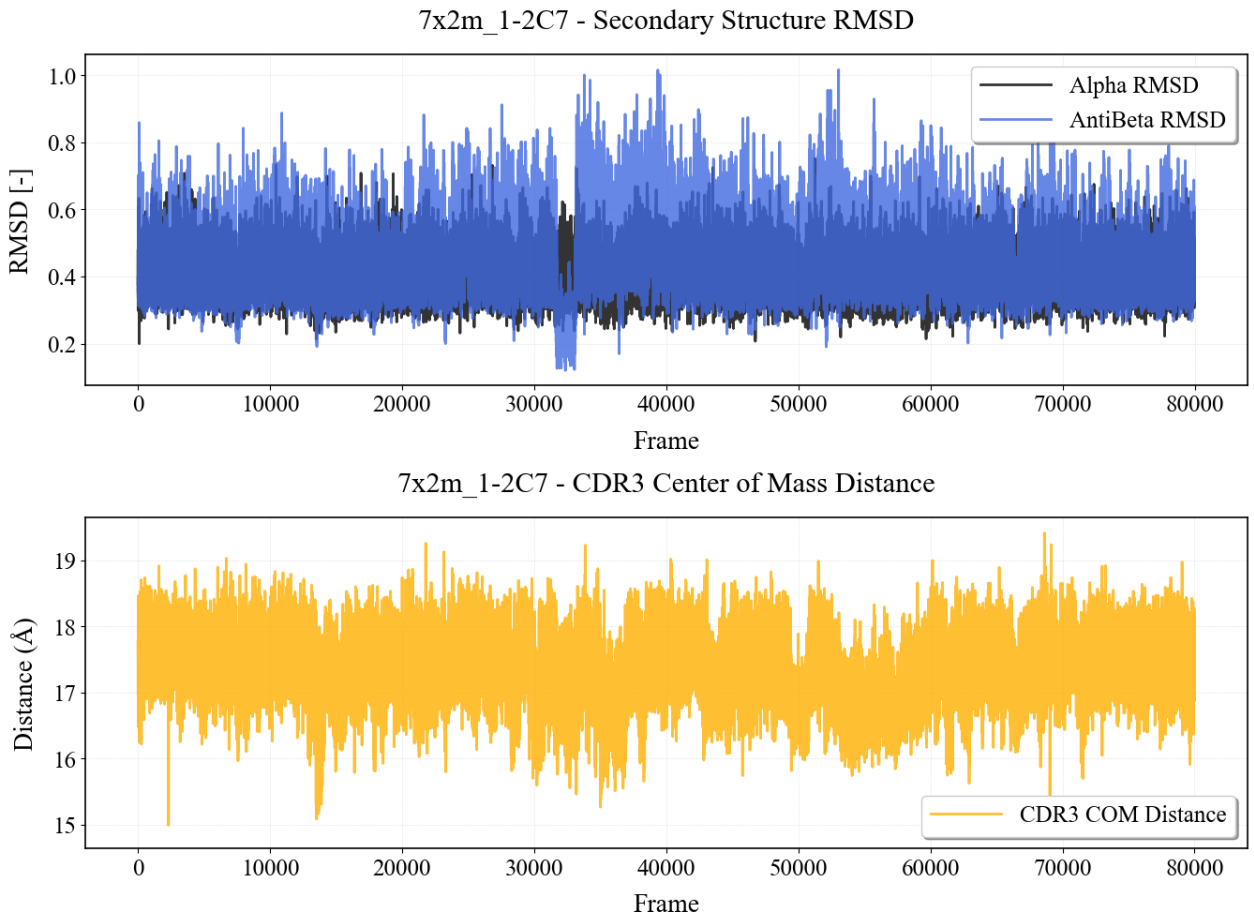

Figura 4.7: Serie temporali per il *nanobody* 7x2m_1-2C7 (allosterico).

La Figura 4.7 illustra un caso particolare che si discosta dal comportamento precedentemente osservato.

Nel pannello superiore, è presentata una configurazione incerta, in cui gli *score* di propensione alla formazione di un'α-elica e di un foglietto-β antiparallelo risultano sovrapposti e quindi confrontabili, indicando una indeterminatezza conformazionale; allo stesso tempo, questo andamento mostra uno *score* basso in termini di similarità conformazionale (il punteggio di similarità si mantiene su valori sotto 1, mentre nel caso mostrato Figura 4.6 si mantiene su valori più alti, superando anche picchi di 2.5).

Nel pannello inferiore, a differenza di quanto osservato nella Figura 4.6, non si verifica un cambiamento conformazionale significativo del moto del CDR3 che, come si evidenzia dalla stabilità della distanza $d(t)$, non mostra *breathing* dinamico.

Si sottolinea come gli esempi rappresentati non siano caratteristici delle due classi di riferimento e, di conseguenza, non possano essere considerati rappresentativi di un comportamento caratteristico prettamente ortosterico o allosterico.

4.3 *Workflow* di Apprendimento Supervisionato

4.3.1 *Feature Engineering*

Per l'addestramento dei modelli di *supervised learning* è stato costruito un *dataset* progettato secondo uno schema concettuale diviso in tre famiglie di *features*, corrispondenti proprio ai descrittori ricavati dalle traiettorie di MD (per-residuo e *time series*), integrati con la predizione strutturale dei residui costituenti le CDR. Questi, opportunamente adattati alla natura tabellare di un *dataset*, forniscono una rappresentazione tra loro ortogonale e complementare del *nanobody*. Di seguito sono riportate le tre famiglie:

(i) *Features* strutturali e amminoacidiche, che rappresentano la composizione chimica e forniscono informazioni sulla struttura primaria dei *nanobodies*;

(ii) *Features* per residuo;

(iii) *Features* di *time series*.

Per massimizzare il segnale informativo, e al contempo contenere il rumore, l'analisi è stata circoscritta agli amminoacidi del *framework* (il *backbone* del *nanobody*, ovvero gli amminoacidi etichettati come *body* da NANOCDR-X [62] che costituiscono le regioni conservate della proteina) e alla regione CDR3 (sempre annotate grazie a NANOCDR-X). Tale scelta riflette l'evidenza biologica che il CDR3, congiuntamente anche al *framework*, contribuisce in modo predominante alla definizione del paratopo, e quindi alle interazioni con l'epitopo [11]. Le CDR1 e CDR2, pur strutturalmente annotate, apportano un contributo minore all'interazione, e sono state escluse per ridurre ridondanze e variabilità non utile al problema di classificazione binario.

Features strutturali e amminoacidiche

Questa famiglia di *features* descrive le proprietà chimiche della proteina, basandosi sulla sua costituzione amminoacidica. Il suo scopo fondamentale è identificare *pattern* strutturali specifici, quali la sovraespressione di un determinato residuo amminoacidico (o di un'intera classe di amminoacidi), così come la presenza di motivi ripetuti in una delle due classi.

Questa analisi è stata eseguita nelle due regioni selezionate, ovvero *framework* e CDR3. Per rispettare il formato tabulare dei modelli di *machine learning*, sia L la lunghezza totale della regione *framework*/CDR3 (numero di amminoacidi), sono state calcolate le seguenti *features*:

(i) Frequenza dei singoli amminoacidi: 20 *features* che rappresentano la frequenza relativa di ciascun amminoacido aa_i nella regione considerata, normalizzata rispetto alla lunghezza totale della regione (es: `_cdr3_freq_Y`). In formule:

$$\text{freq\_}aa_i = \frac{\text{Conteggio}(aa_i)}{L} \tag{4.11}$$

(ii) Carica media della regione: si considerano esclusivamente le catene laterali degli amminoacidi costituenti; grazie a tale *feature* si quantifica il potenziale elettrostatico medio della regione, parametro che interviene nelle interazioni proteina-proteina [64]. I valori di carica sono assegnati secondo la scala: Arginina (R) e Lisina (K) = +1; Acido aspartico (D) e Acido glutammico (E) = -1; Istidina (H) = +0.5; tutti gli altri = 0:

$$\overline{Q} \;=\; \frac{1}{L}\sum_{k=1}^{L} q(a_k), \qquad q(a) = \begin{cases} +1 & a \in \{R,K\}, \\ -1 & a \in \{D,E\}, \\ +0.5 & a = H, \\ 0 & \text{altrimenti.} \end{cases} \tag{4.12}$$

(iii) Volume medio della regione: determinato attraverso i volumi delle catene laterali degli amminoacidi costituenti la regione, descrive l'ingombro sterico medio della regione. I volumi atomici sono espressi in ^{3} e coprono un *range* da 0.0 (Glicina) a 167.8 (Triptofano) [64]:

$$\overline{V} \;=\; \frac{1}{L}\sum_{k=1}^{L} v(a_k) \;\; [\text{Å}^3]. \tag{4.13}$$

(iv) Idrofobicità media: calcolata utilizzando la scala di Kyte-Doolittle[5] [65], questa *feature* misura la tendenza media della regione ad interagire con ambienti idrofobici. I valori variano da -4.5 (Arginina, più idrofilica) a +4.5 (Isoleucina, più idrofobica):

[5]La scala di idrofobicità di Kyte-Doolittle assegna un valore numerico a ciascun amminoacido basato sulle proprietà idrofobiche della catena laterale. Valori positivi indicano idrofobicità, valori negativi indicano idrofilicità.

$$\overline{H} \;=\; \frac{1}{L}\sum_{k=1}^{L} \mathrm{KD}(a_k). \tag{4.14}$$

(v) Frequenza delle classi funzionali: insieme di *features* che rappresentano la frequenza di amminoacidi appartenenti a classi funzionali specifiche, anch'esse normalizzate rispetto alla lunghezza della regione. Sia C una delle seguenti classi funzionali:

- Aromatici: Fenilalanina (F), Triptofano (W), Tirosina (Y).
- Carichi positivamente: Arginina (R), Lisina (K), Istidina (H).
- Carichi negativamente: Acido aspartico (D), Acido glutammico (E).
- Polari non carichi: Asparagina (N), Glutammina (Q), Serina (S), Treonina (T).
- Idrofobici: Alanina (A), Isoleucina (I), Leucina (L), Valina (V), Fenilalanina (F), Metionina (M), Triptofano (W), Glicina (G), Prolina (P).

Allora:

$$\mathrm{freq}_C \;=\; \frac{1}{L}\sum_{k=1}^{L} \mathbf{1}_{\{a_k \in C\}}, \tag{4.15}$$

dove freq_C è la frazione (compresa tra 0 e 1) di residui della regione che ricadono nella classe funzionale C. La $\mathbf{1}_{\{a_k \in C\}}$ è una funzione che assegna una *flag* che vale 1 se l'amminoacido in posizione k appartiene alla classe C, e 0 altrimenti.

(vi) Entropia di Shannon [66]: quantifica la variabilità della composizione amminoacidica della regione, in termini di misura della diversità sequenziale. Valori elevati di entropia indicano una maggiore eterogeneità nella composizione amminoacidica.

$$H \;=\; -\sum_{i=1}^{20} p_i \log_2 p_i, \qquad \text{con } p_i \equiv \mathrm{freq\_}aa_i. \tag{4.16}$$

Features di dinamica molecolare per residuo

I descrittori per residuo (Sezione 4.1.1) presentano una dimensionalità variabile a causa della diversa lunghezza delle sequenze amminoacidiche dei *nanobodies*, sia in termini di *framework*, sia delle loro regioni CDR3 [67]. Questa eterogeneità nella lunghezza, rende tali descrittori incompatibili con i modelli di *machine learning*, i quali richiedono un numero fisso e coerente di *features* per ogni campione. Per ovviare a questa limitazione, è stato adottato un approccio basato su metriche aggregate di natura statistica, che permette di normalizzare la struttura dei dati e renderli così adatti per le attività di classificazione. In particolare, sono state calcolate le seguenti metriche:

(i) Misure di tendenza centrale: Media, mediana.
(ii) Misure di dispersione: Deviazione standard, percentili (25° e 75°).
(iii) Misure di forma distributiva: *skewness* (asimmetria), *kurtosis* (curtosi).
(iv) Misure di estremi: Valore massimo, rapporto massimo/media.
(v) *Peak Analysis*: Conteggio dei picchi e prominenza dei picchi[6].
(vi) Norme vettoriali: Norma L1 (somma dei valori assoluti), norma L2 (radice quadrata della somma dei quadrati), norma infinito (valore massimo assoluto) applicate ai contributi degli autovettori del metodo REBELOT (4.7).

In aggiunta, è stato introdotto un raggruppamento basato sulla soglia teorica $t = 1/\sqrt{N}$ proposta da Capelli et al. [58], dove N rappresenta la lunghezza totale in residui del *nanobody*. Questa metodologia permette di suddividere ciascuna regione strutturale (CDR3 e *framework*) in sotto-regioni distinte, in base al loro profilo di accoppiamento energetico. Specificamente, vengono identificate sotto-regioni costituite esclusivamente dai

[6]La prominenza dei picchi misura l'altezza relativa di un picco rispetto alla linea di base circostante, indicando l'importanza locale del segnale nell'analisi delle distribuzioni.

residui con contributo al primo autovettore inferiore alla soglia t. Su questi sotto-insiemi di residui vengono quindi ricalcolate le metriche statistiche esposte precedentemente, permettendo di caratterizzare in modo mirato il comportamento dei domini funzionalmente più rilevanti dal punto di vista energetico. Questa analisi stratificata genera due categorie di *features* aggiuntive:

(i) Frazione sotto soglia: rapporto tra il numero di residui con $v_i^1 < t$ (4.7) e il numero totale di residui nella regione, che quantifica l'estensione relativa delle sotto-regioni debolmente accoppiate, e quindi più preminenti per l'inteazione con un ligando.

(ii) Statistiche aggregate per i residui sotto soglia: quando la frazione sotto soglia è diversa da zero, vengono ricalcolate tutte le metriche statistiche precedentemente descritte esclusivamente per questo sottoinsieme di residui (ad esempio: `cdr3_eigvec_l2_norm_below_threshold`). Questo approccio consente di catturare *pattern* specifici, quali la presenza di micro-domini potenzialmente coinvolti nel *binding* con il paratopo.

Infine, per ciascun descrittore per residuo, ovvero *average* DF (4.3), RMSF (4.4), contributo al primo autovettore con REBELOT, indicato per brevità nel *dataset* con `eigvec` (4.7), è stato calcolato il rapporto tra il valore medio nella regione CDR3 e il valore medio nel *framework*:

- `cdr3_fw_rmsf_ratio`
- `cdr3_fw_eigvec_ratio`
- `cdr3_fw_df_ratio`

Features di Time Series

La gestione dei descrittori di *time series* rappresenta un aspetto metodologico cruciale nel presente lavoro. Tali descrittori presentano una duplice natura: da un lato forniscono informazioni chimico-fisiche sul comportamento dinamico (*frame-by-frame*) della regione CDR3; dall'altro, consentono di superare il limite imposto dalla dimensione campionaria originale mediante una mirata strategia di *data augmentation*.

In quest'ottica, la scelta di simulare 50 *nanobodies* costituisce un compromesso ottimale, determinato non solo dai criteri di selezione strutturale e qualitativa, illustrati nella Sezione 3.2.1, ma anche da vincoli computazionali. L'intero protocollo, descritto nella Sezione 3.2, dalla preparazione del sistema alla produzione della traiettoria, richiede, per ogni *nanobody*, circa tre giorni di calcolo sul *cluster*. Questa notevole richiesta di risorse giustifica così il numero finale di campioni selezionati.

L'idea fondante di questo processo è quella per cui ciascun *frame* della traiettoria rappresenta uno *snapshot* unico del *nanobody* in una specifica conformazione e, pertanto, un potenziale campione aggiuntivo per l'addestramento dei modelli.

In analogia con l'approccio proposto da Frasnetti e Cucchi et al. [68], la procedura implementata si articola nei seguenti passaggi:

(i) *Preprocessing* tramite *shuffle*: per garantire l'indipendenza statistica dei campioni ed eliminare eventuali correlazioni temporali residue, i dati di *time series* sono stati sottoposti a uno *shuffle* preliminare completamente casuale a livello di *frame* prima dell'applicazione della strategia di *data augmentation*. In tal modo, si evita che la sequenzialità temporale introduca *bias* nel *dataset* finale, trattando ogni *frame* come osservazione indipendente dello spazio conformazionale del *nanobody*.

(ii) Suddivisione in finestre temporali: la serie temporale completa (80.000 *frame*) di ogni descrittore per ciascun *nanobody* è suddivisa in $M = 40$ finestre temporali di uguale dimensione. Ciascuna finestra risulta così composta da $80.000/M = 2.000$ *frame*, garantendo una copertura uniforme dell'intera simulazione.

(iii) Campionamento intra-finestra: da ciascuna finestra temporale vengono campionati $N = 200$ *frame* mediante selezione di indici relativi equispaziati. Questo approccio di campionamento periodico assicura una rappresentazione omogenea e sufficientemente esaustiva della dinamica all'interno di ogni

finestra temporale.

L'applicazione di questa strategia genera un *dataset* ampliato attraverso due meccanismi complementari: l'espansione delle *features* dinamiche (quelle relative ai descrittori di *time-series*, Sezione 4.1.3) e l'aumento del numero di campioni, intesi come repliche (*snapshot*) di un *nanobody* in una conformazione temporale differente.

Nello specifico:

(i) Espansione delle *features* dinamiche: le $M = 40$ finestre temporali, combinate con i 3 descrittori di *time series* (Alpha-RMSD, Anti-Beta-RMSD, Distanza CDR3-COM), generano un *set* di $3M = 120$ nuove *features* dinamiche strutturate come segue:
   - `alpha_section_1`, `alpha_section_2`, ..., `alpha_section_40`
   - `beta_section_1`, `beta_section_2`, ..., `beta_section_40`
   - `distance_section_1`, `distance_section_2`, ..., `distance_section_40`

(ii) Aumento dei campioni: il campionamento equispaziato di 200 *frame* (indici relativi) per ciascuna delle 40 finestre genera 200 *snapshot* unici per ogni *nanobody*. Ciascuno *snapshot* rappresenta una replica del *nanobody* campionata in un istante temporale specifico all'interno della serie temporale, e quindi con una conformazione strutturale differente.

Più in generale, denotando con $\mathscr{N}$ il numero di *nanobodies* unici selezionati, la struttura complessiva del *dataset augmentato* risulta quindi:

$$\text{Numero totale campioni} = \mathscr{N} \text{ } \textit{nanobodies} \times 200 \text{ } \textit{snapshot} = 200\mathscr{N} \text{ campioni}$$

Ciascun campione nel *dataset* finale integra due componenti distinte:

(i) *Features* invarianti: rimangono statiche tra tutti i 200 *snapshot* dello stesso *nanobody*, e includono le medesime *features* amminoacidiche (in quanto i residui la costituenti la proteina rimangono i medesimi), così come le stesse *features* di dinamica molecolare per residuo (4.3.1). Questo blocco di *features*, ammonta a 157.

(ii) Features dinamiche (o variabili): 120 differenti *features* per ogni *snapshot*, si differenziano in funzione della finestra temporale e del *frame* campionato.

La scelta dei parametri adottati $M = 40$ e $N = 200$, determinati empiricamente, rappresenta un compromesso ottimale per effettuare *data-augmentation*, e al contempo garantire una riduzione della dimensionalità del problema. In primo luogo, riduce significativamente la dimensionalità dello spazio dei campioni, passando da 80.000×3 possibili *features* dinamiche (corrispondenti a tutte le pose conformazionali della proteina) a 120, e al contempo aumenta considerevolmente il numero di campioni rispetto a un approccio che considera ciascuna biomolecola come singola osservazione. Questo bilanciamento permette di ottenere un *dataset* con una dimensionalità gestibile dai modelli, preservando al contempo, grazie al campionamento per indici relativi equispaziati all'interno di ogni finestra, gran parte dell'informazione della MD.

La Figura 4.8 illustra schematicamente l'intero processo di *data augmentation*.

4.3.2 Selezione delle *Features*

La combinazione tra il blocco di *features* invarianti e il blocco dinamico determina, per ciascuno *snapshot*, una dimensionalità elevata rispetto al numero modesto di campioni effettivamente indipendenti (a livello di *nanobody* ID unici). La dimensionalità dello spazio delle *features* ammonta a

$$p_{\text{tot}} \;=\; p_{\text{inv}} + p_{\text{dyn}} \;=\; 157 + 120 \;=\; 277, \tag{4.17}$$

potenzialmente soggetto alla *curse of dimensionality* [69]. In pratica, anche se l'*augmentation* temporale

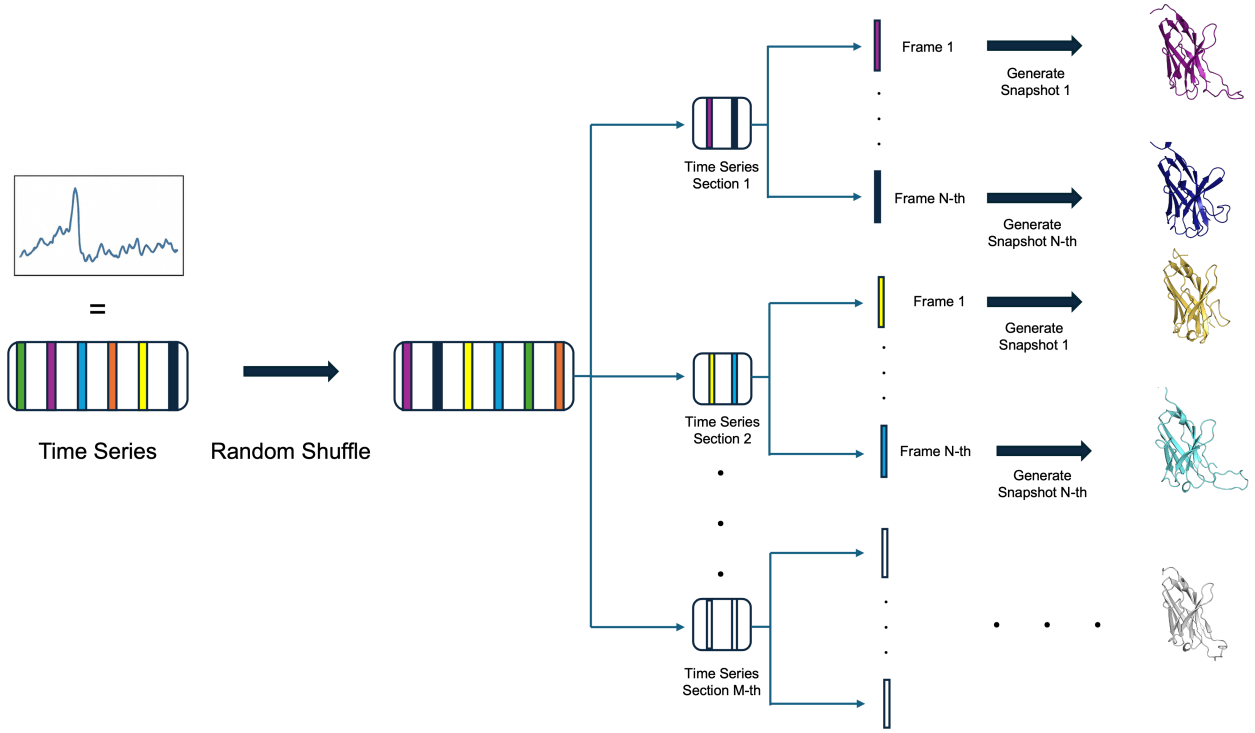

Figura 4.8: Schema della strategia di *data augmentation*. In Figura è riportato il *nanobody 7tpr 8a2 (ortosterico)*. Lo schema riporta l'intera *time-series* di un generico descrittore *frame-by-frame*, schematizzata attraverso delle barre colorate per evidenziare il processo di *shuffle*. Ogni *frame* campionato, indicato con uno specifico colore, è proiettato anche sulla struttura 3D della proteina, rappresenta una conformazione spaziale unica del *nanobody*.

aumenta il numero di osservazioni, non incrementa il numero di sorgenti biologiche indipendenti: di qui la necessità di una fase di selezione che riduca le ridondanze, e concentri l'informazione sul sotto-spazio più rilevante. Per massimizzare il contributo informativo complessivo, è stata adottata una strategia di selezione *greedy* applicata *esclusivamente* alle *features* invarianti.

La metodologia di selezione è stata eseguita utilizzando la libreria `SciPy` (v1.16.1) [70], e si è articolata in due fasi distinte:

(i) Filtro di varianza: sono state rimosse le *features* con varianza campionaria inferiore a 0.01, poco informative e che aumentano la dimensionalità senza dare contributo informativo apprezzabile [71].

(ii) Filtro di correlazione di Pearson: sulle variabili residue è stata calcolata la matrice di correlazione di Pearson in valore assoluto $|r|$; limitandosi al triangolo superiore (per evitare duplicazioni), tutte le *features* che presentavano almeno un coefficiente $|r| > 0.95$ sono state considerate ridondanti e rimosse secondo una regola *greedy*, per cui si mantiene la prima variabile nell'ordinamento delle colonne del *dataset* e si eliminano le successive ad essa altamente correlate [72].

Al termine del processo, le *features* invarianti passano da 157 a 47. In questo modo si ottiene un *dataset* che massimizza la capacità informativa sulla parte invariante, e preserva integralmente il contributo delle serie temporali.

4.3.3 Problema di classificazione

Il problema di classificazione affrontato in questo studio consiste nella predizione di classi binarie , corrispondenti rispettivamente alle categorie ortosterico (0) e allosterico (1). Formalmente, ogni campione del *dataset* è rappresentato come una coppia $(\mathbf{x}_i, y_i) \in X \times Y$, dove $X \subseteq \mathbb{R}^{167}$ è lo spazio delle *features* e $Y = \{0, 1\}$ l'insieme delle classi *target*. Ogni campione è quindi descritto da un vettore di *features* $\mathbf{x}_i \in \mathbb{R}^{167}$ e

un'etichetta di classe $y_i \in Y$.
Un modello di classificazione $f : X \to Y$ viene addestrato per associare ad ogni input $\mathbf{x} \in X$ una probabilità stimata $P(y_i = 1 \mid \mathbf{x})$ che, nel presente lavoro, rappresenta la confidenza del modello nell'appartenenza del campione alla classe allosterica (1). La predizione binaria finale $\hat{y}_i$ è determinata applicando una soglia di decisione alla probabilità stimata:

$$\hat{y}_i = \begin{cases} 1 \text{ (allosterico)} & \text{se } P(y_i = 1 \mid \mathbf{x}) \geq 0.5 \\ 0 \text{ (ortosterico)} & \text{se } P(y_i = 1 \mid \mathbf{x}) < 0.5 \end{cases} \tag{4.18}$$

4.3.4 Strategia di validazione

Per valutare la robustezza dei modelli di classificazione sviluppati, è stata adottata una strategia di validazione progressiva basata su due distinti *dataset* costruiti a partire dal *pool* completo di 50 *nanobodies* selezionati (Tabella 3.1):

(i) *Dataset Base*: composto dai primi 40 *nanobodies* della tabella, equamente ripartiti in 20 *nanobodies* ortosterici e 20 allosterici, impiegato per lo sviluppo iniziale e l'ottimizzazione dei modelli;
(ii) *Dataset Extended*: comprendente l'intero *set* di 50 *nanobodies*, introducendo una variabilità aggiuntiva del 25% rispetto al *dataset* base, con 10 nuovi *nanobodies*, mantenendo sempre il bilanciamento delle due classi.

La validazione è stata condotta in due fasi: inizialmente l'intera *pipeline* analitica è stata sviluppata e testata sul *dataset base*. Successivamente, la stessa metodologia è stata applicata al *dataset extended* per verificare se i risultati si mantenessero consistenti nonostante l'aumentata variabilità del *dataset* e il conseguente aumento di proteine uniche. Questo approccio consente di valutare la capacità di generalizzazione dei modelli e la stabilità delle prestazioni al crescere della complessità e variabilità dei dati.

4.3.5 Standardizzazione dei *dataset*

Tutte le *features* del *dataset* sono state preventivamente standardizzate utilizzando la classe STANDARDSCALER della librearia `scikit-learn` 1.7.1 [73], per garantire una distribuzione normale con media zero e deviazione standard unitaria, condizione essenziale per la corretta convergenza di molti algoritmi di *machine learning* e per la comparabilità dei coefficienti nei modelli lineari.

4.3.6 *Group-Stratified 5-Fold Cross-Validation* (GS5FCV)

La valutazione delle *performance* predittive è stata condotta attraverso una strategia di *cross-validation* applicata sia al *dataset* base ($\mathcal{N} = 40$), sia al *dataset extended* ($\mathcal{N} = 50$). In particolare, la strategia implementata segue il paradigma della *Group-Stratified 5-Fold Cross-Validation* (GS5FCV) [74]. Di seguito è riportata una spiegazione della strategia adottata in termini delle sue componenti:

(i) Il criterio di *grouping* per *Nanobody ID* assicura che tutti i $N = 200$ *snapshot* strutturalmente correlati di ciascun *nanobody* rimangano nello stesso *fold*, prevenendo ogni forma di *data leakage*, dato che il blocco di *features* invarianti è comune a tutte le repliche di ogni *nanobody*. Questo è essenziale per valutare la capacità di generalizzazione su strutture proteiche completamente nuove, piuttosto che su conformazioni già osservate durante l'addestramento.
(ii) La componente di stratificazione, in termini di mantenimento del bilanciamento fra classe ortosterica e allosterica, gioca un ruolo altrettanto fondamentale nel mantenere la proporzione originale delle classi in ciascun *fold*. In tale contesto di classificazione binaria, questa strategia impedisce che partizioni casuali risultino sbilanciate, garantendo che ogni *fold* di training contenga un numero sufficiente di esempi per entrambe le classi.
(iii) Lo schema a 5 *fold* organizza i $\mathcal{N}$ *nanobodies* unici in partizioni disgiunte, con ogni iterazione che

utilizza $\mathscr{N} \times \frac{5-1}{5}$ *nanobodies* ($200 \times \mathscr{N} \times \frac{5-1}{5}$ *snapshot* totali) per il *training* e $\mathscr{N} \times \frac{1}{5}$ *nanobodies* ($200 \times \mathscr{N} \times \frac{1}{5}$ *snapshot*) per il *test*.

La metodologia assicura che ogni *nanobody* compaia esattamente una volta nel *test set*.

Grazie alla CV (*Cross-Validation*), le metriche di *performance* vengono aggregate come media ± deviazione *standard* sui 5 *fold*, fornendo una stima robusta e statisticamente affidabile della capacità dei modelli di generalizzare su diverse distribuzioni di dati.

In pratica, questo approccio è stato selezionato per bilanciare in modo rigoroso due esigenze di eguale criticità: *in primis* garantire l'indipendenza tra *training* e *test set*, e successivamente preservare la rappresentatività della distribuzione bilanciata delle classi in tutte le partizioni.

Formalmente, sia $\mathscr{N} = \{n_1, n_2, \ldots, n_{\mathscr{N}}\}$ l'insieme degli identificativi ID univoci dei *nanobodies*. La partizione in *fold* viene effettuata rispettando la condizione:

$$\forall n \in \mathscr{N}, \forall k \in \{1, \ldots, 5\} : \text{Tutti gli } \textit{snapshot} \text{ di } n \text{ appartengono allo stesso } \textit{fold}.$$

Ad ogni iterazione k, il *training set* $D_{train}^{(k)}$ e *test set* $D_{test}^{(k)}$ sono definiti come:

$$D_{train}^{(k)} = \{(\mathbf{x}_i, y_i) \mid n_i \in \mathscr{N}_{train}^{(k)}\}, \quad D_{test}^{(k)} = \{(\mathbf{x}_i, y_i) \mid n_i \in \mathscr{N}_{test}^{(k)}\},$$

dove $\mathscr{N}_{train}^{(k)} \subset \mathscr{N}$ contiene $\mathscr{N} \times \frac{5-1}{5}$ *nanobodies* e $\mathscr{N}_{test}^{(k)} \subset \mathscr{N}$ contiene $\mathscr{N} \times \frac{1}{5}$ *nanobodies*, con $\mathscr{N}_{train}^{(k)} \cap \mathscr{N}_{test}^{(k)} = \emptyset$ e $\bigcup_{k=1}^{5} \mathscr{N}_{test}^{(k)} = \mathscr{N}$.

Per il *dataset* base ($\mathscr{N} = 40$):

- *Training*: $40 \times \frac{4}{5} = 32$ *nanobodies* ($200 \times 32 = 6.400$ *snapshot*)
- *Test*: $40 \times \frac{1}{5} = 8$ *nanobodies* ($200 \times 8 = 1.600$ *snapshot*)

Per il *dataset extended* ($\mathscr{N} = 50$):

- *Training*: $50 \times \frac{4}{5} = 40$ *nanobodies* ($200 \times 40 = 8.000$ *snapshot*)
- *Test*: $50 \times \frac{1}{5} = 10$ *nanobodies* ($200 \times 10 = 2.000$ *snapshot*)

Questa strategia è stata applicata separatamente a entrambi i *dataset* (base ed *extended*), consentendo un confronto diretto delle *performance* predittive, e sulla stabilità delle stesse.

4.3.7 Analisi Esplorativa della Separabilità delle Classi tramite UMAP (*Uniform Manifold Approximation and Projection*)

Prima di procedere con l'addestramento dei classificatori, è stata condotta un'analisi esplorativa della distribuzione dei campioni nel *dataset* utilizzando UMAP (*Uniform Manifold Approximation and Projection*), implementata nella libreria `umap-learn v0.5.9` [75]. UMAP è una tecnica di riduzione dimensionale non lineare che permette di proiettare dati ad alta dimensionalità in uno spazio *embedded* a dimensionalità ridotta. Questa proiezione preserva la struttura topologica dello spazio originale, mappando prossimità e distanze tra punti nello spazio *embedded*.

In questo lavoro è stato scelto uno spazio bidimensionale per facilitare la visualizzazione.

In linea con il *framework* proposto da Cucchi e Frasnetti et al. [76], la proiezione è stata ottenuta configurando UMAP con i seguenti parametri: *n_neighbors=20*, metrica di similarità di Jaccard (adeguata per la natura binaria dei dati di classificazione) e 1000 epoche per la fase di ottimizzazione.

L'impiego principale di UMAP in questo contesto è stato puramente diagnostico, finalizzato a valutare visivamente la partizione dei dati adottata nella procedura di *cross-validation*. Per ciascun *fold*, è stata generata una visualizzazione distinta mostrante la distribuzione dei *nanobodies* nello spazio bidimensionale, suddivisi tra *training set* e *test set*.

L'obiettivo dell'analisi è duplice e concettualmente distinguibile in due approcci:

(i) Analisi *Inter-set* (*Training vs. Test*): valutare, per ogni iterazione di *cross-validation*, la coerenza della distribuzione dei campioni di *test* quando questi vengono proiettati in uno spazio UMAP appreso esclusivamente sul *training set*. In particolare, si cerca di identificare campioni di *test* che ricadono in regioni dello spazio dominate da una singola classe del *training set*. Questa condizione suggerisce una potenziale facilità di classificazione per tali istanze; in alternativa, si può verificare il caso in cui campioni di *test* che, al contrario, si posizionano in regioni di sovrapposizione tra classi diverse del *training set* o in prossimità dei confini decisionali, prefigurando possibili incertezze per il classificatore.

(ii) Analisi *Intra-set* (nel *Test Set*): esaminare la distribuzione interna al solo *test set* per identificare la presenza di sovrapposizioni spaziali (*overlap*) tra campioni di classi diverse. La presenza di tali sovrapposizioni, a livello di proiezione 2D, fornisce un'indicazione preliminare sulla complessità della *task* di classificazione e sulle *performance* attese del modello, segnalando regioni dello spazio delle *features* dove la separabilità delle classi è critica.

4.3.8 Selezione dei Modelli di *Machine Learning* e *Deep Learning* e Costruzione delle Architetture

L'impianto sperimentale adottato integra modelli di *machine learning* della librearia *scikit-learn* e architetture di *deep learning*, implementati tramite *PyTorch* (`torch` v2.4.1, `torchvision` v0.19.1, `torchaudio` v2.4.1) [77], adattati per dati tabellari.

L'uso di *PyTorch* è motivato dalla flessibilità nella regolazione degli iperparametri e nella definizione delle architetture.

La selezione di un campionario algoritmico esteso ed eterogeneo, comprendente tredici fra modelli di *machine learning* e *deep learning*, risponde a un duplice obiettivo metodologico.

In primo luogo, l'eterogeneità dei modelli implementati garantisce l'ortogonalità degli approcci, permettendo di studiare il problema di classificazione attraverso approcci algoritmici complementari. Inoltre, l'insieme include modelli di complessità crescente, che consente di analizzare il *trade-off* tra complessità e guadagno prestazionale per identificare la tipologia architetturale più adatta a gestire il *dataset* in esame.

In secondo luogo, la motivazione principale che giustifica l'impiego di un numero elevato di modelli risiede nella successiva fase di *features importance analysis*, dove è richiesta una robustezza statistica significativa. L'analisi dell'importanza delle features basata su un singolo modello, o su pochi modelli, sarebbe fortemente esposta al rischio di risultati fortuiti o *bias* algoritmico-specifici. Al contrario, l'impiego simultaneo di tredici architetture permette di mediare l'effetto *cross-modello* e quantificare la variabilità. Questo approccio, consente di distinguere le *features* informative da quelle che potrebbero apparire rilevanti solo in specifici modelli, fornendo così una stima più stabile e affidabile dell'importanza delle stesse.

Di seguito è riprotata la lista dei modelli impiegati, divisi in funzione della libreria che ha permesso la loro implementazione.

La lista completa comprendente anche i parametri adottati è riportata nella Appendice del lavoro.

Modelli di *Deep Learning* (PyTorch)

- *Neural Network*: *Multi-Layer Perceptron* (MLP) costituito da strati *fully-connected* [78]. Sono state implementate tre diverse architetture, che differiscono per profondità (numero di *layer*) e ampiezza (numero di neuroni per ogni *layer*):
  - MLP *Light*
  - MLP *Wide*
  - MLP *Deep*

  Per tutti i modelli neurali sono stati applicati queste configurazioni: *BatchNormalization* [79] e *Dropout* [80]. La *Batch Normalization* (BN) stabilizza l'addestramento normalizzando le attivazioni di ciascuno strato, riducendo il cosiddetto *internal covariate shift* e favorendo una convergenza più rapida. Il

Dropout (DO) contrasta l'*overfitting* azzerando casualmente una frazione dei neuroni della rete durante il *training*: così il modello impara rappresentazioni ridondanti e più robuste, che generalizzano meglio su dati non visti. Insieme, BN e DO aumentano la stabilità del modello, accelerano l'apprendimento e riducono l'*overfitting*, soprattutto in presenza di dati di *input* complessi ed eterogenei.

- *Deep ResNet*: basata sui principi di ResNet [81] ma specificamente adattata per l'elaborazione di dati tabellari. L'innovazione rispetto ad una *Neural Network standard* risiede nell'implementazione di *blocchi residui* che utilizzano *skip connections* per migliorare le proprietà numeriche quali, ad esempio, il flusso del gradiente attraverso gli strati interconnessi.
- *Attention Network*: Architettura neurale con meccanismo di *self-attention features-wise* [82] che apprende pesi di rilevanza specifici per ogni campione e *feature*. Composta da due componenti principali: un modulo di *attention* e un *classifier* MLP.

Modelli di *Machine Learning* (scikit-learn)

- *Logistic Regression* [83];
- *K-Nearest Neighbors* [84];
- *Random Forest* [85];
- *Gradient Boosting* [86];
- *XGBoost* (implementato grazie alla sua apposita libreria `xgboost` v3.0.4) [87];
- *SVM* [88];
- *Deep Forest*: Approccio *deep learning* non neurale con *cascade forest* [89];
- *Voting Classifier*: Combinazione di modelli, secondo una strategia *wisdom of crowds* [90].

4.3.9 Metriche di Valutazione

La valutazione delle performance dei modelli è stata condotta attraverso l'analisi della matrice di confusione binaria [91].

Per semplicità, e per coerenza con la nomenclatura tradizionale, si considerano come positivi i *nanobodies* allosterici (1), mentre negativi quelli ortosterici (0). La matrice di confusione organizza i risultati predittivi in quattro categorie distinte:

- *True Positive* (TP): campioni positivi correttamente classificati;
- *True Negative* (TN): campioni negativi correttamente classificati;
- *False Positive* (FP): campioni negativi erroneamente classificati come positivi (errore di Tipo I);
- *False Negative* (FN): campioni positivi erroneamente classificati come negativi (errore di Tipo II).

Il calcolo della matrice di confusione è espresso in percentuale utilizzando la funzione `confusion_matrix` della libreria `sklearn.metrics` [73]. La normalizzazione è stata applicata per riga, in modo che ciascun valore rappresenti la percentuale di campioni di una specifica classe reale classificata in ciascuna categoria predetta. A partire dalla matrice di confusione sono state calcolate le seguenti metriche di classificazione *standard* [92]:

(i) Accuracy (ACC): misura la proporzione complessiva di predizioni corrette rispetto al totale delle istanze

$$\mathrm{ACC} = \frac{TP+TN}{TP+TN+FP+FN}. \tag{4.19}$$

(ii) Sensitivity (*Recall/True Positive Rate, TPR*): quantifica la capacità del modello di identificare correttamente i positivi (allosterici)

$$\mathrm{TPR} = \frac{TP}{TP+FN}. \tag{4.20}$$

(iii) Specificity (*True Negative Rate, TNR*): valuta l'abilità del modello di riconoscere correttamente i negativi

(ortosterici)

$$\text{TNR} = \frac{TN}{TN+FP}. \tag{4.21}$$

(iv) Precision (*Positive Predictive Value, PPV*): misura l'affidabilità delle predizioni positive (allosterici)

$$\text{PPV} = \frac{TP}{TP+FP}. \tag{4.22}$$

(v) F1-score: metrica definita come media armonica tra *precision* (PPV) e *recall* (TPR)

$$\text{F1} = 2\frac{\text{PPV}\cdot\text{TPR}}{\text{PPV}+\text{TPR}} \tag{4.23}$$

(vi) Matthew's Correlation Coefficient (MCC): misura completa che considera tutti gli elementi della matrice di confusione

$$\text{MCC} = \frac{TP\times TN - FP\times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}. \tag{4.24}$$

Le metriche adottate sono tutte normalizzate nell'intervallo [0,1] ad eccezione del MCC, che spazia nell'intervallo [-1,1].

Nel presente lavoro, ai fini di garantire una maggiore interpretabilità dei risultati, tutte le metriche ad eccezione del MCC (4.24) saranno espresse in percentuale.

4.3.10 Analisi dell'Importanza delle *Features*

Come precedentemente discusso nel Capitolo 2, l'analisi dell'importanza delle *features* costituisce una componente fondamentale di questo lavoro, ponendosi come tramite metodologico tra le predizione dei modelli di apprendimento supervisionato e l'indagine dei determinanti chimico-fisici sui meccanismi molecolari di classificazione dei *nanobodies*.

L'obiettivo primario di questa analisi è quantificare il contributo discriminante di ciascuna delle 167 *features* (ripartite in 47 invarianti e 120 dinamiche, come descritto nella Sezione 4.3.2) per tutti i modelli f_i con $i \in [1,...,13]$, generando per ciascuno un vettore di importanza $\mathbf{I} \in \mathbb{R}^{167}$. Questo approccio, cerca di identificare le *features* con un fondamento chimico-fisico rilevante nel processo decisionale dei modelli, fornendo così un riscontro interpretabile alle loro *performance* predittive; al tempo stesso, validare la coerenza biologica delle predizioni, verificando se le *features* identificate come importanti siano effettivamente riconducibili a proprietà molecolari peculiari nelle due classi.

Importanza delle *features* per modelli neurali

Per le architetture basate su MLP *classifier*, è stato sviluppato un approccio ibrido che combina due contributi complementari, entrambi valutati *a posteriori* rispetto alla fase di addestramento.

Si consideri un MLP con primo *layer fully-connected*, definito dalla trasformazione lineare:

$$\mathbf{z}^{(1)} = W^{(1)}\mathbf{x} + \mathbf{b}^{(1)} \tag{4.25}$$

dove $\mathbf{x} \in \mathbb{R}^d$ è il vettore di *input* delle *features* di un singolo campione, con $d = 167$; $W^{(1)} \in \mathbb{R}^{h\times d}$ è la matrice dei pesi del primo *layer* nascosto, e h rappresenta la dimensione dell'*hidden layer*, ovvero, il numero di neuroni che compongono lo strato della rete; $\mathbf{b}^{(1)} \in \mathbb{R}^h$ è il vettore dei *bias*; $\mathbf{z}^{(1)} \in \mathbb{R}^h$ è il vettore dei *logits*.

Il primo contributo all'analisi dell'importanza è fornito dall'esame della matrice dei pesi $W^{(1)}$ dopo il *training* della rete, secondo la relazione:

$$I_j^w = \frac{1}{h}\sum_{k=1}^{h} |W_{k,j}^{(1)}| \tag{4.26}$$

Questa metrica quantifica l'importanza della *feature* j-esima basata sulla configurazione finale dei parametri. Ciascun elemento $W_{k,j}^{(1)}$ rappresenta l'intensità con cui la *feature* j influenza il valore di pre-attivazione del neurone nascosto k nella rete addestrata. La media su tutti gli h neuroni fornisce una misura aggregata dell'impatto complessivo della *feature* sulla trasformazione della rappresentazione al primo *layer* [93].

Complementarmente, l'approccio basato sui gradienti valuta la sensibilità della funzione di *loss* $\mathscr{L}$ rispetto alle *features* di *input*, utilizzando il modello già addestrato durante l'inferenza sul *set* di *test*. Dato un *batch* B di $m = |B|$ campioni $X_B \in \mathbb{R}^{m \times d}$, l'importanza basata sui gradienti è definita come:

$$I_j^{\nabla} = \frac{1}{|B|} \sum_{i \in B} \left| \frac{\partial \mathscr{L}_i}{\partial x_{ij}} \right|. \tag{4.27}$$

Le derivate parziali $\partial \mathscr{L}_i / \partial x_{ij}$ vengono calcolate automaticamente tramite *backpropagation*, applicando la regola della catena attraverso tutta l'architettura della rete [94]. Questo approccio identifica le regioni dello spazio delle *features* dove il modello addestrato mostra alta sensibilità decisionale (*saliency*), indicando che piccole variazioni della *feature* producono significativi cambiamenti nell'*output* predittivo.

I due contributi complementari sono stati integrati attraverso un processo di normalizzazione e media per cui:

$$\tilde{I}_j^{w} = \frac{I_j^{w}}{\sum_{\ell=1}^{d} I_\ell^{w}}, \quad \tilde{I}_j^{\nabla} = \frac{I_j^{\nabla}}{\sum_{\ell=1}^{d} I_\ell^{\nabla}}, \quad I_j^{\text{MLP}} = \frac{\tilde{I}_j^{w} + \tilde{I}_j^{\nabla}}{2} \tag{4.28}$$

Importanza delle *features* per modelli non parametrici

La *Permutation Importance* è stata applicata ai modelli che non dispongono di una misura intrinseca di importanza delle *features*, in particolare *Support Vector Machines* (SVM) e *K-Nearest Neighbors* (KNN). Questa tecnica valuta l'importanza di ciascuna *feature* misurando la degradazione delle *performance* del modello quando la relazione statistica tra la *feature* e la classe viene alterata attraverso permutazione casuale [95].

Sia $D_{\text{test}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_{\text{test}}}$ il *test set*, e sia $M(f, D_{\text{test}})$ una metrica di *performance* del modello f (in questo studio è stata scelta l'*accuracy*). L'importanza per permutazione della *feature* j-esima è definita come:

$$\mathrm{I}_j^{P} = \frac{1}{R} \sum_{r=1}^{R} \left[M(f, D_{\text{test}}) - M(f, D_{\text{test}}^{(j,r)}) \right] \tag{4.29}$$

dove $D_{\text{test}}^{(j,r)}$ indica il *dataset* di *test* in cui i valori della *feature* j sono stati permutati casualmente alla r-esima iterazione. Formalmente:

$$\mathbf{x}_i^{(j,r)}[j] = \pi^{(r)}(\mathbf{x}_i[j]) \quad \forall i \in \{1, \ldots, n_{\text{test}}\} \tag{4.30}$$

La permutazione $\pi^{(r)}$ viene applicata campionando senza ripetizione i valori della *feature* j tra tutti gli n_{test} campioni del *test set*, preservando così la distribuzione marginale della *feature* ma alterando la sua relazione con la variabile *target* (ovvero la classe di appartenenza del *nanobody*).

Nel presente studio, il numero di ripetizioni R è stato fissato a 3, seguendo un compromesso tra robustezza statistica e costo computazionale. Un valore positivo di I_j^{P} indica che la *feature* j contribuisce positivamente alle performance del modello, mentre valori prossimi allo zero suggeriscono che la *feature* è irrilevante o ridondante.

Importanza delle *features* per modelli ad albero

Per i modelli *tree-based* (*Random Forest, Gradient Boosting, XGBoost e Deep Forest*) è stata utilizzata la misura di importanza basata sulla riduzione dell'impurità di Gini, implementata direttamente nelle librerie `scikit-learn` e `XGBoost` seguendo il *framework* di Breiman et al. [96].

Questo approccio quantifica l'importanza di ciascuna *feature* attraverso la riduzione media di impurità ottenuta in tutti gli *split* dell'*ensemble* che utilizzano quella *feature* specifica. Formalmente, l'importanza della *feature* j-esima è definita come:

$$I_j^{\text{tree}} = \frac{1}{T} \sum_{t=1}^{T} \sum_{s \in S_j} \Delta\text{Gini}(s), \tag{4.31}$$

dove: T è il numero totale di alberi nell'*ensemble*, S_j rappresenta l'insieme di tutti gli *split* che utilizzano la *feature* j, $\Delta\text{Gini}(s)$ è la riduzione di impurità di Gini ottenuta dallo *split* s, calcolata come:

$$\Delta\text{Gini}(s) = \text{Gini}_{\text{parent}} - \left(\frac{N_{\text{left}}}{N_{\text{parent}}} \text{Gini}_{\text{left}} + \frac{N_{\text{right}}}{N_{\text{parent}}} \text{Gini}_{\text{right}} \right). \tag{4.32}$$

Nella (4.32), $\text{Gini}_{\text{parent}}$ misura l'impurità del nodo genitore prima dello *split*, N_{parent} è il numero totale di campioni nel nodo genitore, N_{left} e N_{right} sono, rispettivamente, il numero di campioni nei nodi figlio sinistro e destro, mentre $\text{Gini}_{\text{left}}$ e $\text{Gini}_{\text{right}}$ misurano l'impurità dei nodi figlio sinistro e destro, rispettivamente.
L'impurità di Gini per un nodo è definita come $\text{Gini} = 1 - \sum_{c=1}^{C} p_c^2$, dove p_c è la proporzione dei campioni della classe c nel nodo e C è il numero di classi (nel caso binario, $C = 2$).

Modelli Lineari: Coefficienti Standardizzati

Per la regressione logistica, con *features* standardizzate tramite STANDARDSCALER, l'importanza è assunta proporzionale ai coefficienti:

$$I_j^{\text{LR}} = |\beta_j|. \tag{4.33}$$

I coefficienti standardizzati della regressione logistica sono un indicatore di importanza relativa tra variabili, a parità di scala [97].

Normalizzazione e Aggregazione *Cross*-Modello

La natura eterogenea dei tredici algoritmi considerati in questo studio ($|M| = 13$ modelli), nonchè dei diversi approcci di calcolo della *features importance*, genera punteggi di importanza su scale e distribuzioni differenti. Per ottenere una misura comparabile di importanza tra modelli diversi, è stato implementato un processo di normalizzazione e aggregazione.
Inizialmente, per ciascun modello $m \in M$, le importanze ottenute dai diversi *fold* di *cross-validation* vengono aggregate attraverso una media aritmetica:

$$\bar{I}_j^{(m)} = \frac{1}{5} \sum_{k=1}^{5} I_j^{(m,k)} \tag{4.34}$$

dove $I_j^{(m,k)}$ rappresenta l'importanza *raw* della *feature* j per il modello m nel fold k. Questo serve per generare una stima robusta e stabile dell'importanza per ciascun modello, attenuando la variabilità nelle partizioni dei dati.
Successivamente, per ciascun modello m, il vettore di importanza mediato $\bar{I}^{(m)} \in \mathbb{R}^{167}$ viene normalizzato nell'intervallo [0,1] tramite *min–max scaling*, cioè riportando tutte le componenti del vettore di quel modello sulla scala $[0,1]$ usando esclusivamente i suoi valori interni. In particolare, ponendo

$$\min_l \bar{I}_l^{(m)} = \min_{l \in \{1,\ldots,167\}} \bar{I}_l^{(m)} \quad \text{e} \quad \max_l \bar{I}_l^{(m)} = \max_{l \in \{1,\ldots,167\}} \bar{I}_l^{(m)},$$

la normalizzazione di ciascuna *feature* j è definita da:

$$\bar{I}^{(m)}_{j,\text{norm}} = \frac{\bar{I}^{(m)}_j - \min_l \bar{I}^{(m)}_l}{\max_l \bar{I}^{(m)}_l - \min_l \bar{I}^{(m)}_l}. \tag{4.35}$$

Se $\max_l \bar{I}^{(m)}_l = \min_l \bar{I}^{(m)}_l$ (tutte le *features* presentano identica importanza), per evitare divisioni per zero si assegna $\bar{I}^{(m)}_{\text{norm}} = \mathbf{0}$. Questa trasformazione preserva l'ordinamento relativo delle *features* all'interno di ciascun modello, rendendo i punteggi direttamente confrontabili.

Infine, una volta individuato il modello con le migliori prestazioni di classificazione, verrà presentata e discussa la relativa *feature importance*.

Il punteggio di consenso finale (*consensus score*) tra tutti i modelli è ottenuto come media delle importanze normalizzate:

$$I^{\text{cons}}_j = \frac{1}{|M|} \sum_{m=1}^{|M|} \bar{I}^{(m)}_{j,\text{norm}}, \tag{4.36}$$

dove $|M| = 13$ rappresenta il numero totale di modelli implementati. La scelta di un campionario algoritmico ampio ed eterogeneo, come motivato nella Sezione 4.3.3, risulta cruciale per ottenere una stima affidabile dell'importanza delle *features*, attenuando i *bias* specifici di ciascun approccio.

In questo modo, è possibile stilare un *ranking* finale dell'importanza delle *features*, nel quale viene considerata una sola metrica riassuntiva che tiene in considerazione la variabilità fra i *fold* (attraveso la media).

In altre parole, il *consensus score* così ottenuto quantifica l'accordo dei modelli sulla *feature* j e al contempo le ordina per scala di importanza *cross-modello*.

Per migliorare l'interpretabilità dei risultati, le *features* selezionate sono state ulteriormente raggruppate in base al loro significato fisico-chimico. Inoltre, è stata condotta un'analisi della distribuzione per regione, distinguendo le *features* appartenenti alle regioni CDR3 e al *framework*, allo scopo di identificare *pattern* specifici associati a ciascuna regione del *nanobody*.

4.3.11 *Test* Statistici sulle *Features*

L'identificazione delle *features* più discriminanti costituisce un risultato in parte insufficiente, che richiede una validazione statistica per distinguere le differenze tra classi da eventuali artefatti predittivi. Infatti, la necessità di questa validazione nasce dalla consapevolezza che il *ranking* fornito dall'analisi di *feature importance* potrebbe essere fortuitamente influenzato da *bias* algoritmici o da preferenze di alcuni modelli che, sebbene discriminanti, potrebbero non riflettere differenze di natura fisico-chimiche fondamentali [98]. A supporto di questa esigenza, la validazione è stata condotta sulle *top*-10 *features* emerse dal *consensus score*, così da confrontare la separazione tra classi con misure inferenziali.

Le librerie `Python` che sono state impiegate in questa fase sono STATSMODELS (v0.14.5) e PATSY (v1.0.1) [99, 100].

Scelta del *test* statistico

In prima battuta, è stata implementata una strategia gerarchica per la selezione del *test* statistico più appropriato. Il processo inizia con la valutazione delle assunzioni distributive delle *features* del *ranking* all'interno di ciascuna classe. Il test di Shapiro–Wilk [101] verifica l'ipotesi di normalità, mentre il test di Brown–Forsythe [102] valuta l'omoschedasticità tra le classi. La condizione di normalità è considerata soddisfatta quando almeno una delle classi supera il *test* di Shapiro-Wilk con $p > 0.05$.

La scelta del *test* statistico finale avviene attraverso un algoritmo decisionale automatizzato che privilegia gli approcci parametrici quando le assunzioni distributive sono soddisfatte, ricorrendo ai *test* non-parametrici in

caso contrario.

In forma compatta, la regola decisionale può essere espressa come:

$$Test(features) = \begin{cases} \text{t-test (a due code)[103]} & \text{se } (p_{SW,0} > 0.05 \vee p_{SW,1} > 0.05) \wedge (p_{BF} > 0.05), \\ \text{Mann–Whitney U Test (a due code)[104]} & \text{altrimenti.} \end{cases}$$

dove i simboli "$\vee$" e "$\wedge$" indicano rispettivamente l'operatore logico OR e AND, $p_{SW,0}$ e $p_{SW,1}$ sono i *p-value* del *test* di Shapiro–Wilk calcolati rispettivamente nella classe *ortosterica* (0) e nella classe *allosterica* (1), mentre p_{BF} è il *p-value* del *test* di Brown–Forsythe sull'uguaglianza delle varianze tra le due classi.

La strategia di validazione con i *test* statistici adotta criteri fortemente conservativi per minimizzare il rischio di falsi positivi: la soglia di significatività per i *test*, una volta scelto quale adottare tramite codice automatizzato precedentemente delineato, è stata fissata a $\alpha = 0.005$.

Stima del valore rappresentativo per ogni classe

Per ciascuna *feature s* che supera il *test* statistico di significatività, il valore rappresentativo per ogni classe viene calcolato come valore atteso della corrispondente distribuzione stimata mediante *Kernel Density Estimation* (KDE) gaussiano [105]. Per una classe (ad esempio, indicando con $p_s(x)$ la classe ortosterica) la densità è:

$$p_s(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right), \qquad K(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}, \tag{4.37}$$

dove n è il numero di osservazioni della *feature* nella classe, x_i i valori osservati e h la larghezza di banda scelta automaticamente tramite la regola di Scott, $h = \sigma n^{-1/5}$, con σ deviazione standard campionaria [106]; in modo del tutto analogo si ottiene $q_s(x)$ per l'altra classe (allosterica).

Il valore rappresentativo per ciascuna classe è definito come:

$$\mu_s^{(\text{ort})} = \mathbb{E}_{p_s}[X] = \int x\, p_s(x)\, dx, \quad \mu_s^{(\text{all})} = \mathbb{E}_{q_s}[X] = \int x\, q_s(x)\, dx. \tag{4.38}$$

Per un *kernel* gaussiano, simmetrico e a media nulla, queste quantità coincidono esattamente con le medie aritmetiche dei valori osservati della *feature s* nelle rispettive classi. Pertanto, i valori rappresentativi corrispondono alle tradizionali medie campionarie, ma concettualmente derivati come momenti primi delle distribuzioni stimate.

I valori $\mu_s^{(\text{ort})}$ e $\mu_s^{(\text{all})}$, per ciascuna *feature s* significativa, sono stati indicati da linee verticali tratteggiate sovrapposte alle corrispondenti curve di densità.

Visualizzazione delle distribuzioni delle *features* significative

Per ciascuna *feature* che supera la validazione statistica (Sezione 5.1.4), le distribuzioni dei valori nelle due classi sono state rappresentate mediante istogrammi affiancati, per permettere un confronto diretto.

La costruzione degli istogrammi segue una procedura in tre fasi:

(i) Stima della densità: i conteggi per ciascun *bin* c_k vengono convertiti in una stima di densità di probabilità normalizzata:

$$\hat{f}_k = \frac{c_k}{n \cdot h}, \tag{4.39}$$

dove n rappresenta la numerosità campionaria della classe e h la larghezza del *bin*.

(ii) Scelta ottimale della larghezza dei *bin*: la larghezza h è determinata secondo la regola di Freedman–Diaconis [107]:

$$h_{FD} = 2\frac{Q_{0.75} - Q_{0.25}}{n^{1/3}}, \tag{4.40}$$

dove $Q_{0.75} - Q_{0.25}$ costituisce l'intervallo interquartile dei dati.

(iii) Normalizzazione per il confronto visuale: per enfatizzare la forma della distribuzione indipendentemente dalla scala assoluta, le densità vengono riscalate mediante normalizzazione L^∞ [108] secondo:

$$\tilde{f}_k = \frac{\hat{f}_k}{\max_j \hat{f}_j}. \tag{4.41}$$

Il risultato finale $\tilde{f}_k \in [0,1]$ rappresenta l'altezza di ciascun *bin* come frazione del valore massimo dell'istogramma. Questo approccio preserva integralmente la forma della distribuzione, mentre garantisce la comparabilità tra le due classi, essendo ciascuna istogramma normalizzato rispetto al proprio picco massimo.

Validazione di risultati sul *Dataset Extended*

La medesima metodologia è stata integralmente estesa anche al *dataset* comprendente 50 *nanobodies*. Una modifica metodologica significativa riguarda la soglia di significatività statistica, inasprita da $\alpha = 0.005$ a $\alpha = 0.001$.

Questa scelta è motivata dall'aumentata potenza statistica del campione più numeroso, che consente di adottare criteri più conservativi per la validazione delle differenze tra classi.

5 Risultati: predizione delle preferenze di *Binding* dei *Nanobodies* a partire dalla Dinamica Molecolare

Nella prima parte sono presentate le *performance* dei modelli di *machine learning* e *deep learning* sul *dataset* base, seguite da una *feature importance analysis*. A corredo di questa indagine, saranno proposti i risultati di *test* statistici dedicati a validare la significatività delle *features* selezionate dai modelli come maggiormente discriminanti.

La seconda parte illustra i risultati della validazione applicata all'*extended dataset*, che comprende 10 *nanobodies* aggiuntivi, in cui viene ripetuto il medesimo *workflow* adottato per il caso base. Nella stessa sezione è inoltre proposto un confronto delle prestazioni predittive con altri approcci computazionali, come specificato nella Sezione 1.4.

5.1 Risultati sul *Dataset* Base

In questa sezione vengono presentate le *performance* predittive dei tredici modelli di *machine learning* e *deep learning* implementati, addestrati e testati sul *dataset* base attraverso GS5FCV, secondo la strategia riportata in Sezione 4.3.

Per la trattazione concernente la parte di *feature engineering* si rimanda alla Sezione 4.3.1.

La rappresentazione matematica del *dataset* base risultante può essere formalizzata come una matrice $D \in \mathbb{R}^{n \times m}$, dove $n = 40\,nanobodies \times 200\,snapshot = 8000$ rappresenta il numero totale di campioni (comprendenti le repliche, o *snapshot*) e $m = 167$ il numero di *feature* selezionate, ripartite fra 47 invarianti e 120 dinamiche. La scelta circa la predominanza delle *feature* dinamiche (120 su 167) è stata determinata prevalentemente da considerazioni computazionali e di ottimizzazione delle *performance* dei modelli: da qui lo sbilanciamento verso le *feature* temporali, che risponde primariamente all'esigenza di massimizzare l'efficacia predittiva, e al contempo di mantenere un sotto-campionamento completo della traiettoria originale, necessaria per catturare adeguatamente l'evoluzione temporale dei descrittori conformazionali attraverso le finestre temporali (per una trattazione più estesa si rimanda alle Sezioni 4.3.1 e 4.3.2).

Le informazioni strutturali e di MD per residuo rimangono essenziali per la caratterizzazione dei *nanobodies*, ma la loro rappresentazione nel *dataset* è risultata sufficientemente informativa da un numero più contenuto di *feature* altamente informative dopo la *features selection* 4.3.2.

5.1.1 Risultati dell'analisi esplorativa tramite UMAP

Prima di approfondire le capacità predittive dei modelli, è fondamentale contestualizzare i risultati alla luce della distribuzione del *dataset*. L'analisi UMAP condotta in fase esplorativa (Sezione 4.3.7) rivela una struttura complessa caratterizzata da una duplice tipologia di sovrapposizione.

Come evidenziato in alcuni esempi della Figura 5.1 tramite i cerchi rosa, e indicati con le lettere maiuscole, si osserva un marcato *overlap* tra le classi nello spazio *embedded* del *test set*, con regioni dello spazio in cui istanze ortosteriche e allosteriche coesistono (*overlap intra-set*), senza una ben definita *clusterizzazione* fra le istanze.

In secondo luogo, e ancor più significativamente, emerge una sovrapposizione tra *training* e *test set* che coinvolge *nanobodies* appartenenti a classi differenti. Come mostrato dagli esempi evidenziati con i cerchi azzurri (in particolare sono indicati fra i due *set* tramite la coppia 1a–1b, e così via), si riscontrano regioni dello spazio UMAP in cui istanze di *training* di una data classe si trovano in prossimità di istanze di *test* della classe opposta (fenomeno di *overlap inter-set*). Questo fenomeno suggerisce l'esistenza di *nanobodies* che presentano proprietà strutturalmente simili, che tuttavia appartengono a meccanismi di legame differenti.

Questa doppio fenomeno di *overlap* fornisce una prima verifica di natura visiva della complessità del problema.

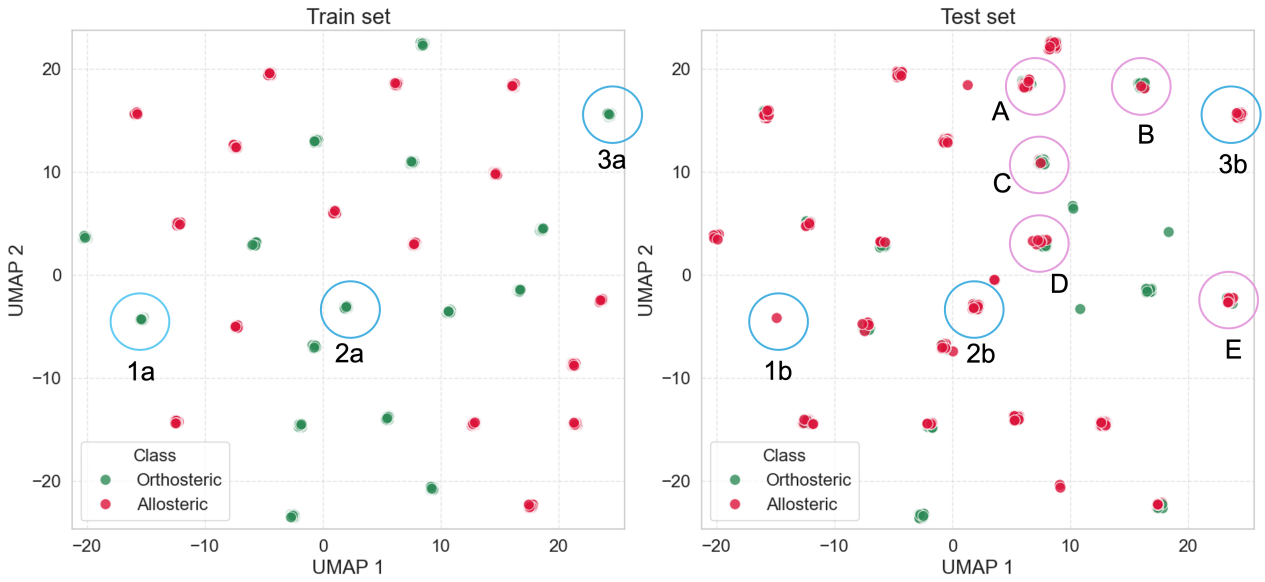

Figura 5.1: Proiezione UMAP del *dataset* base, in uno degli *split* della *cross-validation*.

5.1.2 Risultati predittivi

Tabella 5.1: *Performance* comparative dei *top-5* classificatori sul *dataset* base. I valori sono calcolati su GS5FCV e sono espressi come media ± deviazione *standard* (i valori sono espressi in percentuale, eccetto MCC).

Modello	**ACC (%)**	**TPR (%)**	**TNR (%)**	**F1 (%)**	**PPV (%)**	**MCC**
MLP - Light Architecture (PyTorch)	73.7±6.4	70.0±18.7	77.5±17.0	71.8±8.9	78.9±13.1	0.509±0.137
Attention Network (PyTorch)	70.0±10.0	75.0±15.8	65.0±12.2	70.9±10.6	68.3±8.0	0.410±0.205
MLP - Wide Architecture (PyTorch)	67.6±9.7	65.3±19.8	70.0±18.7	66.0±10.7	71.8±16.0	0.385±0.219
K-Nearest Neighbors (scikit-learn)	67.3±17.0	70.2±18.5	64.4±18.6	68.0±17.2	66.6±17.0	0.349±0.339
Deep ResNet (PyTorch)	67.0±8.2	65.1±19.8	68.9±21.5	65.5±10.3	69.7±10.8	0.372±0.163

In primo luogo, dall'analisi comparativa dei modelli di classificazione presentati in Tabella 5.1, si osserva come i modelli di *machine learning* implementati con SCIKIT-LEARN (*Random Forest, SVM, Gradient Boosting*, etc.) non compaiano tra i primi cinque classificatori, ad eccezione del *K-Nearest Neighbors*. La classifica è, infatti, dominata da architetture di *deep learning* implementate con PYTORCH, che occupano quattro delle prime cinque posizioni. In generale, i modelli di *machine learning* implementati con SCIKIT-LEARN mostrano *performance* complessivamente inferiori, ma soprattutto una marcata instabilità tra i diversi *fold* di validazione, evidenziata dalle elevate deviazioni *standard* che accompagnano le metriche di accuratezza ed *F1-score*, mostrando una difficoltà di questi approcci nel generalizzare *pattern* complessi presenti in dati eterogenei, come nel caso del *dataset* in esame. Le *features* estratte dai descrittori, infatti, non catturano uno stato conformazionale ben definito fra le due classi, generando quindi una situazione in cui i classificatori falliscono a identificare un confine decisionale. Risulta evidente, quindi, come questi modelli siano altresì dipendenti dalle partizioni in ogni *fold* del *training set*.

Al contrario, le architetture neurali *fully-connected* (basate su MLP *Classifier*) dimostrano non solo prestazioni

superiori in termini assoluti, con il modello *Light Architecture* che raggiunge un'accuratezza di $73.7 \pm 6.4\%$ e un *F1-score* di $71.8 \pm 8.9\%$, ma anche una stabilità significativamente maggiore tra i *fold*.

Per il vero, è importante sottolineare come le *performance* complessive, sebbene modeste per gli *standard* di classificazione tradizionale, siano in realtà pienamente giustificabili con l'impostazione metodologica del progetto, sia con la natura biologica di queste biomolecole, oltre che con la complessa distribuzione dei dati evidenziata dall'analisi UMAP. Lo scopo di questa indagine è, difatti, la classificazione tramite un approccio *machine learning physics-based* dei *nanobodies*, nonchè la caratterizzazione fisico-chimica degli stessi in condizione *target-free*. Le simulazioni di dinamica molecolare (MD) sono state condotte in assenza del ligando specifico proprio con l'intento di catturare la variabilità conformazionale di ciascun *nanobody*. Di conseguenza, il *dataset* risultante è caratterizzato da un'elevata eterogeneità, poiché riflette il comportamento non vincolato delle proteine in soluzione.

Inoltre, questa variabilità riflette anche l'origine di tali composti proteici, in virtù del fatto che essi derivano da un processo di ricombinazione di tipo V(D)J, fenomeno che come spiegato nella Sezione 1.1.4, è peculiare per tutte le immunoglobuline. Per tale ragione, le *feature* estratte da queste simulazioni catturano quindi una forte variabilità di natura biologica, che rende di conseguenza la classificazione particolarmente complessa, dove la UMAP (Figura 5.1) ha confermato visivamente questa ridotta separabilità tra le classi e una sovrapposizione dei *cluster*.

In questo contesto, quindi, i risultati ottenuti, specialmente con le architetture neurali, sono da considerarsi più che soddisfacenti, in quanto sono comunque in grado di predire il meccanismo di riconoscimento grazie all'integrazione dei descrittori delle simulazioni di MD, nonostante una ridotta separabilità del *dataset*. La capacità dei modelli MLP di raggiungere buone prestazioni testimonia il successo del *workflow* adottato.

In considerazione di quanto esposto, le *performance* inferiore dei modelli di *machine learning* tradizionali rispetto alle architetture *deep learning* trova giustificazione in fattori matematici.

In prima battuta, risulta evidente come la capacità delle architetture neurali di approssimare funzioni non lineari, anche in presenza di dati ad alta dimensionalità, sia una soluzione ottimale al problema di classificazione oggetto del lavoro. Allo stesso tempo, modelli generalmente robusti ai fenomeni di *overfitting* grazie all'approccio di *bagging* [85], come ad esempio la *Random Forest*, non raggiungono prestazioni comparabili agli MLP *classifier*. Ciò avviene perché, pur essendo in grado di approssimare funzioni complesse, in *dataset* in cui i campioni *bootstrap* (che idealmente dovrebbero essere differenti) presentano elevato *overlap*, gli insiemi di addestramento risultano eccessivamente simili e quindi tra loro fortemente correlati. Da questa analisi, emerge quindi il *trade-off* fra prestazioni e complessità computazionale, evidenziando come per *dataset* altamente eterogenei, è necessario ricorrere ad architetture più sofisticate che permettano una gestione più fine degli iperparametri.

Si precisa che i risultati riportati in Tabella 5.1 sono stati calcolati considerando tutte le repliche (*snapshot*) di ogni *nanobody* (identificato da un ID unico), secondo lo schema descritto nella Sezione 4.3.1. Da una successiva analisi dei risultati predittivi, emerge che la maggior parte dei *nanobodies* (in particolare, 28 campioni, equamente ripartiti in 14 esemplari ortosterici e 14 allosterici) presenta una classificazione sempre corretta su tutte le 200 repliche, dimostrando l'accordo del modello per singola proteina. Tuttavia, in situazioni miste, in cui alcune repliche dello stesso *nanobody* sono classificate correttamente ed altre in modo errato, è stato adottato un criterio conservativo: un *nanobody* viene considerato correttamente classificato se almeno il 90% delle sue repliche sono classificate in maniera corretta; in caso contrario, viene considerato come errato.

L'applicazione di questo criterio ha permesso di identificare due casi dubbi: il *nanobody* 7fau_Nb1B11 (ortosterico), che con 182 repliche corrette su 200 (91.0%) supera la soglia del 90% e viene quindi considerato complessivamente corretto; e il nanobody 7f5g_DL4 (ortosterico), che con 117 predizioni corrette su 200 (58.5%) non raggiunge la soglia minima e viene quindi classificato come errato. Proprio l'esistenza di questi casi limite, sebbene numericamente minoritari, non fa che confermare la complessità del problema, emergente dalla sostanziale variabilità conformazionale che le simulazioni di MD sono in grado di catturare.

Tenuto conto di ciò, in termini assoluti, il modello migliore (MLP *Light Architecture*) riesce a classificare correttamente 15 *nanobodies* ortosterici e 14 allosterici (29/40), corrispondente al 72.5% del *pool* di proteine che costituiscono il *dataset* base.

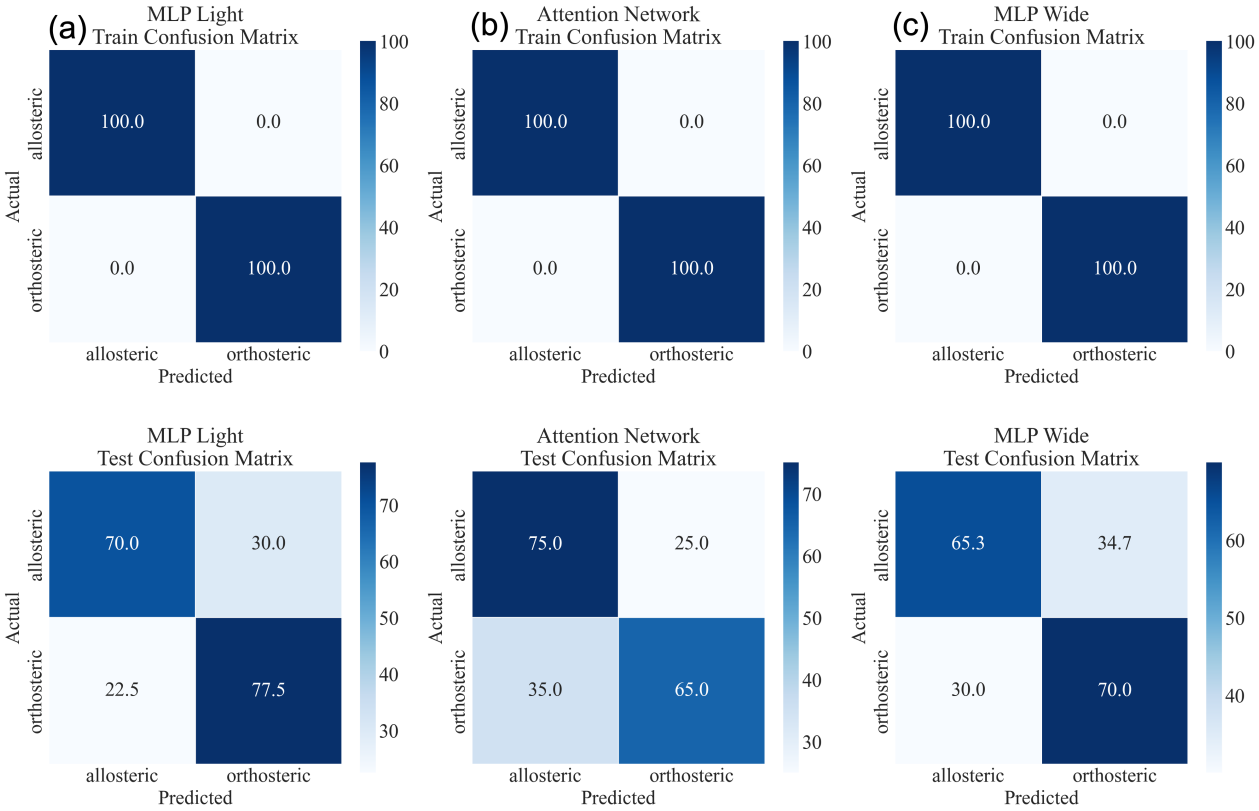

Figura 5.2: Matrici di confusione (in percentuale) dei tre classificatori con le migliori performance nel *dataset* base: colonna (a) MLP *Light*, colonna (b) *Attention Network*, e colonna (c) MLP *Wide*. I valori rappresentano la distribuzione percentuale delle predizioni sul *train* e *test set* aggregato da tutti i *fold* di *cross-validation*.

Per approfondire il comportamento dei modelli migliori, ridotti ai *top-3* per brevità, la Figura 5.2 mostra le matrici di confusione, in *train* e in *test* dei classificatori con le performance più elevate: MLP *Light Architecture*, *Attention Network* e MLP *Wide Architecture*.

Il modello MLP *Light Architecture*, nonostante raggiunga l'accuratezza media complessiva più alta (73.7%), e un *F1-score* medio di 71.8%, mostra una modesta asimmetria nella capacità discriminativa tra le due classi: in particolare, si osserva una leggera tendenza a classificare preferenzialmente i campioni ortosterici. Una situazione diametralmente opposta si verifica, invece, nel secondo modello migliore, cioè l'*Attention Network*, che tende a predire la classe allosterica.

Si nota come, alla luce di quanto precedentemente mostrato in Figura 5.1, dalle matrici di confusione si evince come entrambi i fenomeni di *overlap* portino ad un abbassamento delle *performance* dal *train* (praticamente perfette) al *test set*, come naturale conseguenza del problema di generalizzabilità e di riconoscimento di *pattern* dei classificatori.

La Tabella completa riportante i risultati per tutti i classificatori impiegati sul *dataset* base è riportata nell'Appendice di questo lavoro.

5.1.3 Analisi dell'importanza delle *features*

Pur a fronte di performance predittive complessivamente modeste, l'analisi dell'importanza delle *feature* evidenzia una capacità dei modelli di discriminare meccanismi di riconoscimento ortosterici e allosterici grazie a determinanti chimico–fisici rilevanti. Applicando le metodologie descritte nella Sezione 5.1.3, è stato

calcolato un *consensus score* che integra i risultati dei 13 modelli, a valle di un processo di normalizzazione e aggregazione. Tale approccio privilegia le *feature* che coniugano elevato potere discriminante, e al contempo anche bassa variabilità tra gli *split* del *dataset*, fornendo così una stima robusta e statisticamente affidabile della rilevanza sia tra i classificatori tra i diversi *split* della CV.

Tabella 5.2: *Top*-10 *feature* più importanti del *dataset* base ordinate per *consensus score* calcolato sui 13 modelli.

Rank	Feature Name	Consensus Score	Physical Category	Region
1	cdr3_eigvec_skew	0.812	MD - Eigenvector	CDR3
2	cdr3_eigvec_kurtosis	0.738	MD - Eigenvector	CDR3
3	cdr3_eigvec_peak_count	0.581	MD - Eigenvector	CDR3
4	fw_eigvec_skew_below_threshold	0.576	MD - Eigenvector	Framework
5	cdr3_mean_volume	0.564	Structural - Volume	CDR3
6	cdr3_rmsf_peak_count	0.491	MD - RMSF	CDR3
7	fw_eigvec_kurtosis	0.475	MD - Eigenvector	Framework
8	fw_mean_hydrophobicity	0.451	Structural - Hydrophobicity	Framework
9	fw_eigvec_l1_norm	0.433	MD - Eigenvector	Framework
10	cdr3_rmsf_kurtosis	0.429	MD - RMSF	CDR3

La Tabella 5.2 mette in evidenza *pattern* di particolare rilevanza. Innanzitutto, si osserva una live prevalenza di *feature* riferite alla regione CDR3, che occupano 6 delle 10 posizioni: un risultato coerente con le aspettative biologiche, poiché il CDR3 è tipicamente la porzione più direttamente coinvolta nel riconoscimento dell'antigene [11]. In particolare, la *kurtosis* (curtosi) e la *skewness* (assimetria) del contibuto del primo autovettore degli amminoacidi CDR3 emergono come i descrittori più discriminanti, suggerendo che aspetti riguardanti la distribuzione energetica, siano determinanti più promettenti per distinguere i meccanismi di legame. All'interno delle prime dieci posizioni dominano i descrittori MD di tipo *eigenvector*, descritti nella Sezione 4.1, e presenti in 6 casi su 10.

Accanto alle *feature* del CDR3, compaiono in modo assolutamente non marginale *feature* relative al *framework*, a sottolineare come le proprietà delle regioni conservate (il *backbone*/*core* dei *nanobodies*) contribuiscano in maniera significativa alla classificazione.

Per completare la composizione delle *feature* maggiormente rilevanti, la Tabella 5.3 riassume la distribuzione per categoria fisica, aggregando inoltre i contributi per macro–categoria, e riportando la ripartizione complessiva per regione.

Tabella 5.3: Distribuzione delle *top*-10 *feature* del *consensus score* (*dataset* base) raggruppate per categoria fisica e regione.

Physical Category	Feature Count	Percentage
MD - Eigenvector	6	60.0%
MD - RMSF	2	20.0%
Structural - Steric	1	10.0%
Structural - Hydrophobicity	1	10.0%
Macro-category totals (summed across subcategories)		
MD (overall)	8	80.0%
Structural (overall)	2	20.0%
Overall region distribution		
CDR3	6	60.0%
Framework	4	40.0%

L'analisi per categoria fisica (Tabella 5.3) evidenzia la netta predominanza delle *feature* di dinamica molecolare, le quali, nel loro complesso (MD *overall*), costituiscono l'80.0% delle *top-10 feature*. Tale risultato conferma la validità dell'impianto sperimentale adottato, in quanto le simulazioni di MD si dimostrano una metodologia solida e informativa per lo studio delle proprietà di queste biomolecole, creando dei modelli di apprendimento *physics-based* efficienti.

La componente RMSF contribuisce per un ulteriore 20%, suggerendo che anche la flessibilità atomica media/locale concorre in maniera apprezzabile.

Dall'analisi della tabella si evince, inoltre, che le *feature* di costituzione amminoacidica non rappresentano un segnale discriminante preponderante, contribuendo solo per il 20% della classificazione. Questo risultato costituisce un'importante conferma biologica e predittiva: come descritto nella Sezione 1.1.4, il meccanismo V(D)J genera infatti una variabilità tale da non garantire l'identificazione di motivi amminoacidici ripetuti o di *pattern* strutturali.

Per il vero, è doveroso sottolineare come, sebbene le *feature* di *time-series* non siano presenti nel *rank consensus* finale, il loro contributo nel lavoro sia stato di fondamentale importanza sotto un duplice profilo. In primo luogo, l'impiego di descrittori temporali, quali la propensione ad assumere una conformazione ad α-elica o a foglietto β (4.9) e la metrica di distanza temporale (4.10), risponde a un'esigenza di carattere biologico-strutturale. Tali descrittori permettono di catturare aspetti dinamici alla proteina, rendendo conto della sua evoluzione conformazionale in soluzione, in virtù della loro rilevanza nel riconoscimento dell'epitopo (Sezione 4.1.3).

In secondo luogo, queste *features* hanno permesso di implementare una robusta strategia di *data augmentation*. Attraverso un *sampling via-snapshot*, è stato possibile espandere il *dataset* iniziale di *nanobodies*, generando un campionario più ampio che cattura ciascun *nanobody* in stati conformazionali specifici. Questa strategia, la cui configurazione è stata ottimizzata empiricamente, si è rivelata determinante per il miglioramento delle prestazioni dei modelli.

A valle dell'analisi di consensus, è interessante esaminare le *feature* più rilevanti secondo il modello con le migliori *performance* predittive (MLP *Light Architecture*). La Tabella 5.4 riporta le prime 10 *feature* per importanza normalizzata (scala 0-1), valori che si sono dimostrati coerenti attraverso tutti i 5 *split* della *cross-validation*.

Tabella 5.4: MLP *Light Architecture* — Top-10 *features* per importanza normalizzata (0–1) e distribuzione per categoria fisica nei 5 *split* di CV.

Rank	Feature	Normalized Importance
1	cdr3_eigvec_skew	1.000
2	cdr3_eigvec_peak_count	0.735
3	cdr3_rmsf_peak_count	0.727
4	fw_eigvec_kurtosis	0.725
5	cdr3_eigvec_kurtosis	0.647
6	fw_eigvec_skew_below_threshold	0.601
7	cdr3_df_kurtosis	0.568
8	fw_rmsf_skew	0.499
9	cdr3_structural_entropy	0.491
10	fw_eigvec_kurtosis_below_threshold	0.483

-	Feature Count	Percentage
Macro-category totals		
MD (overall)	9	90.0%
Structural (overall)	1	10.0%
Region distribution		
CDR3	6	60.0%
Framework	4	40.0%

Tra queste, le *feature* derivate dal metodo REBELOT (4.7) da sole pesano il 60% delle *top-10*, a evidenza del fatto che le proprietà energetiche costituiscono il segnale potenzialmente più informativo. Questo, inoltre, è conforme alla letteratura, e conferma quanto dimostrato da Bagordo e Trèves et al. [62]. Infatti, gli Autori mostrano che le regioni identificate come CDR da NANOCDR-X presentano una *signature* energetica caratteristica: risultano debolmente accoppiate al *fold* globale (*weakly coupled*), come evidenziato da un'analisi complementare basata su decomposizione energetica (MLCE, Eq. 4.8), eseguita anch'essa attraverso il metodo REBELOT. Su 121 *nanobodies*, gli Autori dimostrano che le *patches*, ovvero le regioni a basso accoppiamento, includono residui di CDR3 in 112 casi (circa 91%), sostenendo un contesto in cui la *fuzziness*, ovvero il grado di disordine/adattabilità intrinseci di questi composti, emerge come tratto distintivo delle CDR (in particolar modo della CDR3) e spiega la loro plasticità conformazionale e polireattività per il riconoscimento dell'epitopo.

La presenza di *feature* come `cdr3_structural_entropy`, che non compariva nella *top*-10 del *rank consensus* ma si rivela importante per il modello migliore, suggerisce come il modello neurale sia in grado di cogliere, in maniera seppur modesta, ulteriori *pattern* legati alla struttura amminoacidica del *loop* ipervariabile.

In generale, le *features* di natura energetica costituiscono un interessante *insight* per la caratterizzazione fisico-chimica di queste biomolecole, e in particolar modo per la predizione *target-free* del *binding* con l'epitopo. Tuttavia, per una valutazione completa di carattere *structural-biology*, risulta necessario considerare tutte le proprietà emergenti da questa analisi.

5.1.4 Analisi statistica

L'identificazione delle *feature* più discriminanti, sebbene informativa, costituisce un risultato di per sé parziale senza un'adeguata validazione inferenziale. Risulta infatti fondamentale accertare che le differenze tra classi osservate per queste *feature* non siano meri artefatti algoritmici dovuti a *bias* modello-specifico, ma riflettano distinzioni di natura chimico-fisica supportate da evidenza statistica.

Per confermare questo aspetto, è stato applicato il protocollo descritto in Materiali e Metodi (Sezione 5.1.4) a ciascuna delle *top*-10 *feature* emerse dal *consensus score* (Tabella 5.2).

Tabella 5.5: Esito dei test inferenziali sulle *top-10* feature del *consensus* sul *dataset* base. Per ciascuna *feature* viene indicato il *test* selezionato dall'algoritmo decisionale e l'esito del test rispetto alla soglia $\alpha = 0.005$.

Rank	Feature	Test selezionato	Esito ($p_{\text{final}} < 0.005$)
1	**cdr3_eigvec_kurtosis**	**Wilcoxon**	**Sì**
2	**cdr3_eigvec_skew**	***t*-test**	**Sì**
3	cdr3_mean_volume	Wilcoxon	No
4	fw_eigvec_skew_below_threshold	*t*-test	No
5	cdr3_eigvec_peak_count	*t*-test	No
6	cdr3_eigvec_l1_norm_below_threshold	*t*-test	No
7	cdr3_rmsf_peak_count	Wilcoxon	No
8	fw_eigvec_kurtosis	*t*-test	No
9	cdr3_rmsf_kurtosis	*t*-test	No
10	fw_eigvec_peak_count	*t*-test	No

I risultati di questa analisi inferenziale, riportati in Tabella 5.5, mostrano che solo due *feature* superano il *test* statistico applicato, e presentano un *p-value* al di sotto della soglia di significatività: `cdr3_eigvec_kurtosis` e `cdr3_eigvec_skew`. Entrambe appartengono al raggruppamento fisico *MD – Eigenvector* (metodo REBELOT) della regione CDR3, un risultato che rafforza la coerenza con l'interpretazione energetica discussa nei risultati nella Sezione 5.1.3.

Queste due *feature* sono degli indicatori statistici applicati alla distribuzione del contributo del primo autovettore v_i^1 (4.7) per i residui del *loop* CDR3. Come descritto in dettaglio nella Sezione 4.1, ciascun valore v_i^1 rappresenta il contributo energetico del residuo i-esimo al modo di interazione cooperativo globalmente più stabilizzante della proteina, ottenuto dalla decomposizione spettrale della matrice delle interazioni residuo-residuo. Pertanto, la distribuzione su cui vengono calcolati curtosi e asimmetria descrive quanto siano energeticamente accoppiati al *backbone* proteico i singoli amminoacidi del CDR3.

Nello specifico, la `cdr3_eigvec_kurtosis` (curtosi) quantifica la concentrazione di questa distribuzione, indicando se i contributi energetici dei residui CDR3 sono perlopiù uniformi o, al contrario, concentrati in pochi valori significativi di energia. Invece, la `cdr3_eigvec_skew` (asimmetria) ne misura l'asimmetria (*skewness*), rivelando uno sbilanciamento nella distribuzione tra residui fortemente e debolmente accoppiati.

Risulta importante sottolineare come la non significatività univariata delle altre *feature* non ne sminuisce l'importanza in un contesto predittivo. Infatti, un *test* statistico univariato valuta la capacità di separare le classi una feature alla volta, mentre un classificatore considera l'insieme di tutte le *feature* che costituiscono il *dataset*.

Coerentemente con il protocollo descritto in Sezione 4.3.11, per le due *feature* risultate significative sono state generate delle visualizzazioni che riportano, oltre alle distribuzioni di densità (KDE) per classe, le medie rappresentative (linee tratteggiate, Eq. 4.38)

Si riportano i risultati ottenuti nelle Figure 5.3 e 5.4.

Per la `cdr3_eigvec_kurtosis` (Figura 5.3), il test di Wilcoxon conferma una differenza statisticamente significativa ($p = 0.0045$). I *nanobodies* ortosterici mostrano un valor medio (4.38) di curtosi positivo, indicando una distribuzione ipoteticamente più leptocurtica caratterizzata da una campana centrale più stretta con code pesanti e maggior presenza di valori estremi. Al contrario, i *nanobodies* allosterici presentano una curtosi media negativa, corrispondente a una distribuzione ipoteticamente platicurtica.

Per la `cdr3_eigvec_skew` (Figura 5.4) si osserva una separazione ancor più marcata, come evidenziato dal *t-test* ($p = 0.0020$).

La classe ortosterica mostra un valor medio di *skew* più elevato, indicando una coda destra più marcata nella distribuzione dei contributi energetici, mentre i *nanobodies* allosterici presentano un valor medio della

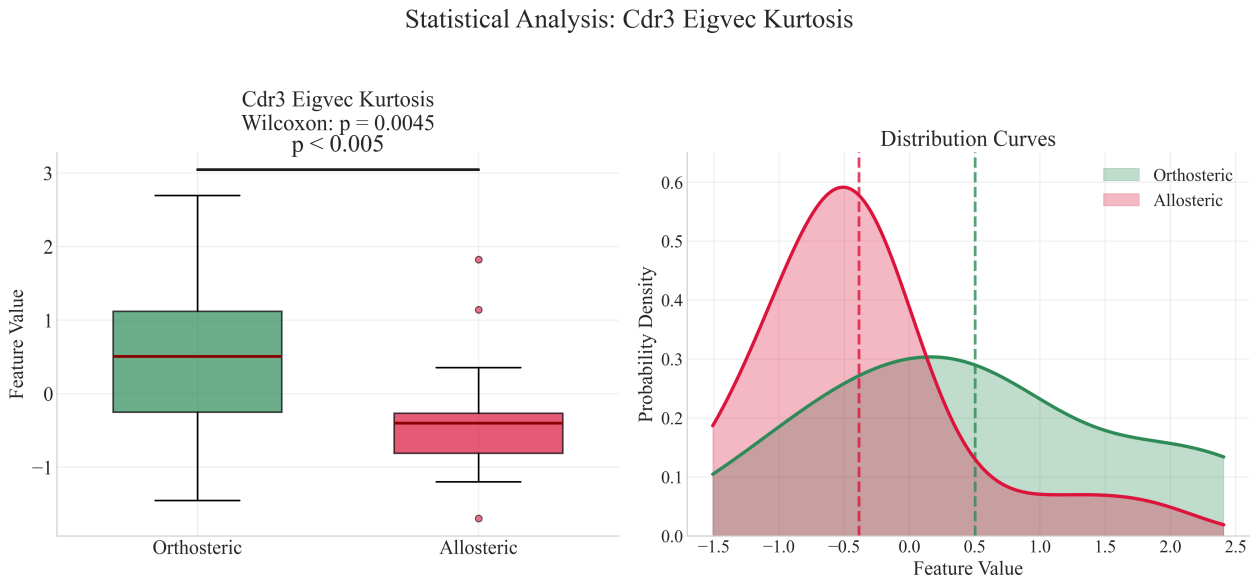

Figura 5.3: Distribuzione della `cdr3_eigvec_kurtosis` nel *dataset* base.

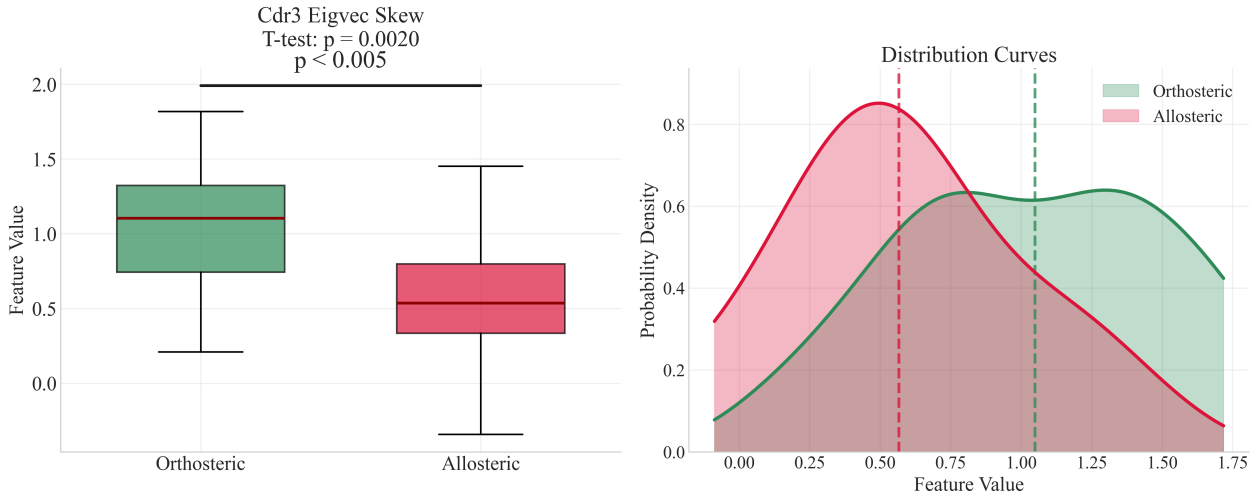

Figura 5.4: Distribuzione della `cdr3_eigvec_skew` nel *dataset* base.

distribuzione più basso, che tende ad essere più simmetrica (maggiormente *gaussian-like*), con contributi energetici più uniformi tra i residui del CDR3.

Questi risultati suggeriscono che i *nanobodies* ortosterici sono caratterizzati da una distribuzione energetica più eterogenea nel CDR3, con pochi residui fortemente accoppiati energeticamente al *framework* proteico (coda destra lunga). Al contrario, i *nanobodies* allosterici mostrano una distribuzione energetica più uniforme e simmetrica, compatibile con un accoppiamento energetico più diffuso e meno polarizzato tra i residui del *loop*.

Come descritto nella Sezione 4.3.11, per verificare la correttezza delle interpretazioni emergenti sulle *feature* preminenti (*skewness* e *kurtosis*), e fornire quindi una interpretazione fisica diretta dei risultati, è stata analizzata la distribuzione effettiva dei valori del primo autovettore (v_i^1) per i residui della regione CDR3. Tale analisi consente di visualizzare direttamente le differenze nelle due distribuzioni.

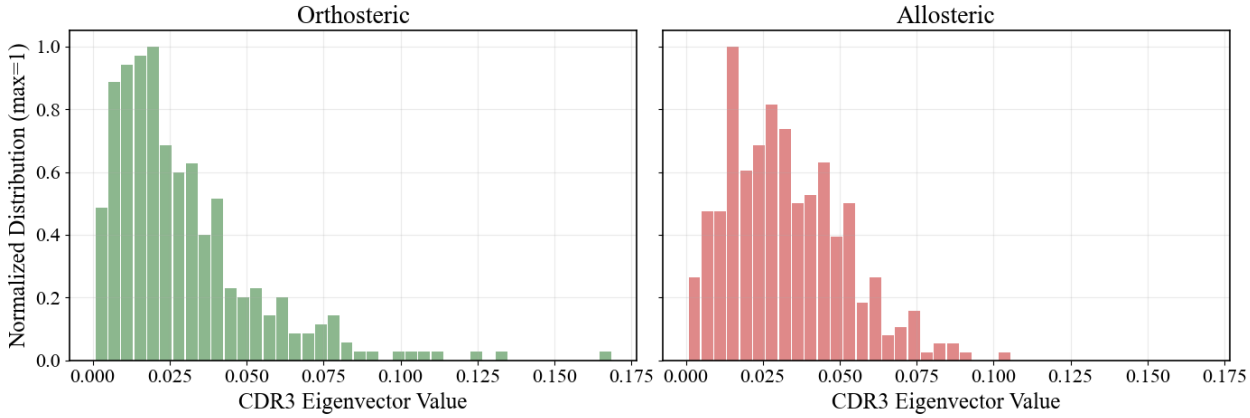

Figura 5.5: Distribuzione normalizzata del contributo al primo autovettore nel *dataset* base.

La Figura 5.5 fornisce un significato fisico concreto. Per i *nanobodies* ortosterici, la distribuzione mostra una marcata coda destra e la maggior parte dei residui del CDR3 che si concentrano in valori energetici molto bassi, ma con la presenza di alcuni residui altamente energetici. Questa configurazione evidenzia la presenza di specifici residui fortemente accoppiati energeticamente al *framework* proteico. Sebbene questi residui costituiscono una piccola parte, risultano essenziali per la stabilizzazione selettiva di alcune porzioni del CDR3, ottimizzando così il riconoscimento ortosterico. Al contrario, la maggioranza dei restanti residui del CDR3 risulta essere fortemente disaccoppiato al *core* proteico, suggerendo quindi un meccanismo di riconoscimento peculiare per il sito ortosterico.

I *nanobodies* allosterici presentano invece una distribuzione più uniforme che tende a una distribuzione *gaussian-like*, con una leggera asimmetria destra. Questa configurazione indica un CDR3 uniformemente disaccoppiato, dove l'energia di interazione è distribuita in modo più omogeneo tra i residui.

Questi dati permettono quindi di ipotizzare un meccanismo di riconoscimento dell'epitopo diverso per le due classi. In particolare, è possibile ipotizzare che questo sia guidato in generale dal disordine adattivo dei CDR3 e dalla tendenza dei residui che li compongono a interagire con l'antigene, misurata tramite il disaccoppiamento energetico. Nello specifico, i residui dei *nanobodies* ortosterici presentano residui tendenzialmente più disaccoppiati energeticamente e quindi con valori più bassi di v_i^1, e in alcuni casi possono presentare pochi ma (probabilmente) fondamentali residui fortemente accoppiati al *core* proteico. Al contrario, i *nanobodies* allosterici presentano valori meno estremi di *eigenvector*, ma distribuiti più uniformemente lungo il CDR3.

Questa differenza può guidare ipoteticamente il riconoscimento selettivo dell'epitopo.

5.2 Risultati sul *Dataset Extended*

Per validare la robustezza e la generalizzabilità dei modelli sviluppati, è stata condotta un'analisi supplementare su una coorte estesa di 50 *nanobodies*. Il *dataset* esteso può essere formalizzato come una matrice $D_{\text{ext}} \in \mathbb{R}^{n_{\text{ext}} \times m}$, dove $n_{\text{ext}} = 50\,\textit{nanobodies} \times 200\,\textit{snapshot} = 10000$ rappresenta il numero totale di campioni (repliche) e $m = 167$ il numero di *feature* mantenute, invariate anche in termini di ripartizione rispetto al *dataset* base (Sezione 5.1).

5.2.1 Risultati dell'analisi esplorativa tramite UMAP

Per validare la complessità distributiva riscontrata nel *dataset* base, e di conseguenza la difficoltà del *task* di classificazione, è stata ripetuta l'analisi UMAP sulla coorte estesa di 50 *nanobodies* (Sezione 4.3.7).

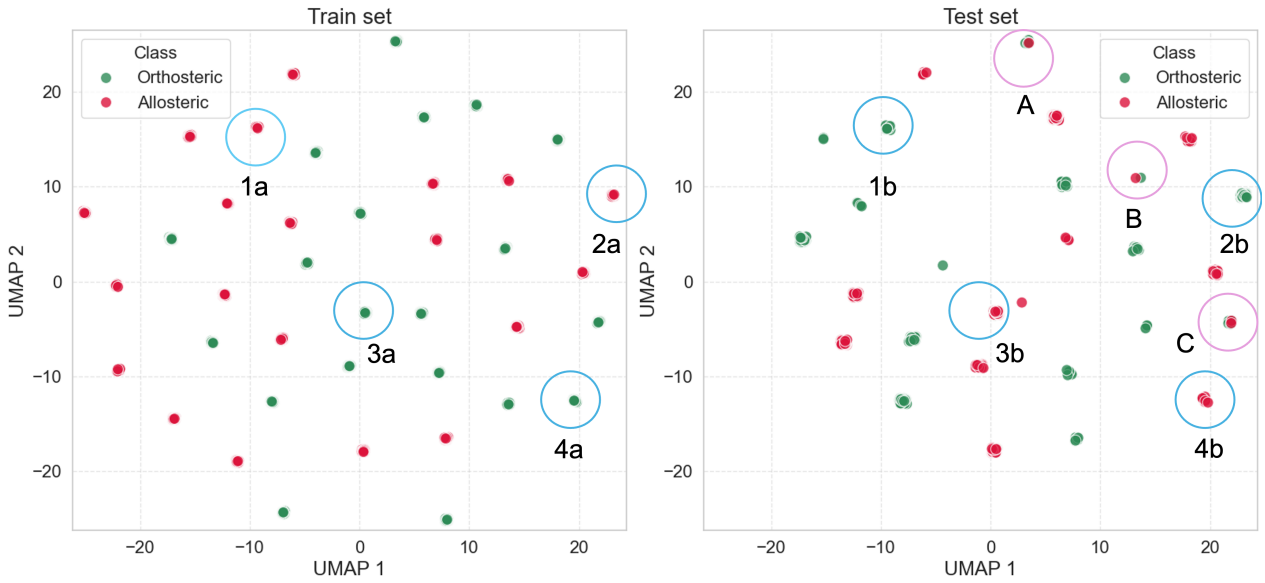

Figura 5.6: Proiezione UMAP del *dataset extended*, in uno degli *split* della *cross-validation*.

Come evidenziato da alcuni esempi evidenziati dai cerchi rosa (e dalle lettere maiuscole) e dai cerchi azzurri (e dalla coppia di lettere minuscole), anche in questo caso si osserva un marcato *overlap intra-set* e *overlap inter-set* rispettivamente, confermando quanto è emerso nel *dataset* base (Sezione 5.1).

5.2.2 Risultati predittivi

Tabella 5.6: *Performance* comparative dei *top-5* classificatori sul *dataset extended*. I valori sono calcolati su GS5FCV e sono espressi come media ± deviazione *standard* (i valori sono espressi in percentuale, eccetto MCC).

Modello	ACC (%)	TPR (%)	TNR (%)	F1 (%)	PPV (%)	MCC
MLP - Deep Architecture (PyTorch)	69.7±6.2	67.8±16.0	71.6±20.5	68.4±7.4	75.5±14.5	0.422±0.125
MLP - Wide Architecture (PyTorch)	69.3±8.1	66.3±15.0	72.3±24.8	68.0±7.7	77.1±17.4	0.418±0.167
K-Nearest Neighbors (scikit-learn)	67.4±18.7	73.0±21.2	61.8±20.6	68.8±19.2	65.8±18.9	0.355±0.374
SVM (scikit-learn)	65.7±14.5	59.9±12.6	71.4±26.6	64.0±12.5	74.3±21.6	0.337±0.302
MLP - Light Architecture (PyTorch)	65.5±5.2	62.3±13.5	68.7±20.5	63.8±4.6	71.7±15.4	0.335±0.125

Come dimostrato precedentemente nel caso del *dataset* base (Sezione 5.1.2), anche nel caso *extended*, dall'analisi comparativa dei modelli di classificazione presentati in Tabella 5.6, emerge come i *classifier* MLP siano ancora i modelli maggiormente performanti su questa tipologia di *dataset*, sia in termini di valor medio in CV, ma anche quelli maggiormente stabili.

Una differenza rispetto al caso base è la presenza, tra i primi cinque migliori classificatori, di un ulteriore modello di *machine learning* implementato con SCIKIT-LEARN, l'SVM. Insieme al *K-Nearest Neighbors*, esso mostra una marcata instabilità di *accuracy* ed *F1-score* tra i diversi *fold*, come evidenziato dall'elevata deviazione *standard*. Questo risultato conferma l'inadeguatezza degli approcci non neurali in entrambi i *dataset*.

Per questo motivo, i risultati ottenuti in questa fase, specialmente con la conferma delle prestazioni migliori di architetture neurali *fully-connected*, consentono di mettere in evidenza un intrinseco *plateau* prestazionale (*performance cap*), sia di natura biologica, sia derivante dall'impostazione metodologica adottata nel presente studio (ovvero simulazioni di MD eseguite in condizioni *target-free*), per le stesse motivazioni avanzate nella Sezione 5.1.2.

Anche in questo caso, come nel *dataset* base, le metriche riportate nella Tabella 5.6, sono calcolate su tutti gli *snapshot*, e dalla successiva analisi approfondita dei risultati emerge una situazione analoga di disaccordo fra le repliche di uno stesso *nanobody*. Dei 50 *nanobodies* testati, 16 della classe ortosterica e 15 della classe allosterica sono stati classificati sempre correttamente in tutte le loro repliche, dimostrando una notevole robustezza predittiva. Tuttavia, permangono situazioni dubbie: per la classe ortosterica, i *nanobodies* 6yz5_H11-D4 (97.5%) e 7z1c_B5 (93.0%) presentano accuratezze superiori alla soglia del 90%, considerabili quindi complessivamente corretti. Parimenti, per la classe allosterica, i casi di 7x2m_1-2C7 (99.0%) e 8elq_C4-255 (97.0%) superano ampiamente la soglia minima. Alla luce di questi casi, il modello migliore (MLP *Deep Architecture*) nel caso *extended* riesce a classificare correttamente 18 esemplari ortosterici e 17 allosterici (35/50), che equivale al 70% dei *nanobodies* totali.

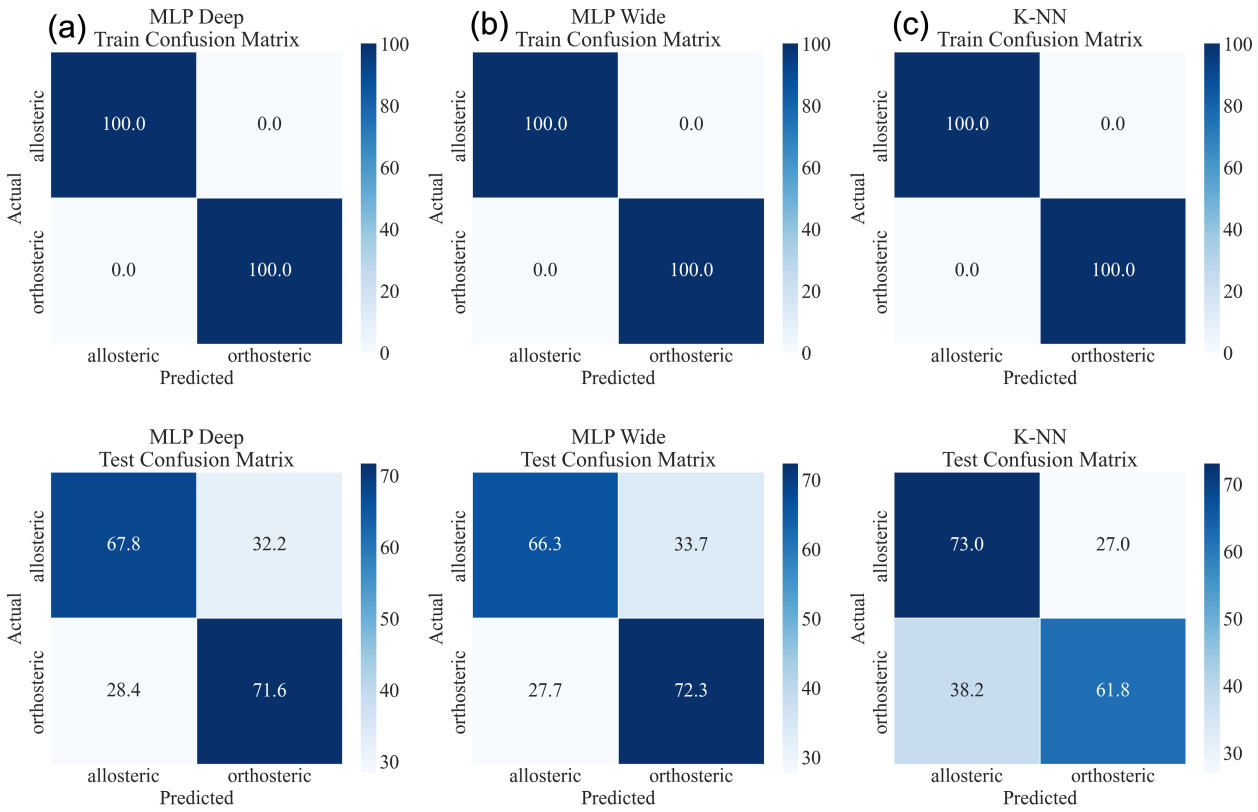

Figura 5.7: Matrici di confusione (in percentuale) dei tre classificatori con le migliori performance nel *dataset extended*: colonna (a) MLP *Deep*, colonna (b) MLP *Wide*, e colonna (c) K-NN. I valori rappresentano la distribuzione percentuale delle predizioni sul *train* e *test set* aggregato da tutti i *fold* di *cross-validation*.

In Figura 5.7 sono riportate le matrici di confusione dei tre classificatori con le *performance* più elevate. Anche in questo caso, valgono le stesse considerazioni riportate nella Sezione 5.1.2.

La Tabella completa riportante i risultati per tutti i classificatori impiegati sul *dataset extended* è riportata nell'Appendice.

5.2.3 Confronto delle *performance* tra *dataset base* ed *extended*

Il confronto tra i dataset base ed *extended*, riportato in Tabella 5.7, rivela diversi aspetti significativi riguardanti la robustezza dei modelli.

Si osserva che l'architettura neurale MLP implementata in *PyTorch* mantiene prestazioni confrontabili anche con il dataset *extended*, dimostrando una notevole stabilità anche a valle dell'introduzione di nuovi *nanobodies* (aumento di variabilità introdotta pari al 25% rispetto al caso base). Di conseguenza, il calo prestazionale, risulta comunque contenuto: -4.0% medio in *accuracy* e -3.4% medio in *F1*.

Un aspetto particolarmente significativo emerge dall'analisi architetturale: mentre sul *dataset* base il modello ottimale era l'MLP *Light Architecture* (costituito da 2 *hidden layers*), sul dataset *extended* la migliore *performance* è ottenuta dall'MLP *Deep Architecture* (4 *hidden layers*).

Per una spiegazione approfondita delle architetture si rimanda all'Appendice.

Questo *shift* è coerente con le aspettative teoriche: l'aumento della variabilità richiede modelli in grado di catturare *pattern* più complessi, grazie ad una rete più profonda. L'MLP *Deep Architecture* si dimostra quindi più adatto a gestire l'eterogeneità introdotta dai nuovi *nanobodies*.

In definitiva è possibile affermare come le architetture neurali, in particolare modelli tipo MLP *classifier*, si confermano l'approccio più appropriato per questo problema di classificazione, caratterizzato da alta dimensionalità e relazioni non lineari (fra cui l'*overlap*, emergente da indagini visuali) tra le *features*.

Tabella 5.7: Confronto delle *performance* del modello migliore tra *dataset* base ed *extended*.

Modello	**ACC (%)**	**TPR (%)**	**TNR (%)**	**F1 (%)**	**PPV (%)**	**MCC**
Best* (*Dataset* Base): MLP - *Light Architecture (PyTorch)	73.7±6.4	70.0±18.7	77.5±17.0	71.8±8.9	78.9±13.1	0.509±0.137
Best* (*Dataset Extended*): MLP - *Deep Architecture (PyTorch)	69.7±6.2	67.8±16.0	71.6±20.5	68.4±7.4	75.5±14.5	0.422±0.125
avg. Δ (*Light* − *Deep*)	-4.0	-2.2	-5.9	-3.4	-3.4	-0.087

Questo esito, corrispondente ad un lieve calo delle prestazioni del miglior classificatore, è riconducibile alla metodologia adottata: per costruzione, i classificatori considerano ogni *snapshot* come un'istanza; di conseguenza, i casi *mixed* (disaccordo fra repliche di uno stesso *nanobody*), maggiormente presenti nel caso *extended* rispetto al caso base, tendono ad abbassare le metriche complessive.

5.2.4 Confronto con altri metodi computazionali

Complessivamente, questi risultati, se confrontati con metodologie che non impiegano la dinamica molecolare (Sezione 1.4) risultano incoraggianti.

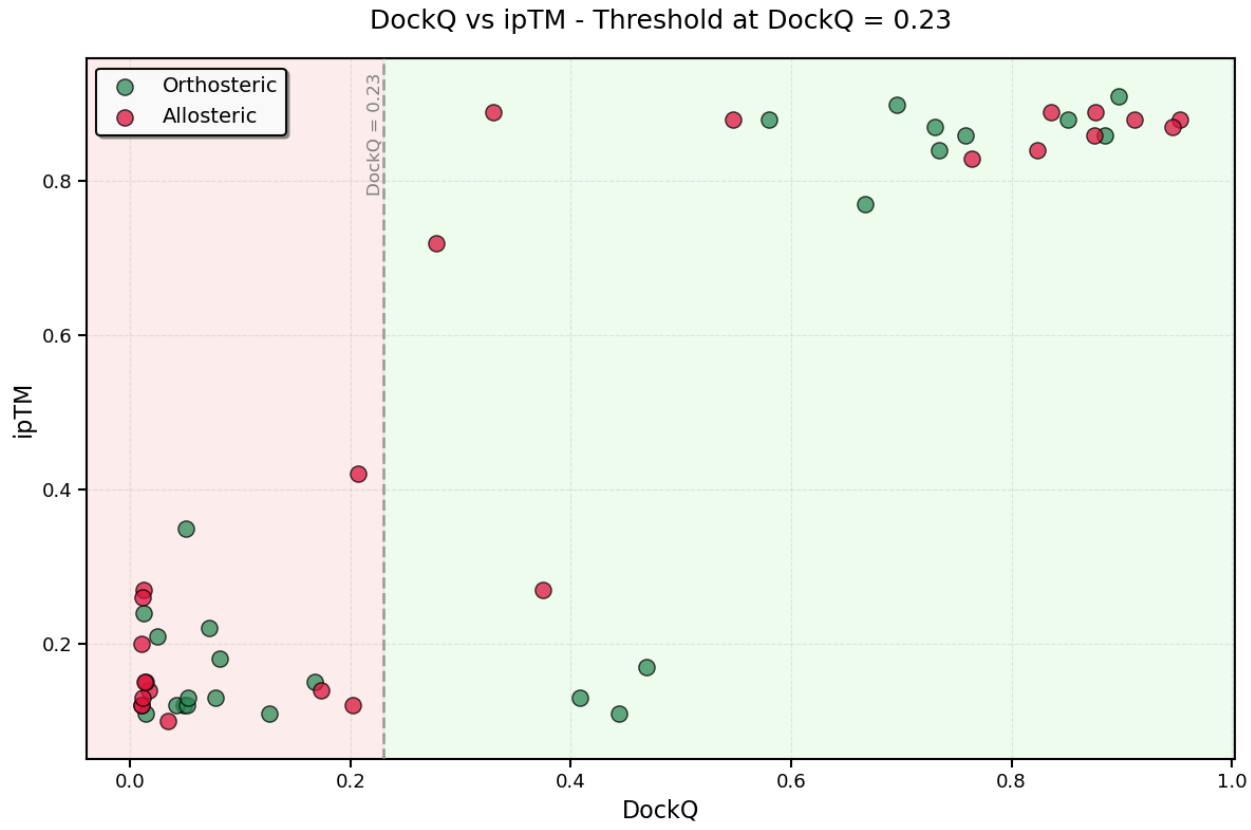

Figura 5.8: *Scatter plot* DockQ–ipTM dei complessi *nanobody*–RBD (50 *nanobodies*): la soglia DockQ = 0,23 (linea tratteggiata) separa la regione dei campioni classificati come *incorrect* (sfondo rosato) da quelli accettabili (sfondo verde).

Per quanto riguarda l'utilizzo di AF3 (Sezione 1.4.1), in Figura 5.8 si osserva un buon accordo tra le due metriche: le predizioni strutturalmente errate (DockQ ≤ 0.23) risultano, nella maggior parte dei casi, associate a valori di ipTM inferiori a 0.8 (spesso < 0.6), in accordo con una bassa confidenza del modello. La regione

centrale del grafico mette in evidenza i casi di confine, in cui la confidenza interna e la qualità esterna non risultano perfettamente concordi.

In alcuni esempi AF3 colloca il *nanobody* nella regione corretta dell'RBD, ma il DockQ resta ≤ 0.23. Una possibile interpretazione è la presenza di un disallineamento o di una rototraslazione del *nanobody* rispetto al sito nativo, che penalizza complessivamente le metriche.

In definitiva i risultati mostrano come AF3 predice correttamente soltanto 24 complessi su 50 ($24/50 \approx 48\%$ del totale), un risultato inferiore al 50% che conferma come il problema resti complesso anche includendo in *input* la sequenza del RBD del bersaglio.

Per questo motivo, le predizioni di AF3 sono state rivalutate utilizzando il criterio geometrico descritto nella Sezione 1.4.2. In particolare, per ciascun complesso si considerano le distanze minime $d_1^{\min}$ e $d_2^{\min}$ tra il centro di massa dell'interfaccia del paratopo e i due epitopi (1.5); applicando la regola decisionale (1.6), si stabilisce quale epitopo è assegnato da AF3 come più vicino, e quindi come più probabile. Una predizione è definita corretta se l'epitopo così assegnato coincide con la classe effettiva. Sulla base di questo criterio, AF3 assegna correttamente l'epitopo in 41 complessi su 50 ($41/50 \approx 82\%$), migliorando sensibilmente rispetto alla valutazione basata sul solo DockQ, che è fortemente influenzata dall'accuratezza globale della superficie di contatto.

In maniera del tutto analoga, impiegando il *docking* proteina-proteina con BIOLUMINATE (Sezione 1.4.3), i risultati mettono in evidenza diversi limiti. In particolare, solo 28 *nanobodies* su 50 vengono correttamente classificati sulla base del *docking* ($28/50 \approx 56\%$ del totale); oltre alle prestazioni nettamente inferiori di questo approccio classico, emerge inoltre un marcato *bias*. Infatti, dei 28 casi corretti, ben 23 riguardano esclusivamente le predizioni della classe ortosterica (23 casi rispetto ai 25 *nanobodies* ortosterici selezionati), mentre i restanti 5 appartengono alla classe allosterica.

Nel confronto tra i due epitopi, questo risultato evidenzia come il *docking* tenda a privilegiare la previsione di un meccanismo ortosterico e, in generale, si dimostri inadeguato a fornire una previsione bilanciata del meccanismo di legame quando la classe del *nanobody* non sia nota a priori, essendo fortemente sbilanciato verso la classificazione degli ortosterici.

Di seguito sono presentati due esempi riassuntivi di tale problematica.

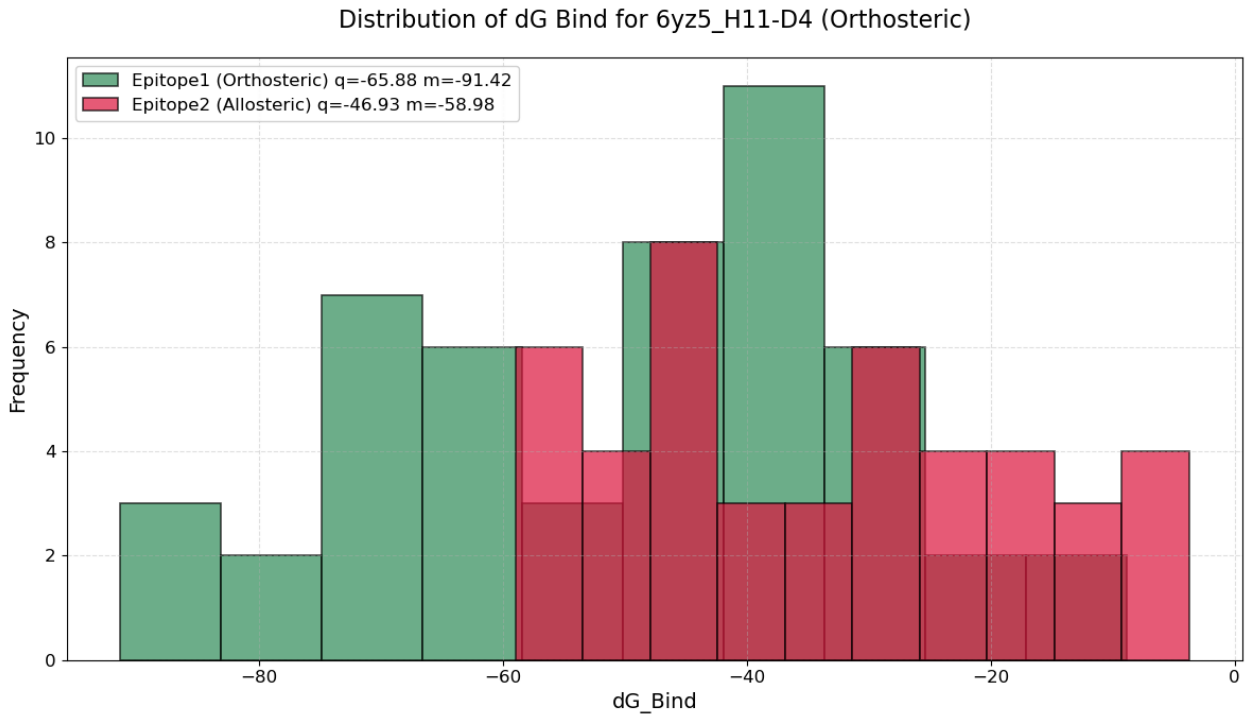

Figura 5.9: Istogramma dei valori dei valori di ΔG_{bind} nel caso del *nanobody* 6yz5_H11–D4 (ortosterico).

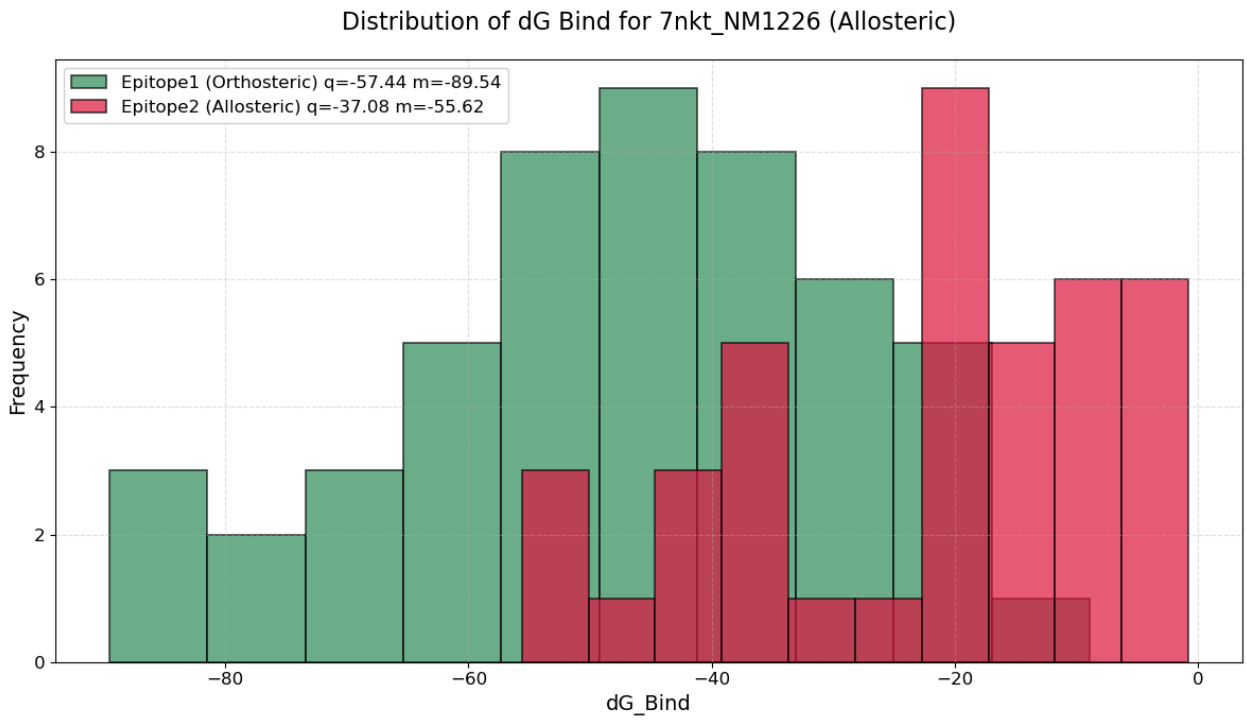

Figura 5.10: Istogramma dei valori dei valori di ΔG_{bind} nel caso del *nanobody* 7nkt_NM1226 (allosterico).

Nella Figura 5.9 è riportato il caso del *nanobody* 6yz5_H11–D4, appartenente alla classe ortosterica, che viene correttamente predetto in quanto il quartile più negativo (1.10) è quello relativo all'epitopo 1 (in verde), cioè l'epitopo effettivamente riconosciuto dal *nanobody*.

Viceversa, la Figura 5.10, relativa al *nanobody* 7nkt_NM1226 (allosterico), illustra un caso in cui il quartile più negativo è quello associato all'epitopo 1 e, di conseguenza, il *nanobody* viene predetto come appartenente alla classe sbagliata.

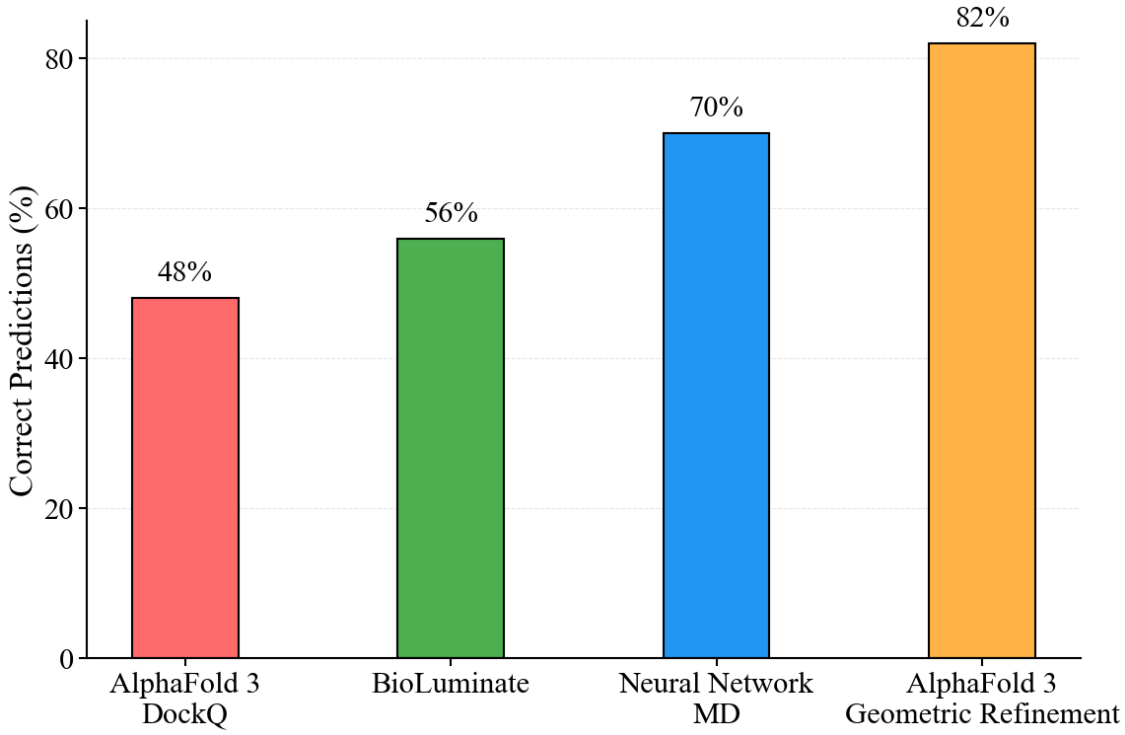

Figura 5.11: Confronto riassuntivo tra la percentuale di *nanobodies* correttamente classificati da AF3–DockQ, BIOLUMINATE, modello neurale *physics-informed* sviluppato e AF3 con le integrazioni di *refinement* del metodo geometrico.

In Figura 5.11 sono riassunte le prestazioni dei tre approcci considerati. Si osserva come il metodo sviluppato in questo lavoro (la *neural network physics-informed*) ottenga risultati superiori rispetto al *docking* classico eseguito con BIOLUMINATE, e solo lievemente inferiori rispetto ad AF3 corretto in base alle informazioni geometriche sugli epitopi *target*.

Si noti, inoltre, che i tre metodi producono *output* eterogenei e si fondano su presupposti metodologici differenti; di conseguenza, essi vanno intesi non come concorrenti diretti, ma come strumenti tra loro complementari. In particolar modo, la natura dei due modelli di *deep learning* impiegati è concettualmente diversa. AF3 è un predittore *sequence-based*, che opera direttamente sulle sequenze aminoacidiche di RBD e *nanobody*, e fornisce una predizione esplicita della struttura tridimensionale del complesso RBD–*nanobody*, predicendo la geometria dell'interfaccia e l'orientazione del paratopo sul bersaglio, informazioni che il modello neurale sviluppato in questo lavoro non è progettato per ricostruire. Al contrario, la rete neurale *physics-informed* è un modello *target-free* che non utilizza le sequenze bersaglio come *input* diretto, ma sfrutta descrittori *physics-based* derivati da simulazioni di MD per inferire la preferenza epitopica del *nanobody* con l'obiettivo di generalizzare il riconoscimento non solo al contesto virologico dell'RBD. In questa prospettiva, gli esiti dei diversi approcci non configurano una competizione, bensì risultati paralleli e integrabili: il grado di accordo tra la classificazione del *binding* sull'epitopo fornita dal modello *physics-informed* e la predizione strutturale di AF3 rappresentano due evidenze indipendenti a supporto della stessa ipotesi di legame, aumentando la robustezza complessiva della predizione.

La Tabella completa riportante i risultati di tutti i metodi impiegati, divisi per *nanobody* ID, è riportata nell'Appendice.

5.2.5 Analisi dell'importanza delle *features*

Per validare la solidità dei determinanti fisico-chimici identificati nell'analisi sul *dataset* base (Sezione 5.1.3), la *feature importance analysis* è stata replicata sulla coorte estesa di 50 *nanobodies*. L'obiettivo è

validare se i descrittori più significativi rimangono consistenti su un *dataset* più ampio, così da confermare la generalizzabilità delle evidenze biologico-strutturali e chimico-fisiche.

Tabella 5.8: *Top*-10 *feature* più importanti del *dataset extended* ordinate per *consensus score* calcolato sui 13 modelli.

Rank	Feature Name	Consensus Score	Physical Category	Region
1	cdr3_eigvec_skew	0.828	MD - Eigenvector	CDR3
2	cdr3_eigvec_kurtosis	0.672	MD - Eigenvector	CDR3
3	cdr3_mean_volume	0.587	Structural - Volume	CDR3
4	cdr3_eigvec_l1_norm_below_threshold	0.576	MD - Eigenvector	CDR3
5	fw_eigvec_skew_below_threshold	0.520	MD - Eigenvector	Framework
6	fw_eigvec_peak_count	0.486	MD - Eigenvector	Framework
7	cdr3_df_kurtosis	0.484	MD - Average DF	CDR3
8	cdr3_rmsf_kurtosis	0.470	MD - RMSF	CDR3
9	cdr3_class_freq_aromatic	0.467	Structural - Functional Classes	CDR3
10	cdr3_structural_entropy	0.467	Structural - Information Theory	CDR3

La Tabella 5.8 mostra una notevole stabilità della natura fisico-chimica discriminante precedentemente identificata (Tabella 5.5). Le prime due posizioni sono occupate dalle stesse *feature* del *dataset* base: *skewness* e *kurtosis*, che descrivono la distribuzione del primo autovettore del CDR3, e valida il potenziale ruolo primario dei descrittori di natura energetica nella discriminazione dei meccanismi di legame.

Tabella 5.9: Distribuzione delle *top*-10 *feature* del *consensus score* (*dataset extended*) raggruppate per categoria fisica e regione.

Physical Category	Feature Count	Percentage
MD - Eigenvector	5	50.0%
MD - RMSF	1	10.0%
Structural - Functional Classes	1	10.0%
Structural - Steric	1	10.0%
Structural - Information Theory	1	10.0%
MD - Average DF	1	10.0%
Macro-category totals (summed across subcategories)		
MD (overall)	7	70.0%
Structural (overall)	3	30.0%
Overall region distribution		
CDR3	8	80.0%
Framework	2	20.0%

La regione CDR3 conferma la sua predominanza, rappresentando 8 delle 10 *feature* più importanti secondo i modelli, mentre il *framework* contribuisce con 2 *features*. Rispetto ai risultati del *dataset* base (Tabella 5.3), questo evidenzia uno squilibrio ancora più marcato (nel caso base, le *feature* del CDR3 erano il 60% delle *top*-10, contro l'80% nel caso *extended*), validando ulteriormente, anche su un *dataset* più variabile, l'importanza di questa regione per l'identificazione del meccanismo di riconoscimento.

L'analisi aggregata per categorie fisiche (Tabella 5.9) rivela un ulteriore andamento coerente rispetto ai risultati del *dataset* base. Le *features* di dinamica molecolare (MD *overall*) costituiscono il 70% del totale, di cui le feature derivanti dal metodo REBELOT da sole contribuiscono per il 50% (contro il 60% del *dataset* base). Le proprietà strutturali mostrano un incremento relativo: 30%, contro il 20% nel caso base.

Il confronto tra i due *dataset* evidenzia quindi una importante stabilità della natura fisico-chimica discriminante.

Le *feature* di dinamica molecolare (MD) mantengono la loro predominanza, sebbene con valori differenti (80% nel base contro 70% nell'*extended*). Parallelamente, emerge una maggiore rilevanza delle proprietà strutturali, come evidenziato dall'ingresso nel *rank* di *feature* quali la frequenza di amminoacidi aromatici nel CDR3 e l'entropia sequenziale, che prima non erano presenti ma emergono con un campionario più ampio di proteine.

Tabella 5.10: MLP *Deep Architecture* — Top-10 *feature* per importanza normalizzata (0–1) e distribuzione per categoria fisica nei 5 *split* di CV.

Rank	Feature	Normalized Importance
1	cdr3_eigvec_skew	1.000
2	cdr3_eigvec_kurtosis	0.723
3	cdr3_df_kurtosis	0.636
4	cdr3_entropy	0.622
5	fw_eigvec_skew_below_threshold	0.619
6	cdr3_class_freq_aromatic	0.569
7	cdr3_rmsf_kurtosis	0.562
8	cdr3_mean_volume	0.544
9	fw_mean_hydrophobicity	0.503
10	fw_rmsf_skew	0.497

-	Feature Count	Percentage
Macro-category totals		
MD (overall)	6	60.0%
Structural (overall)	4	40.0%
Region distribution		
CDR3	7	70.0%
Framework	3	30.0%

A completamento dell'analisi di *consensus*, come ulteriore convalida, sono riportate in Tabella 5.10 le *features* più rilevanti secondo l'architettura con le migliori *performance* predittive sul *dataset extended* (MLP *Deep Architecture*). I risultati, confermano e rafforzano le evidenze emerse nel caso base (Tabella 5.4), validando come il modello con le prestazioni migliori, considera l'informazione di natura energetica come quella maggiormente discriminante.

5.2.6 Analisi statistica

Al fine di convalidare la robustezza delle evidenze emerse dall'analisi di *feature importance*, è stato applicato il medesimo protocollo statistico del *dataset* base, con un'importante modifica nella soglia di significatività. Come descritto nella Sezione 4.3.11, data l'aumentata potenza statistica derivante dal maggior numero di campioni disponibili la soglia di significatività è stata ristretta a $\alpha = 0.001$, riducendo la tolleranza per falsi positivi rispetto alla soglia più conservativa $\alpha = 0.005$ adottata nel *dataset* base. Questo approccio più stringente garantisce una validazione ancor più rigorosa delle differenze osservate.

I risultati sono riportati in Tabella 5.11.

Tabella 5.11: Esito dei test inferenziali sulle *top-10 feature* del *consensus* sul *dataset extended*. Per ciascuna *feature* viene indicato il test selezionato dall'algoritmo decisionale e l'esito del *test* rispetto alla soglia $\alpha = 0.001$.

Rank	Feature	Test selezionato	Esito ($p_{\text{final}} < 0.001$)
1	**cdr3_eigvec_skew**	***t*-test**	**Sì**
2	**cdr3_eigvec_kurtosis**	**Wilcoxon**	**Sì**
3	cdr3_eigvec_l1_norm_below_threshold	*t*-test	No
4	cdr3_mean_volume	*t*-test	No
5	fw_eigvec_skew_below_threshold	*t*-test	No
6	fw_eigvec_peak_count	Wilcoxon	No
7	fw_rmsf_skew	*t*-test	No
8	cdr3_class_freq_aromatic	*t*-test	No
9	cdr3_structural_entropy	Wilcoxon	No
10	cdr3_df_kurtosis	*t*-test	No

I risultati dimostrano che, nonostante l'applicazione di una soglia di significatività statistica più stringente (α ridotto da 0.005 a 0.001), le due feature più importanti per *consensus score*, ovvero `cdr3_eigvec_skew` e `cdr3_eigvec_kurtosis`, mantengono la loro significatività statistica, analogamente al caso base (Tabella 5.5). Questa persistenza, consente di affermare con ragionevole certezza il ruolo biologico fondamentale di tali descrittori, e conferma in modo robusto il ruolo primario delle proprietà energetiche degli amminoacidi del CDR3 nella discriminazione dei meccanismi di riconoscimento.

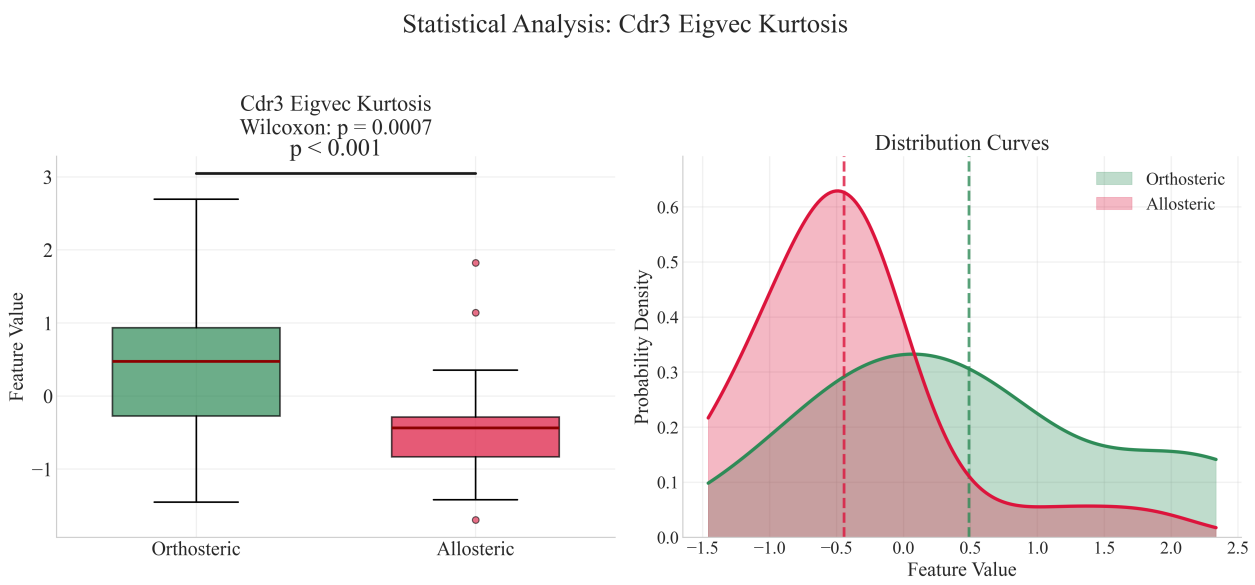

Figura 5.12: Distribuzione della `cdr3_eigvec_kurtosis` nel dataset *extended*.

Per la `cdr3_eigvec_kurtosis` (Figura 5.12), analogamente al caso base, è stato eseguito il *test* di Wilcoxon, ottenendo un *p-value* = 0.0007. Anche in questo caso, come nel *dataset* base, i *nanobodies* ortosterici, rivelano un valor medio di curtosi positivo, indicando una distribuzione ipoteticamente più leptocurtica, mentre i *nanobodies* allosterici presentano una curtosi media negativa, corrispondente a una distribuzione ipoteticamente platicurtica, validando i risultati sulla ipotetica forma della distribuzione energetica sulla CDR3.

In modo analogo, la `cdr3_eigvec_skew` (Figura 5.13) è stata testata con un *t-test*, ottenendo $p = 0.0004$. La classe ortosterica mostra un valor medio di *skew* più elevato (circa doppio), indicando una coda destra

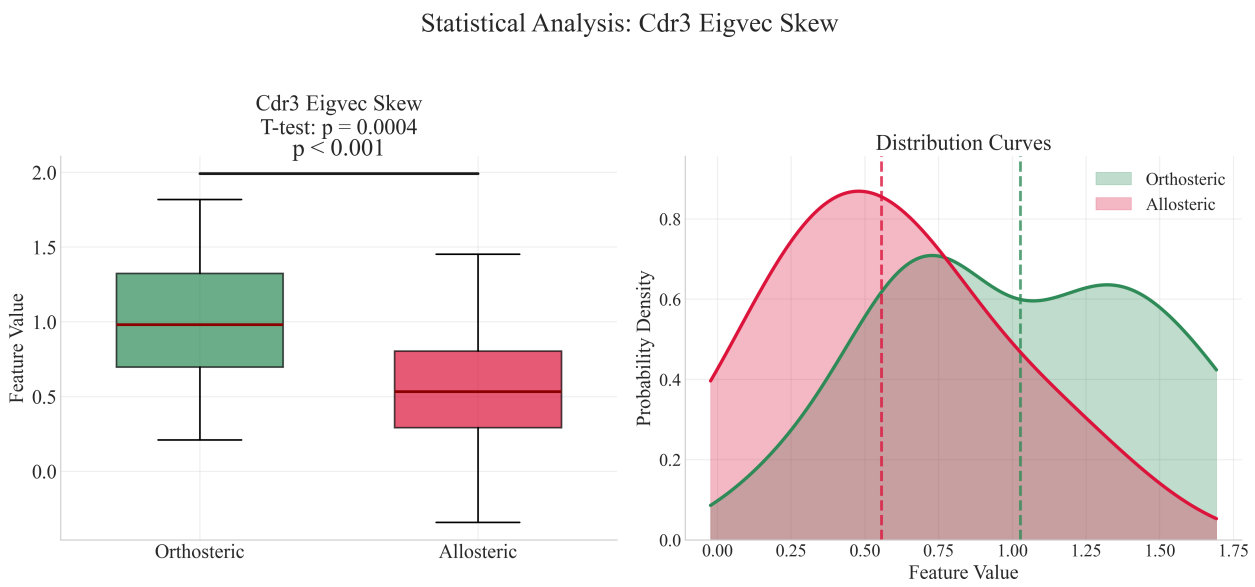

Figura 5.13: Distribuzione della `cdr3_eigvec_skew` nel dataset *extended*.

più marcata nella distribuzione dei contributi energetici, mentre i *nanobodies* allosterici confermano una distribuzione maggiormente *gaussian-like* e meno asimmetrica.

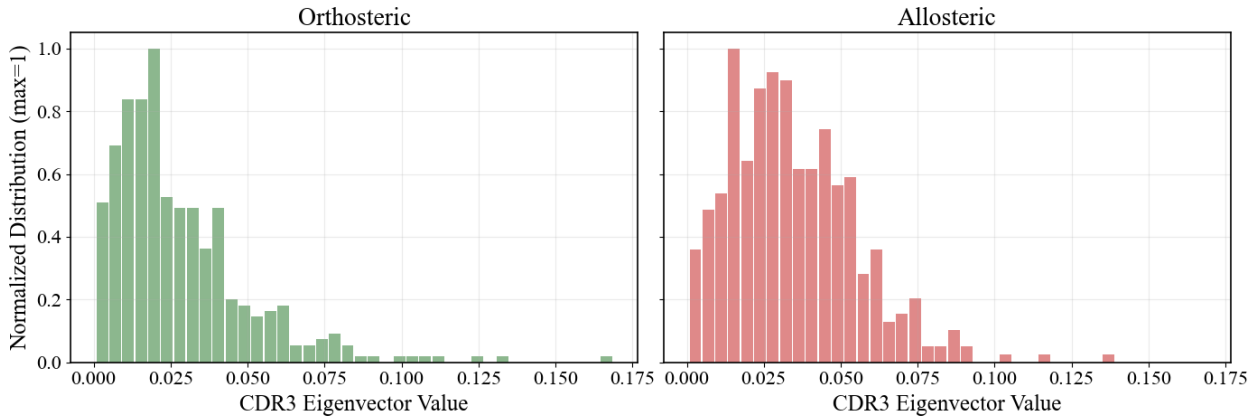

Figura 5.14: Distribuzione normalizzata del contributo al primo autovettore nel *dataset extended*.

La Figura 5.14 conferma in modo inequivocabile la riproducibilità dei *pattern* distributivi osservati nel *dataset* base. La distribuzione normalizzata dei valori del primo autovettore per i residui CDR3 mostra le stesse caratteristiche distintive tra i meccanismi ortosterici e allosterici riportate nella Figura 5.5, validando così le interpretazioni fisiche basate sulle statistiche di *skewness* e *kurtosis*.

6 Conclusioni e Prospettive Future

Nel presente lavoro, è stato proposto un *framework* che integra simulazioni di dinamica molecolare di *nanobodies* in condizioni *target-free* con modelli di *machine learning* e *deep learning* per predire il legame dei *nanobodies* sull'epitopo.

A partire dallo studio proposto da Xiang et al. [17], è stata effettuata una sotto-*clusterizzazione* ulteriore sulla base del solo meccanismo di legame (ortosterico e allosterico), costituito unicamente dai *nanobodies* di classe I e II, riconducendo lo studio ad un problema di classificazione binario. A partire da questo presupposto, sono state selezionata dal PDB un *pool* di biomolecole al fine di costruire una coorte bilanciata, costituita da 25 proteine con riconoscimento ortosterico e 25 proteine allosterico.

Nella prima parte del lavoro, corrispondente anche al primo obiettivo della ricerca, per ogni *nanobody* selezionato sono state eseguite simulazioni di MD seguendo un ben delineato protocollo, producendo un totale di 4 repliche per biomolecola, ognuna di esse dalla durata di $1\mu s$. L'impiego di repliche inizializzate da condizioni iniziali diverse aumenta la copertura dello spazio conformazionale, e consente di accrescere la robustezza statistica delle inferenze. Lo scopo di eseguire questi esperimenti *in silico* è simulare il comportamento del *nanobody* in condizioni il più fedeli possibile a quelle fisiologiche in soluzione. In questa prospettiva, la dinamica molecolare integra e va oltre l'informazione cristallografica del PDB, per sua natura statica e talvolta condizionata dalle condizioni sperimentali, permettendo di esplorare lo spazio conformazionale accessibile in solvente e di osservare pose rilevanti per il riconoscimento dell'epitopo. Sulla base delle traiettorie prodotte, sono stati estratti descrittori per-residuo, che caratterizzano il comportamento di ogni amminoacido della proteina, e di *time-series*, per descrivere l'evoluzione temporale di specifiche tendenze della biomolecola e della plasticità conformazionale dei domimi ipervariabili. A corredo di questa analisi è stata integrata anche la predizione di quali residui costituiscono i *loop* CDR (non distinti nelle *depository* del PDB), al fine di annotare i residui coinvolti nel riconoscimento antigenico. L'obiettivo di questa prima fase è stato quello di ottenere delle quantità che descrivessero la biomolecola sotto il duplice profilo strutturale e dinamico, per prevedere la polireattività e la differente modalità di *binding*.

Successivamente, è stata sviluppata una *pipeline* di apprendimento supervisionato. La coorte di 50 *nanobodies* selezionati è stata suddivisa per poter generare due *dataset*, con lo scopo di ottimizzare le *performance* dei classificatori: il primo, denominato *dataset* base, esteso su 40 strutture mantenendo sempre una proporzione bilanciata fra le classi, e poi successivamente sul *dataset extended*, esteso sull'intera popolazione di proteine, per validare la stabilità e generalizzabilità dei risultati. Grazie ai descrittori di *time series*, è stata implementata anche una strategia di *data-augmentation*, che permette di ampliare il numero di campioni, catturando ogni proteina in uno stato conformazionale differente (*sampling via-snapshot*). I risultati mostrano come in entrambi i *dataset*, i modelli con le prestazioni migliori, e che al contempo garantiscono una maggiore generalizzabilità, sono le *neural network fully-connected*, in particolare gli *MLP classifier*.

Inoltre, emerge come la strategia adottata mostra prestazioni di classificazione superiori se comparata a metodi computazionali classici di predizione del *binding* come il *docking* con il *software* commerciale MAESTRO, e ottiene risultati leggermente inferiori a un metodo *state-of-the-art* come AF3, pur usando un approccio differente ad esso. Infatti, il metodo sviluppato utilizza informazioni strutturali e dinamiche e un approccio *target-free*. Ciò valida la difficoltà del problema di classificazione affrontato, nonchè il successo della metodologia *physics-based* proposta.

In aggiunta, gli obiettivi ultimi del lavoro non si sono limitati al solo al *task* di classificazione, ma attraverso quest'ultimo, nella parte di *features importance analysis*, corrispondente alla seconda parte del lavoro, è stato possibile indagare attraverso un aproccio *data-driven physics-informed* i determinanti che hanno permesso la discriminazione dei due meccanismi di riconoscimento. Le evidenze emergenti risultano promettenti in quanto confermano la letteratura scientifica, evidenziando che i descrittori a prevalente natura energetica che discriminano l'interazione epitopo–paratopo si concentrano principalmente sui residui del *loop* CDR3

esteso [62]. Questa analisi mette in luce il differente profilo energetico tra i *nanobodies* appartenenti a classi diverse, altrimenti non evidenziabile usando differenti approcci computazionali. La strategia di apprendimento supervisionato sviluppata, unita all'analisi dell'importanza delle *features* usata in questo lavoro, permette di identificare *pattern* in dati unicamente ottenibili tramite approcci *physics-based in silico*, dai quali estrarre informazioni importanti per la caratterizzazione del comportamento di biomolecole intrinsecamente disordinate come i *nanobodies*.

In definitiva, è possibile concludere che la ricerca ha permesso di implementare modelli di *deep learning* robusti anche a fronte di un aumento della variabilità introdotta da nuove biomolecole nel *dataset*, con prestazioni superiori o comparabili ad altri approcci computazionali. Inoltre, quest'ultimo ha permesso di migliorare la comprensione dei meccanismi di riconoscimento dei *nanobodies* a livello molecolare. Contestualmente, la scelta di adottare una strategia *target-free*, con l'obiettivo di indagare in maniera più generalizzabile possibile il meccanismo di riconoscimento fra antigene e anticorpo, apre le strade a nuove prospettive applicative di queste proteine, che esulano dal nativo contesto virologico.

In questa prospettiva, l'integrazione delle predizioni *in silico* fornite dal modello *physics-informed* e da AF3 configura di fatto una procedura di *virtual screening*: le informazioni complementari sui probabili epitopi di legame e sull'orientazione tridimensionale del complesso possono essere sfruttate per guidare il *docking* e il *design in silico*, restringendo progressivamente lo spazio delle biomolecole candidate. In particolare, la convergenza di evidenze indipendenti a favore dello stesso epitopo consente di aumentare la priorità assegnata a specifici *nanobodies*, concentrando le successive analisi *in vivo* proprio sulle molecole più stabili e razionalmente più promettenti.

Un primo ambito di sviluppo riguarda l'ottimizzazione dei classificatori, ricercando nuovi descrittori in grado di catturare informazioni più fini di natura chimica che sono in grado di discriminare meglio i meccanismi di legame.

Un secondo miglioramento di rilievo consisterebbe nell'introduzione di una metodologia di *imputing*, capace di attribuire valori ai descrittori dinamici fornendo al modello la sola sequenza amminoacidica del *nanobody*. Tale strategia sarebbe *time-saving*, poiché elimina sia la fase preparatoria sia la produzione delle simulazioni di MD, attività peraltro onerose in termini di infrastrutture di calcolo; al contempo, rappresenterebbe un progresso significativo sul piano della fruibilità (*user-friendly*). In termini operativi, una volta prodotta la proteina, sarebbe possibile, a valle del solo sequenziamento e senza ricorrere alla cristallografia, caratterizzare il meccanismo di riconoscimento del *nanobody* direttamente dalla sequenza. Ciò favorirebbe la progettazione di nuovi composti basati su *nanobodies*, ampliando ulteriormente le potenzialità diagnostiche e terapeutiche grazie alla previsione delle proprietà di legame proteina–proteina, e indirizzando in modo mirato lo sviluppo di terapie e farmaci.

Appendice

Modelli di *Deep Learning* (PyTorch)

Se non esplicitato diversamente, queste architetture richiedono la conversione preliminare degli array `NumPy` in tensori `float32` PyTorch [77].

Ove possibile, tutti i modelli sono stati addestrati integrando la parallelizzazione multicore tramite il parametro `n_jobs=-1`.

- *Neural Network*: *Multi-Layer Perceptron* (MLP) costituito da strati *fully-connected* [78]. Sono state implementate tre diverse architetture, che differiscono per profondità (numero di *layer*) e ampiezza (numero di neuroni per ogni *layer*):
  - MLP *Light*: Architettura baseline con 2 strati nascosti [64, 32], *dropout rate* del 20% e *learning rate* di 0.001.
    - Proiezioni lineari: $167 \to 64 \to 32 \to 1$.
  - MLP *Wide*: Architettura con estesa capacità rappresentativa al primo *layer* (3 strati: [512, 256, 128]), *dropout rate* del 40% e *learning rate* di 0.0005.
    - Proiezioni lineari: $167 \to 512 \to 256 \to 128 \to 1$.
  - MLP *Deep*: Architettura di profonda moderata con 4 strati nascosti [256, 128, 64, 32], *dropout rate* del 30% e *learning rate* di 0.001.
    - Proiezioni lineari: $167 \to 256 \to 128 \to 64 \to 32 \to 1$.

    Tutte le architetture condividono la seguente configurazione:
    - *Training Configuration*:
      - *Batch Size*: $N = 32$ campioni;
      - Epoche massime: 200;
      - *Validation Split*: 20% del training set.
    - *Batch Normalization*: Applicata dopo ogni *layer* lineare per stabilizzare il *training*;
    - *Activation Function*: Funzione di attivazione non lineare $\text{ReLU}(x) = \max(0, x)$ negli *hidden layers*;
    - *Output*: 1 neurone che produce un *logit* reale $z \in (-\infty, +\infty)$ in uscita (*output* dello strato lineare finale prima di qualsiasi attivazione).
    - *Probability Conversion*: per le predizioni finali, si applica una sigmoide ai *logits*:

      $$P(class = 1) = \sigma(z) = \frac{1}{1 + e^{-z}} \tag{6.1}$$

      La *sigmoid* quindi mappa il *logit* z in una probabilità $\in [0, 1]$;
    - *Loss function*: *Binary Cross Entropy (BCE)*, applicata dopo l'attivazione *sigmoid*:

      $$\mathscr{L}_{BCE} = -\frac{1}{N} \sum_{i=1}^{N} [y_i \cdot \log(p_i) + (1 - y_i) \cdot \log(1 - p_i)] \tag{6.2}$$

      dove:
      - $p_i = \sigma(z_i)$: probabilità predetta dalla *sigmoid*;
      - $y_i \in \{0, 1\}$: *label* di classe (0 = ortosterico, 1 = allosterico);
      - N: *batch size*.
    - *Optimizer*: Adam [109].
  - *Deep ResNet*: basata sui principi di ResNet [81], ma specificamente adattata per l'elaborazione di

dati tabellari. L'innovazione rispetto ad una *Neural Network standard* risiede nell'implementazione di *blocchi residui* che utilizzano *skip connections*.

* *Input projection*: 167 → 256 neuroni;
* BatchNormalization, ReLU Activation, Dropout: 30%;
* *Residual Block*: Due *layer* lineari con *BatchNorm* e *skip connection*;
* *Classifier*: MLP con 2 *hidden later* 256 → 128 → 2 neuroni;
* *Output*: 2 neuroni (*logits* per classi ortosterico/allosterico);
* *Conversione a Probabilità*: *Softmax*

$$P(y_i = k) = \frac{\exp(z_{i,k})}{\sum_{j=0}^{1} \exp(z_{i,j})} \quad \text{per } k \in \{0,1\} \tag{6.3}$$

* *Loss function*: *CrossEntropyLoss* + *Softmax*

$$\mathscr{L}_{CE} = -\frac{1}{N} \sum_{i=1}^{N} \log \left(\frac{\exp\left(z_{i,y_i}\right)}{\sum_{j=0}^{1} \exp(z_{i,j})} \right) \tag{6.4}$$

* *Decisione*: *Argmax* delle probabilità *softmax*

$$\text{Classe predetta} = \arg \max_{k \in \{0,1\}} P(class = k) \tag{6.5}$$

* Epoche massime: 300;
* *Batch Size*: 64;
* *Learning Rate*: 0.001 con *weight decay* 1e-5;
* *Optimizer*: Adam con regolarizzazione L2;
* *Validation Split*: 15% del *training set*.

– *Attention Network*: Architettura neurale con meccanismo di *self-attention feature-wise* [82] che apprende pesi di rilevanza specifici per ogni campione e *feature*. Composta da due componenti principali: un modulo di *attention* e un *classifier* MLP.

* Meccanismo di *Attention*:
  · Linear Projection: 167 → 256 con attivazione tanh;
  · Linear Projection: 256 → 167 con attivazione *softmax*;
  · Maschera dei pesi: $\alpha = \text{softmax}(\tanh(X \cdot W_1 + b_1) \cdot W_2 + b_2)$;
  · Feature pesate: $X_\alpha = X \odot \alpha$ (prodotto di Hadamard).
* *Classifier* MLP:
  · *Linear Layer* 1: 167 → 256 con *BatchNorm*, *ReLU*, *Dropout* (30%);
  · *Linear Layer* 2: 256 → 128 con *BatchNorm*, *ReLU*, *Dropout* (30%);
  · *Output Layer*: 128 → 2 neuroni;
  · *Output*: 2 neuroni (logits per classificazione binaria);
  · *Loss function*: CrossEntropyLoss.
* *Training Protocol*:
  · Epoche massime: 200;
  · *Batch Size*: 64;
  · *Learning Rate*: 0.001 con *weight decay* 1e-5;
  · *Optimizer*: Adam con regolarizzazione L2;
  · *Validation Split*: 15% del *training set*.

Modelli di *Machine Learning* (scikit-learn)

- *Logistic Regression*: Classificatore lineare probabilistico
  - `solver='liblinear'`;
  - `C=1.0`: Forza della regularizzazione ($C = 1/\lambda$);
  - `max_iter=2000`: Iterazioni massime per convergenza;
  - *Loss*: *log-loss* (cross-entropy) con regolarizzazione L2.
- *K-Nearest Neighbors*: Classificazione basata su similarità locale
  - `n_neighbors=5`: Numero di vicini considerati;
  - `weights='distance'`: Pesi inversamente proporzionali alla distanza;
  - `metric='minkowski'`: Distanza euclidea generalizzata (p=2).
- *Random Forest*
  - `n_estimators=300`: Numero di alberi nell'*ensemble*;
  - `max_depth=6`: Profondità massima.
- *Gradient Boosting*
  - `n_estimators=300`: Stadi di *boosting*;
  - `learning_rate=0.1`;
  - `max_depth=6`: Profondità massima *weak learners*.
- *XGBoost* (v3.0.4)
  - `n_estimators=400`: Stadi *boosting* aumentati;
  - `max_depth=6`: Profondità alberi;
  - `learning_rate=0.1`;
  - `reg_alpha=0.1`: Regularizzazione L1 (Lasso);
  - `reg_lambda=1.0`: Regularizzazione L2 (Ridge);
  - `eval_metric='mlogloss'`: Metrica di valutazione.
- *SVM*:
  - `kernel='rbf'`: *Kernel* radiale per non linearità;
  - `C=10`: Parametro di regularizzazione (bassa penalizzazione);
  - `gamma='scale'`: Scala automatica del *kernel*;
  - `class_weight='balanced'`.
- *Deep Forest*: Approccio deep learning non neurale con cascade forest
  - `n_estimators=200`: Alberi per ogni foresta;
  - `n_forests=4`: Numero di foreste per *layer*;
  - `max_layers=4`: Profondità massima del *cascade*.
- *Voting Classifier*: Combinazione eterogenea di modelli. Strategia *Wisdom of crowds*
  - *Soft voting* (media delle probabilità);
  - Modelli: *Random Forest, Gradient Boosting, Extra Trees, SVM, KNN, Logistic Regression*.

Risultati *Dataset Base*

Tabella 6.1: *Performance* comparative dei classificatori sul *dataset* base. I valori sono calcolati su 5-*fold* CV e sono espressi come media ± deviazione *standard* (i valori sono espressi in percentuale, eccetto MCC).

Modello	ACC (%)	TPR (%)	TNR (%)	F1 (%)	PPV (%)	MCC
MLP - Light Architecture (PyTorch)	73.7±6.4	70.0±18.7	77.5±17.0	71.8±8.9	78.9±13.1	0.509±0.137
Attention Network (PyTorch)	70.0±10.0	75.0±15.8	65.0±12.2	70.9±10.6	68.3±8.0	0.410±0.205
MLP - Wide Architecture (PyTorch)	67.6±9.7	65.3±19.8	70.0±18.7	66.0±10.7	71.8±16.0	0.385±0.219
K-Nearest Neighbors (scikit-learn)	67.3±17.0	70.2±18.5	64.4±18.6	68.0±17.2	66.6±17.0	0.349±0.339
Deep ResNet (PyTorch)	67.0±8.2	65.1±19.8	68.9±21.5	65.5±10.3	69.7±10.8	0.372±0.163
MLP - Deep Architecture (PyTorch)	66.1±4.8	65.1±19.8	67.1±13.9	64.4±8.6	67.4±5.2	0.347±0.123
SVM (scikit-learn)	61.3±8.0	61.7±11.2	60.9±11.6	61.2±8.5	61.6±8.4	0.229±0.161
Voting Classifier (scikit-learn)	58.4±11.6	66.7±18.6	50.0±22.4	61.1±10.7	58.6±11.3	0.192±0.248
Logistic Regression (scikit-learn)	57.6±6.1	59.9±25.4	55.2±29.0	56.2±12.2	58.8±7.6	0.160±0.123
Deep Forest (scikit-learn)	57.1±12.2	58.9±11.0	55.3±19.1	56.7±12.4	57.3±12.7	0.144±0.249
Random Forest (scikit-learn)	56.6±15.9	54.2±18.5	59.0±22.7	55.4±15.6	59.6±20.0	0.147±0.328
Gradient Boosting (scikit-learn)	55.4±17.0	51.7±18.7	59.1±24.2	53.7±16.1	59.3±22.0	0.126±0.355
XGBoost (scikit-learn)	55.0±12.7	50.0±22.4	60.0±25.5	51.0±16.7	58.7±22.4	0.115±0.283

Risultati *Dataset Extended*

Tabella 6.2: *Performance* comparative dei classificatori sul *dataset extended*. I valori sono calcolati su 5-*fold* CV e sono espressi come media ± deviazione *standard* (i valori sono espressi in percentuale, eccetto MCC).

Modello	ACC (%)	TPR (%)	TNR (%)	F1 (%)	PPV (%)	MCC
MLP - Deep Architecture (PyTorch)	69.7±6.2	67.8±16.0	71.6±20.5	68.4±7.4	75.5±14.5	0.422±0.125
MLP - Wide Architecture (PyTorch)	69.3±8.1	66.3±15.0	72.3±24.8	68.0±7.7	77.1±17.4	0.418±0.167
K-Nearest Neighbors (scikit-learn)	67.4±18.7	73.0±21.2	61.8±20.6	68.8±19.2	65.8±18.9	0.355±0.374
SVM (scikit-learn)	65.7±14.5	59.9±12.6	71.4±26.6	64.0±12.5	74.3±21.6	0.337±0.302
MLP - Light Architecture (PyTorch)	65.5±5.2	62.3±13.5	68.7±20.5	63.8±4.6	71.7±15.4	0.335±0.125
Logistic Regression (scikit-learn)	65.2±9.8	69.7±9.0	60.7±27.8	67.3±4.4	69.3±17.3	0.320±0.213
Deep ResNet (PyTorch)	63.8±7.8	60.0±17.9	67.6±19.7	61.4±10.0	68.7±15.4	0.300±0.170
Attention Network (PyTorch)	61.8±4.2	59.6±21.1	64.0±23.3	59.5±8.4	64.7±6.8	0.266±0.078
Deep Forest (scikit-learn)	59.3±8.1	60.3±12.1	58.2±16.1	58.7±8.2	59.9±8.4	0.191±0.165
Voting Classifier (scikit-learn)	59.0±13.7	69.1±14.1	49.0±17.9	62.8±12.0	58.2±12.1	0.186±0.278
Gradient Boosting (scikit-learn)	56.6±11.8	60.0±12.6	53.2±21.0	58.0±10.1	57.7±11.6	0.132±0.244
XGBoost (scikit-learn)	56.0±8.0	60.0±17.9	52.0±16.0	56.6±10.9	55.3±6.9	0.122±0.163
Random Forest (scikit-learn)	55.8±10.1	55.7±14.3	55.9±15.0	55.3±10.4	56.6±10.1	0.121±0.207

Tabella 6.3: Confronto tra le predizioni di *AlphaFold 3* (valutazione tramite DockQ e *refinement* geometrico), BIOLUMINATE e della *Neural Network physics-informed* per ciascun *nanobody*.

Nanobody ID	AF3 DockQ	BIOLUMINATE	*Neural Network*	AF3 *refinement*
Orthosteric				
7c8v_SR4	X	✓	✓	✓
6yz5_H11-D4	✓	✓	✓	✓
8cya_Nb2-67	✓	✓	✓	✓
7kgj_Sb45	X	X	✓	✓
7kgk_Sb16	X	✓	✓	✓
7mfu_Sb14	✓	✓	X	✓
7fau_Nb1B11	X	✓	✓	✓

Continua nella pagina successiva

Tabella 6.3: Confronto tra predizioni AF3, BIOLUMINATE e *Neural Network* (continua)

Nanobody ID	**AF3 DockQ**	**BIOLUMINATE**	***Neural Network***	**AF3 *refinement***
7oao_C5	X	✓	✓	✓
7olz_Re5D06	✓	✓	X	✓
7f5g_DL4	✓	✓	X	✓
8gz5_VHH-P17	X	✓	X	✓
7wd1_R14	✓	✓	✓	✓
8q7s_Ma6F06	✓	X	✓	✓
8q95_Ma16B06	X	✓	✓	X
8owv_H6	X	✓	X	✓
6zxn_Ty1	X	✓	✓	✓
7voa_aRBD5	X	✓	✓	✓
7z1c_B5	✓	✓	✓	✓
7tpr_8A2	X	✓	✓	✓
7w1s_Nb-007	✓	✓	✓	✓
7kn5_VHHE	✓	✓	✓	✓
7oap_H3	✓	✓	X	✓
7rby_Nb112	✓	✓	✓	✓
7x7e_Nb22	X	✓	X	✓
8q94_Re32D03	X	✓	✓	✓
Allosteric				
8cyc_Nb2-34	✓	X	✓	✓
8cy7_Nb2-38	✓	X	X	✓
8cyb_Nb1-8	X	X	✓	X
7klw_Sb68	X	X	X	X
7fat_Nb1A7	X	X	X	✓
7nkt_NM1226	✓	X	✓	✓
7oap_C1	X	X	X	✓
7oay_F2	X	X	✓	✓
7olz_Re9F06	✓	✓	✓	✓
7fbj_17F6	✓	✓	✓	✓
7fbk_20G6	✓	X	✓	✓
7x2j_Nb70	X	✓	✓	✓
7x2m_1-2C7	X	✓	✓	X
7wd2_S43	✓	X	X	✓
8hr2_Nb1B5	X	X	✓	✓
8q7s_Re21H01	X	X	✓	X
8q95_Ma3F05	X	✓	✓	X
8owt_A8	✓	X	X	✓
7tpr_7A3	X	X	✓	✓
8h5u_Nb-021	X	X	X	X
7kn6_VHHV	✓	X	✓	✓
7my2_Nb30	✓	X	X	✓
8elq_C4-255	X	X	✓	X

Continua nella pagina successiva

Tabella 6.3: Confronto tra predizioni AF3, BIOLUMINATE e *Neural Network* (continua)

Nanobody ID	**AF3 DockQ**	**BIOLUMINATE**	***Neural Network***	**AF3 *refinement***
8q93_Re21D01	X	X	✓	X
8q94_Ma3B12	✓	X	✓	✓

Bibliografia

[1] Asher Mullard. Parsing clinical success rates. *Nature Reviews Drug Discovery*, 15(7):447, 2016. doi: 10.1038/nrd.2016.136.

[2] Katarzyna Smietana, Marcin Siatkowski, and Martin Møller. Trends in clinical success rates. *Nature Reviews Drug Discovery*, 15(6):379–380, 2016. doi: 10.1038/nrd.2016.85.

[3] Mohit Pandey, Michael Fernandez, Francesco Gentile, Olexandr Isayev, Alexander Tropsha, Abraham C. Stern, and Artem Cherkasov. The transformational role of GPU computing and deep learning in drug discovery. *Nature Machine Intelligence*, 4(3):211–221, March 2022. doi: 10.1038/s42256-022-00463-x.

[4] Kenneth M. Murphy, Casey Weaver, and Leslie J. Berg. *Janeway's Immunobiology*. W. W. Norton & Company, New York, NY, 10 edition, 2022. ISBN 978-0393884890.

[5] Hélène Kaplon, Alicia Chenoweth, Silvia Crescioli, and Janice M. Reichert. Antibodies to watch in 2022. *mAbs*, 14(1):e2014296, 2022. doi: 10.1080/19420862.2021.2014296.

[6] Mark L. Chiu, Dennis R. Goulet, Alexey Teplyakov, and Gary L. Gilliland. Antibody structure and function: The basis for engineering therapeutics. *Antibodies*, 8(4):55, December 2019. ISSN 2073-4468. doi: 10.3390/antib8040055.

[7] Serge Muyldermans. Nanobodies: natural single-domain antibodies. *Annu. Rev. Biochem.*, 82 (1):775–797, March 2013.

[8] Peter Bannas, Julia Hambach, and Friedrich Koch-Nolte. Nanobodies and nanobody-based human heavy chain antibodies as antitumor therapeutics. *Frontiers in Immunology*, 8:1603, November 2017. ISSN 1664-3224. doi: 10.3389/fimmu.2017.01603.

[9] Els Pardon, Toon Laeremans, Sarah Triest, Søren G. F. Rasmussen, Alexandre Wohlkönig, Armin Ruf, Serge Muyldermans, Wim G. J. Hol, Brian K. Kobilka, and Jan Steyaert. A general protocol for the generation of Nanobodies for structural biology. *Nature Protocols*, 9(3):674–693, March 2014. doi: 10.1038/nprot.2014.039.

[10] Mehdi Arbabi-Ghahroudi. Camelid single-domain antibodies: Promises and challenges as lifesaving treatments. *International Journal of Molecular Sciences*, 23(9):5009, 2022. doi: 10.3390/ijms23095009.

[11] Janusz Wesolowski, Vanina Alzogaray, Jan Reyelt, Mandy Unger, Karla Juarez, Mariela Urrutia, Ana Cauerhff, Welbeck Danquah, Björn Rissiek, Felix Scheuplein, Nicole Schwarz, Sahil Adriouch, Olivier Boyer, Michel Seman, Alexei Licea, David V. Serreze, Fernando A. Goldbaum, Friedrich Haag, and Friedrich Koch-Nolte. Single domain antibodies: promising experimental and therapeutic tools in infection and immunity. *Medical Microbiology and Immunology*, 198(3): 157–174, August 2009. ISSN 0300-8584. doi: 10.1007/s00430-009-0116-7.

[12] A. Sircar, K. A. Sanni, J. Shi, and J. J. Gray. Analysis and modeling of the variable region of camelid single-domain antibodies. *Journal of Molecular Biology*, 407(1):193–208, 2011. doi: 10.1016/j.jmb.2011.01.044.

[13] Renhong Yan, Yuanyuan Zhang, Yaning Li, Lu Xia, Yingying Guo, and Qiang Zhou. Structural basis for the recognition of sars-cov-2 by full-length human ace2. *Science*, 367(6485):1444–1448, 2020. doi: 10.1126/science.abb2762.

[14] Yang Yang, Fang Li, and Lanying Du. Therapeutic nanobodies against sars-cov-2 and other pathogenic human coronaviruses. *Journal of Nanobiotechnology*, 22(1):304, 2024. doi: 10.1186/s12951-024-02573-7.

[15] Yongfei Cai, Jun Zhang, Tianshu Xiao, Hanqin Peng, Sarah M. Sterling, Richard M. Walsh, Shaun Rawson, Sophia Rits-Volloch, and Bing Chen. Distinct conformational states of sars-cov-2 spike protein. *Science*, 369(6511):1586–1592, 2020. doi: 10.1126/science.abd4251.

[16] Daniel Wrapp, Dorien De Vlieger, Kizzmekia S. Corbett, Gretel M. Torres, Nianshuang Wang, Wander Van Breedam, Kenny Roose, Loes van Schie, VIB-CMB COVID-19 Response Team, Markus Hoffmann, Stefan Pöhlmann, Barney S. Graham, Nico Callewaert, Bert Schepens, Xavier Saelens, and Jason S. McLellan. Structural basis for potent neutralization of betacoronaviruses by single-domain camelid antibodies. *Cell*, 181(5):1004–1015.e15, May 2020. ISSN 0092-8674. doi: 10.1016/j.cell.2020.04.031.

[17] Y. Xiang, N.-L. Tanasie, P. Gutierrez-Escribano, S. Jaklin, L. Aragon, J. Stigler, et al. Superimmunity by pan-sarbecovirus nanobodies. *Cell Reports*, 39(13):111004, 06 2022. doi: 10.1016/j.celrep.2022.111004.

[18] Michael Schoof, Bryan Faust, Reuben A. Saunders, Smriti Sangwan, Veronica Rezelj, Nick Hoppe, Morgane Boone, Christian B. Billesbølle, Cristina Puchades, Caleigh M. Azumaya, Huong T. Kratochvil, Marcell Zimanyi, Ishan Deshpande, Jiahao Liang, Sasha Dickinson, Henry C. Nguyen, Cynthia M. Chio, Gregory E. Merz, Michael C. Thompson, Devan Diwanji, Kaitlin Schaefer, Aditya A. Anand, Niv Dobzinski, Beth Shoshana Zha, Camille R. Simoneau, Kristoffer Leon, Kris M. White, Un Seng Chio, Meghna Gupta, Mingliang Jin, Fei Li, Yanxin Liu, Kaihua Zhang, David Bulkley, Ming Sun, Amber M. Smith, Alexandrea N. Rizo, Frank Moss, Axel F. Brilot, Sergei Pourmal, Raphael Trenker, Thomas Pospiech, Sayan Gupta, Benjamin Barsi-Rhyne, Vladislav Belyy, Andrew W. Barile-Hill, Silke Nock, Yuwei Liu, Nevan J. Krogan, Corie Y. Ralston, Danielle L. Swaney, Adolfo García-Sastre, Melanie Ott, Marco Vignuzzi, QCRG Structural Biology Consortium, Peter Walter, and Aashish Manglik. An ultrapotent synthetic nanobody neutralizes SARS-CoV-2 by stabilizing inactive Spike. *Science*, 370(6523): 1473–1479, December 2020. ISSN 0036-8075. doi: 10.1126/science.abe3255.

[19] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The protein data bank. *Nucleic Acids Research*, 28(1): 235–242, 2000. doi: 10.1093/nar/28.1.235.

[20] Marie Scully, Spero R. Cataland, Flora Peyvandi, Paul Coppo, Paul Knöbl, Johanna A. Kremer Hovinga, Ara Metjian, Javier de la Rubia, Katerina Pavenski, Filip Callewaert, Debendra Biswas, Hilde De Winter, and Robert K. Zeldin. Caplacizumab treatment for acquired thrombotic thrombocytopenic purpura. *New England Journal of Medicine*, 380(4):335–346, 2019. doi: 10.1056/NEJMoa1806311.

[21] B. S. Joly, P. Coppo, and A. Veyradier. Thrombotic thrombocytopenic purpura. *Blood*, 129(21): 2836–2846, 2017. doi: 10.1182/blood-2016-10-709857.

[22] Robert C. Sterner and Rosalie M. Sterner. Car-t cell therapy: current limitations and potential strategies. *Blood Cancer Journal*, 11(4):69, 2021. doi: 10.1038/s41408-021-00459-7.

[23] Fen Mo, S. Duan, X. Jiang, X. Yang, X. Hou, W. Shi, C. Carlos, X. Zhang, Y. Chen, H. Wang, C. Zu, P. Hu, J. Ren, D. Wang, C. Ma, T. Zhao, Y. Wang, Y. Ma, X. Liu, Y. Liu, Z. Ma, Y. Li, X. Wang, H. Li, M. Ponnusamy, J. Zhang, and P. Wang. Nanobody-based chimeric antigen

receptor t cells designed by crispr/cas9 technology for solid tumor immunotherapy. *Signal Transduction and Targeted Therapy, Nature*, 6(1):80, 2021. doi: 10.1038/s41392-021-00462-1.

[24] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630:493–500, May 2024. doi: 10.1038/s41586-024-07487-w. URL `https://www.nature.com/articles/s41586-024-07487-w`.

[25] Alphafold server faq. `https://alphafoldserver.com/faq`.

[26] Sankar Basu and Björn Wallner. DockQ: A quality measure for protein-protein docking models. *PLOS ONE*, 11(8):e0161879, 2016. doi: 10.1371/journal.pone.0161879.

[27] Bioluminate. Web page, 2025. URL `https://www.schrodinger.com/platform/products/bioluminate/`.

[28] Glen-Youl Chuang, Dima Kozakov, Ryan Brenke, Stephen R. Comeau, and Sandor Vajda. Dars (decoys as the reference state) potentials for protein–protein docking. *Biophysical Journal*, 95 (9):4217–4227, 2008. doi: 10.1529/biophysj.108.135814.

[29] *BioLuminate: Protein–Protein Docking*. Schrödinger, LLC, New York, NY, 2025. URL `https://learn.schrodinger.com/private/edu/release/current/Documentation/html/bioluminate/bioluminate_help/protein_protein_docking.html`.

[30] Prime mm-gbsa: Method overview and usage, 2025.

[31] Samuel Genheden and Ulf Ryde. The mm/pbsa and mm/gbsa methods to estimate ligand-binding affinities. *Expert Opinion on Drug Discovery*, 10(5):449–461, 2015. doi: 10.1517/17460441.2015.1032936.

[32] Raghuvir R. S. Pissurlenkar, Mushtaque S. Shaikh, Radhakrishnan P. Iyer, and Evans C. Coutinho. Molecular mechanics force fields and their applications in drug design. *Anti-Infective Agents in Medicinal Chemistry*, 8(2):128–150, 2009.

[33] Stewart A. Adcock and J. Andrew McCammon. Molecular dynamics: Survey of methods for simulating the activity of proteins. *Chemical Reviews*, 106(5):1589–1615, 2006. doi: 10.1021/cr040426m.

[34] J. W. Ponder and D. A. Case. Force fields for protein simulations. *Adv. Protein Chem.*, 66:27–85, 2003. doi: 10.1016/S0065-3233(03)66002-X.

[35] Michael P. Allen and Dominic J. Tildesley. *Computer Simulation of Liquids*. Oxford University Press, Oxford, UK, 2 edition, 2017. ISBN 9780198803201.

[36] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, volume 31 of *Springer Series in Computational Mathematics*. Springer, Berlin, Heidelberg, 2 edition, 2006. doi: 10.1007/3-540-30666-8.

[37] Loup Verlet. Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Physical Review*, 159(1):98–103, 1967. doi: 10.1103/PhysRev.159.98.

[38] Mark E. Tuckerman. *Statistical Mechanics: Theory and Molecular Simulation*. Oxford University Press, Oxford, UK, 2010. ISBN 9780198525264.

[39] Donald A. McQuarrie. *Statistical Mechanics*. University Science Books, Sausalito, CA, 1 edition, 2000. ISBN 978-1891389153.

[40] David Chandler. *Introduction to Modern Statistical Mechanics*. Oxford University Press, New York, NY, 1 edition, 1987. ISBN 978-0195042771.

[41] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak. Molecular dynamics with coupling to an external bath. *J. Chem. Phys.*, 81:3684–3690, 1984. doi: 10.1063/1.448118.

[42] D. A. Case et al. *AmberTools24 Reference Manual*. Amber Developers, 2024. URL `https://ambermd.org/doc12/Amber24.pdf`.

[43] Intel® xeon® gold 5318y processor. `https://www.intel.com/content/www/us/en/products/sku/215271/intel-xeon-gold-5318y-processor-36m-cache-2-10-ghz/specifications.html`, 2021.

[44] Nvidia a100 tensor core gpu. `https://www.nvidia.com/it-it/data-center/a100/`, 2020.

[45] NVIDIA CUDA Toolkit. `https://developer.nvidia.com/cuda-toolkit`, 2023. Versione 12.2, consultato il: [DATA].

[46] Els Pardon et al. A general protocol for the generation of nanobodies for structural biology. *Nature Protocols*, 9(3):674–693, 2014. doi: 10.1038/nprot.2014.039.

[47] Yifan Cheng. Single-particle cryo-em at crystallographic resolution. *Cell*, 161(3):450–457, 2015. doi: 10.1016/j.cell.2015.03.049.

[48] LLC Schrödinger. The PyMOL molecular graphics system, version 1.8. URL `https://www.schrodinger.com`.

[49] Chresten R. Søndergaard, Mats H. M. Olsson, Michał Rostkowski, and Jan H. Jensen. Improved treatment of ligands and coupling effects in empirical calculation and rationalization of pKa values. *Journal of Chemical Theory and Computation*, 7(7):2284–2295, 2011. doi: 10.1021/ct200133y.

[50] James A. Maier, Carmenza Martinez, Koushik Kasavajhala, Lauren Wickstrom, Kevin E. Hauser, and Carlos Simmerling. ff14sb: Improving the accuracy of protein side chain and backbone parameters from ff99sb. *Journal of Chemical Theory and Computation*, 11(8):3696–3713, 2015. doi: 10.1021/acs.jctc.5b00255.

[51] William L. Jorgensen, Jayaraman Chandrasekhar, Jeffry D. Madura, Roger W. Impey, and Michael L. Klein. Comparison of simple potential functions for simulating liquid water. *The Journal of Chemical Physics*, 79(2):926–935, 1983. doi: 10.1063/1.445869.

[52] J.-P. Ryckaert, G. Ciccotti, and H. J. C. Berendsen. Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. *J. Comput. Phys.*, 23:327–341, 1977. doi: 10.1016/0021-9991(77)90098-5.

[53] Daniel R. Roe and Thomas E. Cheatham. Ptraj and CPPTRAJ: Software for processing and analysis of molecular dynamics trajectory data. *Journal of Chemical Theory and Computation*, 9 (7):3084–3095, July 2013. ISSN 1549-9618. doi: 10.1021/ct400341p.

[54] ColomboLab. Distance-fluctuation-df-analysis: `distance_fluctuation.py` — distance fluctuation (df) analysis from md trajectories. `https://github.com/colombolab/Distance-Fluctuation-DF-Analysis`, 2024. URL `https://github.com/colombolab/Distance-Fluctuation-DF-Analysis`. GitHub repository, release v1.0.0 (commit b4b0324).

[55] Giulia Morra, Raffaello Potestio, Cristian Micheletti, and Giorgio Colombo. Corresponding functional dynamics across the hsp90 chaperone family: Insights from a multiscale analysis of md simulations. *PLoS Computational Biology*, 8(3):e1002433, 2012.

[56] Matteo Castelli, Andrea Magni, Giorgio Bonollo, Silvia Pavoni, Francesco Frigerio, A. Sofia F. Oliveira, Fabrizio Cinquini, Stefano A. Serapian, and Giorgio Colombo. Molecular mechanisms of chaperone-directed protein folding: Insights from atomistic simulations. *Protein Science*, 33: e4880, 2024. doi: 10.1002/pro.4880.

[57] F. Chiappori, I. Merelli, G. Colombo, L. Milanesi, and G. Morra. An atomistic view of hsp70 allosteric crosstalk: from the nucleotide to the substrate binding domain and back. *Scientific Reports*, 6:23474, 2016. doi: 10.1038/srep23474.

[58] Riccardo Capelli, Giulia Morra, and Giorgio Colombo. Rebelot: Residue-based eigenvector decomposition of interaction energy matrices. Metodo impiegato nel workflow MLCE/REBELOT, 2017. URL `https://github.com/colombolab/MLCE`.

[59] Claudio Peri, Paola Gagni, Fabio Combi, Alessandro Gori, Marcella Chiari, Renato Longhi, Marina Cretich, et al. Rational epitope design for protein targeting. *ACS Chemical Biology*, 8(2): 397–404, 2013. doi: 10.1021/cb300487u.

[60] Guido Scarabelli, Giulia Morra, and Giorgio Colombo. Predicting interaction sites from the energetics of isolated proteins: A new approach to epitope mapping. *Biophysical Journal*, 98(9): 1966–1975, 2010. doi: 10.1016/j.bpj.2010.01.014.

[61] Filippo Marchetti, Riccardo Capelli, Francesca Rizzato, Alessandro Laio, and Giorgio Colombo. The subtle trade-off between evolutionary and energetic constraints in protein–protein interactions. *The Journal of Physical Chemistry Letters*, 10:1489–1497, 2019. doi: 10.1021/acs.jpclett.9b00191.

[62] Davide Bagordo, Gauthier Trèves, Mariangela Santorsola, Giorgio Colombo, and Francesco Lescai. Adaptive disorder as the hallmark of nanobodies antigen-binding loops. *bioRxiv*, 2025. doi: 10.1101/2025.10.10.681624. preprint, version 1, posted 2025-10-10.

[63] Gareth A Tribello, Massimiliano Bonomi, Davide Branduardi, Carlo Camilloni, and Giovanni Bussi. Plumed 2: New feathers for an old bird. *Computer Physics Communications*, 185(2): 604–613, 2014.

[64] Thomas E. Creighton. *Proteins: Structures and Molecular Properties*. W. H. Freeman and Company, New York, 2 edition, 1993.

[65] Jack Kyte and Russell F. Doolittle. A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology*, 157(1):105–132, 1982.

[66] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[67] Akhila Melarkode Vattekatte, Nicolas K. Shinada, Tarun J. Narwani, Floriane Noël, Olivier Bertrand, Jean-Philippe Meyniel, Alain Malpertuy, Jean-Christophe Gelly, Frédéric Cadet, and Alexandre G. de Brevern. Discrete analysis of camelid variable domains: sequences, structures, and in-silico structure prediction. *PeerJ*, 8:e8408, March 2020. doi: 10.7717/peerj.8408.

[68] Elena Frasnetti, Ivan Cucchi, Silvia Pavoni, Francesco Frigerio, Fabrizio Cinquini, Stefano A. Serapian, Luca F. Pavarino, and Giorgio Colombo. Integrating molecular dynamics and machine learning algorithms to predict the functional profile of kinase ligands. *Journal of Chemical Theory and Computation*, 20(20):9209–9229, 2024. doi: 10.1021/acs.jctc.4c01097. URL `https://doi.org/10.1021/acs.jctc.4c01097`. Epub 2024-10-10.

[69] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2 edition, 2009. doi: 10.1007/978-0-387-84858-7.

[70] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, and et al. SciPy 1.0: fundamental algorithms for scientific computing in python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

[71] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[72] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. In *Noise Reduction in Speech Processing*, pages 1–4. Springer, 2009. doi: 10.1007/978-3-642-00296-0_5.

[73] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[74] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. 14(2):1137–1145, 1995.

[75] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[76] Ivan Cucchi, Elena Frasnetti, Francesco Frigerio, Fabrizio Cinquini, Silvia Pavoni, Luca F. Pavarino, and Giorgio Colombo. MOLECULE: Molecular-dynamics and optimized deep learning for entropy-regularized classification and uncertainty-aware ligand evaluation. *Journal of Chemical Theory and Computation*, 21(18):9186–9199, September 2025. doi: 10.1021/acs.jctc.5c01140.

[77] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, pages 8024–8035, Vancouver, BC, Canada, 2019. Curran Associates, Inc.

[78] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. In David E. Rumelhart and James L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, pages 318–362. MIT Press, 1986.

[79] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[80] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[81] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

[82] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017.

[83] David W. Hosmer and Stanley Lemeshow. *Applied Logistic Regression*. Wiley, 2 edition, 2000.

[84] Thomas M. Cover and Peter E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967. doi: 10.1109/TIT.1967.1053964.

[85] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. doi: 10.1023/A:1010933404324.

[86] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, 2001. doi: 10.1214/aos/1013203451.

[87] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2016. doi: 10.1145/2939672.2939785.

[88] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. doi: 10.1007/BF00994018.

[89] Zhi-Hua Zhou and Ji Feng. Deep forest: Towards an alternative to deep neural networks. *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 3553–3559, 2019.

[90] Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms*. CRC Press, 2012.

[91] David M. W. Powers. Evaluation: From precision, recall and f-measure to roc, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2011.

[92] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009. doi: 10.1016/j.ipm.2009.03.002.

[93] Julian D. Olden and Donald A. Jackson. An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data. *Ecological Modelling*, 178(3-4):389–397, 2004. doi: 10.1016/j.ecolmodel.2004.03.013.

[94] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2014.

[95] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models. *Journal of Machine Learning Research*, 20(177):1–81, 2019.

[96] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

[97] David W. Hosmer, Stanley Lemeshow, and Rodney X. Sturdivant. *Applied Logistic Regression*. Wiley, 3 edition, 2013. doi: 10.1002/9781118548387.

[98] Carolin Strobl, Anne-Laure Boulesteix, Achim Zeileis, and Torsten Hothorn. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, 8(25):1–21, 2007. doi: 10.1186/1471-2105-8-25.

[99] Skipper Seabold and Josef Perktold. Statsmodels: Econometric and statistical modeling with python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference (SciPy 2010)*, pages 92–96, 2010.

[100] Nathaniel J. Smith et al. patsy: A python library for describing statistical models, 2018. Version 0.5+; accessed 2025.

[101] S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3-4):591–611, 1965.

[102] Morton B. Brown and Alan B. Forsythe. Robust tests for the equality of variances. *Journal of the American Statistical Association*, 69(346):364–367, 1974.

[103] Student. The probable error of a mean. *Biometrika*, 6(1):1–25, 1908. doi: 10.2307/2331554.

[104] Henry B. Mann and Donald R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, 18(1):50–60, 1947. doi: 10.1214/aoms/1177730491.

[105] Emanuel Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(3):1065–1076, 1962. doi: 10.1214/aoms/1177704472.

[106] David W. Scott. On optimal and data-based histograms. *Biometrika*, 66(3):605–610, 1979. doi: 10.1093/biomet/66.3.605.

[107] David Freedman and Persi Diaconis. On the histogram as a density estimator: L2 theory. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 57(4):453–476, 1981. ISSN 1432-2064. doi: 10.1007/BF01025868.

[108] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, 2006. ISBN 978-0-387-31073-2.

[109] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.