

UNIVERSITA' DEGLI STUDI DI ROMA TOR VERGATA



MACROAREA DI SCIENZE MATEMATICHE, FISICHE E NATURALI

CORSO DI LAUREA MAGISTRALE IN BIOINFORMATICA

TESI DI LAUREA

Next Generation Sequencing del mitogenoma di *Caretta caretta* per l'identificazione di polimorfismi e analisi del loro effetto funzionale sulle strutture proteiche

Relatore:

Prof. Mattia Falconi
Prof. Andrea Novelletto

Candidato:

Andrea Ninni

Correlatore:

Dott. Federico Iacovelli

Anno Accademico 2021/2022

Indice

1 Abstract	3
2 Introduzione	5
2.1 Tecniche di sequenziamento NGS	5
2.2 Analisi NGS per l'identificazione di varianti	6
2.3 Effetti strutturali degli SNP	7
2.4 Analisi delle varianti in <i>Caretta caretta</i>	10
2.5 Scopo della tesi	11
3 Materiali e Metodi	12
3.1 Sequenziamento Illumina	12
3.2 Analisi dei dati di sequenziamento NGS	15
3.2.1 Quality Control	16
3.2.2 Trimming	17
3.2.3 Read Mapping	20
3.3 Metodi di modellazione molecolare	25
3.3.1 Ricerca dei <i>template</i>	27
3.3.2 Modellazione con Modeller	28
3.3.3 Mutazioni con FoldX	29
3.4 Dinamica molecolare	30
3.4.1 Principi di dinamica molecolare classica	31
3.4.2 <i>Force Field</i>	32
3.4.3 Rappresentazione del solvente	33

3.4.4	Calcolo delle interazioni	34
3.4.5	<i>Ensemble</i> di simulazione e controllo delle condizioni del sistema	35
3.4.6	Simulazione di dinamica molecolare delle subunità ND2	35
3.4.7	Analisi delle componenti principali del moto	37
4	Risultati e Conclusioni	39
4.1	Risultati del <i>read mapping</i>	39
4.2	Risultati del modeling per omologia	42
4.3	Risultati della dinamica molecolare	46
4.3.1	Analisi del calcolo delle strutture secondarie	46
4.3.2	Analisi delle componenti principali	47
5	Conclusioni	51
6	Bibliografia	53

1 Abstract

The loggerhead sea turtle (*Caretta caretta*) inhabits the tropical and subtropical waters of the Atlantic, Pacific and Indian Oceans, extending its range to the temperate Mediterranean Sea where it is the most common sea turtle and is found both as a visitor and nesting species. This species exhibit a natal homing behavior, whereby adult females mate in open waters and land to lay their eggs on or near the same beaches where they were born decades before.

The existence of Mediterranean populations implies that they have withstood ongoing climatic fluctuations, a result also proposed for some subtropical species currently found in the Mediterranean (Wilson and Eigenmann Veraguth, 2010). As poikilothermic vertebrates, sea turtles should have been strongly impacted by such fluctuations, given the temperature dependence of many aspects of their biology (phenology, migration, reproduction, and sex determination) (Mazaris et al., 2004; Hawkes et al., 2007). These conditions have the potential to drive the frequency of new positively selected mutations and the surrounding genomic regions.

The strong functional conservation of proteins encoded by the mitogenome allows an inference of their three-dimensional structure from that obtained experimentally in related species. This opens new possibilities for the understanding of variant properties: based on the 3D model of the protein, all substitutions can be evaluated in the appropriate protein context, which includes insertion into the phospholipid bilayer, solvent exposure, and interaction with other subunits in multiprotein complexes (Somero, 2010).

To study the presence of polymorphic variants and analyze their effect on the three-dimensional protein structures, 73 individuals of *Caretta caretta* sampled in the Mediterranean Sea were sequenced using the Illumina sequencing platform. We analyzed the quality of the fastq files using FastQC software and trimmed the read using Trimmomatic software and then proceeded with read mapping: the sequences were aligned to the RefSeq reference mitochondrial genome (Maglott et al., 2000), and using software, all variants that differed from the reference genome were identified and annotated using SnpEff. The variant calling procedure resulted in the identification of 24 unique variants that were mapped to genes encoding tRNA(Cys and Gly), 16S rRNA, genes

encoding protein subunits of mitochondrial complexes, and D-loop.

The project further aims to analyze the effect of missense variants on protein structures, all non-synonymous variants identified were considered and analyzed in their structural context.

To study of the effect of variants, the 6 subunits carrying missense variants were modeled: ATP6, COX3, CYTB, ND2, ND3 and ND5. For each sequence, templates were selected from the PDB (Berman et al., 2000) through BLAST (Altshul et al., 1999), and modeling was performed by homology modeling techniques through Modeller software (Webb and Sali, 2016) using the UCSF Chimera GUI (Pettersen et al., 2004); finally they were mutated through Foldx software. All mutations associated a slight change in free energy, not having a strong impact on protein structure destabilization, except for the ND2 subunit that exhibited a change of 3.04807 Kcal/mol.

To study the effect of mutation on the protein structure of the ND2 subunit, we performed two simulations of the protein subunit in the wild-type and mutated form, which were analyzed by PCA and compared. Molecular dynamics trajectory analyses were performed with GROMACS and scripts in Python.

Analysis of secondary structure evolution revealed few differences between the two subunits, which are limited principally to the C-terminal portion and only for a limited simulation time, consistent with rearrangements that can occur in terminal portion of proteins.

The analysis of the cross-correlation matrices and the projection of the principal motion described by the first eigenvector showed that the mutated subunit exhibits greater stiffness than the wild-type, showing mobility only in the solvent-exposed loop region. In conclusion, the ND2 subunit is able to absorb the effect of the Ala103Thr mutation without affecting functionality although a slight alteration of its mobility.

2 Introduzione

2.1 Tecniche di sequenziamento NGS

Il termine NGS (*Next Generation Sequencing*) non indica una sola tecnica ma si riferisce ad un insieme eterogeneo di tecnologie di sequenziamento post-Sanger sviluppate nell'ultimo decennio e che stanno avendo un enorme impatto sulla genomica comparativa e funzionale. Queste innovazioni includono *sequencing-by-ligation*, *ion semiconductor sequencing* e altri, sebbene ad oggi la tecnica più utilizzata sia il *sequencing-by-synthesis* impiegato dai dispositivi Illumina (Muzzey et al., 2015).

Lo sviluppo di queste nuove tecnologie di sequenziamento massivo parallelo è nato dai recenti progressi nel campo delle nanotecnologie, dalla disponibilità di strumenti ottici in grado di rilevare e differenziare milioni di fonti di luce o fluorescenza sulla superficie di un piccolo vetrino e dell'ingegnosa applicazione dei principi classici della biologia molecolare al problema del sequenziamento. Un'altra considerazione importante è che, nel contesto di una sequenza genomica già disponibile, molti problemi come l'identificazione di varianti a singolo nucleotide (SNP), non richiedono la generazione di sequenze lunghe, consentendo l'assegnazione univoca anche delle *read* più brevi a un locus in un genoma di riferimento. Pertanto, le tecnologie NGS disponibili producono un gran numero di *read* di sequenze brevi e sono tipicamente utilizzate in applicazione di *read mapping* che implicano la disponibilità di una sequenza di riferimento identica, o altamente simile, all'origine del materiale genetico in esame (Horner et al., 2009).

L'innovazione chiave che trasforma la replicazione del DNA nella strategia di sequenziamento del DNA alla base di Sanger e NGS è l'uso di basi modificate non estensibili e marcate con fluorescenza. Nel sequenziamento Sanger, solo una piccola percentuale di basi è modificata, mentre nel NGS tutte le basi sono modificate; in entrambe le tecniche di sequenziamento, quando la polimerasi incorpora una base modificata nel filamento copiato, l'estensione del nuovo filamento si interrompe e questo viene colorato in modo univoco per leggere la base aggiunta più di recente. La sfida fondamentale per il sequenziatore è, quindi, organizzare le molecole in modo che il loro segnale di fluorescenza sia interpretabile: le tecniche NGS, invece di sfruttare la separazione dimensionale per disporre le molecole fluorescenti, utilizza la separazione posizionale, milioni di filamenti

diversi di DNA si legano in posizioni discrete su un vetrino e rimangono fissi nella stessa posizione durante l'intera reazione di sequenziamento.

La rivoluzione in corso nella tecnologia di sequenziamento del DNA consente oggi la lettura di migliaia o milioni di basi nucleotidiche in un'unica esecuzione strumentale. Tuttavia, questa quantità di dati è spesso compromessa da una scarsa fiducia nella qualità della lettura. L'identificazione di polimorfismi genetici a partire da questi dati è quindi problematica e, insieme alla grande quantità di dati, rappresenta una grande sfida bioinformatica (Imelfort, 2009).

2.2 Analisi NGS per l'identificazione di varianti

I polimorfismi a singolo nucleotide (SNP) sono varianti alleliche del DNA che si formano come risultato di singoli errori di replicazione e che hanno una frequenza allelica apprezzabile nella popolazione. Ci sono diverse ragioni per cui gli SNP sono al centro di molti studi sulla genetica. In primo luogo, una piccola parte dei disturbi umani sono semplici malattie monogenetiche: si ritiene che la maggior parte dei fenotipi delle malattie umane sia di natura complessa e coinvolga varianti comuni del DNA insieme a fattori ambientali. L'identificazione delle varianti che aumentano la suscettibilità alle malattie umane è uno dei problemi chiave della genetica medica, in quanto la loro identificazione può portare alla scoperta di nuovi approcci terapeutici. In secondo luogo, l'analisi della variazione genetica in una popolazione può aiutare a trovare soluzioni a molti problemi della genetica evolutiva (Sunyaev et al., 2001).

L'identificazione di SNP avviene attraverso l'analisi di dati di sequenziamento, nella quale le *read* prodotte sono tipicamente allineate al corrispondente riferimento genomico nel processo bioinformatico chiamato *read mapping* o *re-sequencing*, nel caso in cui sia disponibile un genoma di riferimento (Horner et al., 2009), anche se applicazioni più recenti si sono concentrate su metodi di assemblaggio *de novo* di dati genomici, che permettono di scoprire nuovi geni e forniscono un modello per la scoperta di SNP per le molte specie per le quali non esiste ancora un genoma di riferimento strettamente correlato. Oltre alla scelta tra il *read mapping* e il sequenziamento *de novo*, è anche possibile scegliere tra un approccio *whole-genome*, in cui si sequenzia l'intero genoma dell'organismo o a complessità ridotta, nel quale si sequenziano solo alcuni locus ge-

netici: per i genomi grandi e complessi, gli approcci di sequenziamento a complessità ridotta forniscono una profondità di sequenza adeguata per la scoperta di SNP senza la necessità di campionare il genoma completo (Imelfort, 2009).

Il *read mapping* viene utilizzato per identificare le variazioni genetiche tra gli individui, che possono fornire marcatori genetici molecolari e nuove conoscenze sulla funzione dei geni o per la validazione e la valutazione dei marcatori genetici nelle popolazioni. Il processo di *read mapping* dell'intero genoma mediante tecnologie di sequenziamento NGS prevede l'allineamento di un insieme di milioni di *read* a una sequenza genomica di riferimento. Una volta ottenuto questo risultato, è possibile determinare la variazione della sequenza nucleotidica tra il campione e il riferimento (Huang et al., 2009).

Alla fine delle analisi, la probabilità che uno SNP dedotto sia reale può essere valutata utilizzando statistiche di inferenza bayesiana ma le piattaforme NGS possono migliorare l'accuratezza del rilevamento degli SNP grazie ad una maggiore profondità di sequenziamento: singole mutazioni possono essere rilevate in modo affidabile sia tramite una strategia di sequenziamento *paired-end* sia con una *coverage* elevata in modo tale che, dopo l'allineamento, siano selezionate le posizioni con un'alta probabilità di essere SNP (Smith et al., 2008).

Una parte molto importante della ricerca genomica alla base dello studio delle varianti è l'analisi della variazione genetica nelle popolazioni: il nostro focus si concentra sull'applicazione dei dati strutturali per arrivare a nuove conoscenze nei campi di genetica delle popolazioni ed evoluzione.

2.3 Effetti strutturali degli SNP

La predizione dell'effetto degli SNP a livello funzionale è difficile da determinare, anche le variazioni nelle regioni codificanti e nelle regioni regolatorie sono le principali variazioni che vanno ad inficiare sulla funzione del gene. Le mutazioni missenso (o non sinonime) alterano la sequenza amminoacidica del prodotto proteico attraverso la sostituzione di un amminoacido (Mooney, 2004): possono avere effetti neutri, quindi non alterare in modo rilevante il fenotipo oppure possono determinare differenze nelle caratteristiche individuali (Wang and Moult, 2001).

A livello di sequenza amminoacidica, ci si aspetta che le mutazioni missenso che in-

fluenzano la funzione siano quelle che producono le sostituzioni meno conservative: la ricchezza di dati sperimentali sull'effetto della mutagenesi diretta sulla struttura e sulla funzione proteica rende possibile prevedere i probabili cambiamenti nella struttura o nella funzione molecolare, e quindi studiare un modello dell'impatto funzionale di questi SNP missenso (Wang and Moult, 2001).

Ovviamente, l'effetto fenotipico di una sostituzione nucleotidica è sempre causato da cambiamenti strutturali o funzionali nel DNA, nell'RNA o nelle proteine. Sebbene le sostituzioni nucleotidiche in molte regioni del DNA non codificante possano essere importanti dal punto di vista funzionale, l'analisi dello spettro di frequenza allelica suggerisce che gli alleli selettivamente non neutrali presenti nella popolazione sono molto più frequenti tra le varianti alleliche proteiche. Solo una frazione degli SNP rientra in questa categoria, ovvero la sottofrazione di quegli SNP che si verificano nella sequenza codificante. Pertanto, l'analisi dei "fenotipi molecolari", cioè delle caratteristiche allele-specifiche nella struttura, nel ripiegamento, nel legame o nella stabilità proteica, possono aiutare a spiegare il meccanismo biologico dell'effetto fenotipico o addirittura a prevedere tale effetto (Sunyaev et al., 2001).

L'aumento dei dati strutturali e genomici consente un approccio più sistematico a quest'area di ricerca. Recentemente sono stati fatti tentativi sistematici di mappare gli SNP missenso sulle strutture 3D delle proteine e di analizzare il possibile impatto delle varianti alleliche sulla struttura o sulla funzione delle proteine. Sunyaev e collaboratori (Sunyaev et al., 2000) hanno mappato una serie di SNP non sinonimi provenienti da *database* pubblici su strutture 3D delle proteine corrispondenti e hanno analizzato la loro posizione strutturale, insieme alla conservazione della sequenza dei siti SNP in famiglie di proteine omologhe. Le caratteristiche strutturali e di conservazione dei siti SNP sono state poi confrontate con le caratteristiche delle sostituzioni amminoacidiche tra le proteine umane e i loro ortologhi strettamente correlati e con le caratteristiche delle mutazioni note per essere responsabili di malattie. È stato dimostrato che il numero di SNP localizzati in siti strutturalmente o funzionalmente importanti è significativamente più alto rispetto alle sostituzioni tra le specie (anche se è ovviamente inferiore rispetto alle mutazioni patologiche). I risultati di questa analisi suggeriscono che una frazione significativa di SNP non sinonimi probabilmente influisce sulla struttura o sulla funzio-

ne delle proteine e quindi potrebbe costituire alleli deleteri per il fenotipo. Una ricerca successiva dello stesso gruppo mostra che le varianti amminoacidiche deleterie possono essere previste da semplici considerazioni strutturali, insieme alle tecniche convenzionali di analisi della sequenza.

Gli studi di mutagenesi in vitro insieme ai dati sul contesto strutturale delle mutazioni missenso che causano malattia sono stati usati per sviluppare un modello dell'impatto funzionale degli SNP missenso. Ogni mutazione è associata a un effetto su uno o più ruoli del residuo interessato. I ruoli che possono essere influenzati sono: la stabilità o il ripiegamento delle proteine, il legame con i ligandi, la catalisi, regolazione mediante meccanismi allosterici e di altro tipo o modificazioni post-traduzionali (Wang and Moult, 2001).

Una questione importante nel caso di SNP è capire se una particolare mutazione sarà tollerata. Ci sono diversi modi in cui uno SNP può influenzare la funzione di un prodotto genico, l'effetto più probabile è una perdita parziale o totale della funzione del prodotto genico mutato, e in casistiche più rare c'è anche il guadagno di una nuova funzione (Mooney, 2004). L'analisi strutturale di questi dati può aiutare a determinare i residui che sono cruciali per interazioni specifiche o per la formazione della struttura proteica nativa. Dall'altro lato, l'analisi strutturale e funzionale delle sostituzioni amminoacidiche può rivelare il background molecolare di particolari malattie genetiche, determinare i principali meccanismi responsabili dell'effetto distruttivo delle mutazioni che causano la malattia e, eventualmente, aiutare a sviluppare algoritmi di predizione. Recentemente, sono stati sviluppati numerosi algoritmi computazionali per prevedere l'impatto delle sostituzioni nucleotidiche o amminoacidiche sulla struttura, l'espressione e la funzione delle proteine. La conservazione evolutiva della sequenza amminoacidica può essere determinata allineando sequenze amminoacidiche di proteine correlate provenienti da organismi non correlati o da famiglie di geni. Sono stati proposti diversi algoritmi che utilizzano i dati di sequenza del DNA o degli amminoacidi per identificare residui o domini potenzialmente funzionali in un confronto con le sequenze presenti nelle banche dati pubbliche. Questi includono metodi basati sulla struttura tridimensionale (3D) delle proteine, metodi basati su considerazioni evolutive e approcci di apprendimento automatico (Mooney, 2004).

2.4 Analisi delle varianti in *Caretta caretta*

La tartaruga di mare (*Caretta caretta*) abita le acque tropicali e subtropicali dell’Oceano Atlantico, Pacifico e Indiano, estendendo il suo areale anche al temperato Mar Mediterraneo dove è la tartaruga marina più comune e si trova sia come visitatore sia come specie nidificante. La specie mostra un comportamento di *natal homing*, in base al quale le femmine adulte si accoppiano in acque aperte e sbarcano per deporre le uova sulle stesse spiagge, o in prossimità, di quelle in cui sono nate decenni prima. La conseguenza attesa, anche per molte generazioni, è una forte strutturazione tra le colonie per le porzioni di genoma ereditate per via materna (come il DNA mitocondriale), a seconda della composizione del gruppo fondatore, della deriva genica e di colli di bottiglia (Badga et al., 2012; Clusa et al., 2013).

L’esistenza di popolazioni mediterranee implica che queste hanno resistito alle fluttuazioni climatiche in atto, un risultato proposto anche per alcune specie subtropicali attualmente presenti nel Mediterraneo (Wilson and Eigenmann Veraguth, 2010). In quanto vertebrati poichilotermi, le tartarughe marine dovrebbero essere state fortemente impattate da tali fluttuazioni, vista la dipendenza dalla temperatura di molti aspetti della loro biologia (fenologia, migrazione, riproduzione e determinazione del sesso) (Mazaris et al., 2004; Hawkes et al., 2007). Queste condizioni possono potenzialmente consentire l’aumento della frequenza di nuove mutazioni selezionate positivamente, che alla fine vanno a caratterizzare intere regioni genomiche adattate.

In funzione dell’elevato tasso mutazionale nel mitogenoma dei vertebrati, le analisi di regioni polimorfiche possono fornire un’ampia gamma di varianti che potenzialmente influiscono sui prodotti genici corrispondenti. Inoltre, la forte conservazione funzionale delle proteine codificate a livello mitocondriale consente un’inferenza della loro struttura tridimensionale a partire da quella ottenuta sperimentalmente in specie affini. Questo apre nuove possibilità nelle analisi della distribuzione delle varianti: sulla base del modello 3D della proteina, tutte le sostituzioni possono essere valutate nel contesto proteico appropriato, che comprende l’inserimento nel doppio strato fosfolipidico, l’esposizione al solvente e l’interazione con altre subunità in complessi multiproteici (Somero, 2010).

In uno studio precedente (Novelletto et al., 2016), sono state sequenziate porzioni codi-

ficanti del mitogenoma di 170 esemplari di *Caretta caretta* campionati nel Mediterraneo centrale. In particolare il sequenziamento e l'analisi strutturale ha riguardato i geni ND1 e ND3. Per comprendere meglio il significato biologico delle variazioni osservate, sono state analizzate le sostituzioni non sinonime intraspecifiche nel contesto delle strutture tridimensionali modellate de polipeptidi ND1 e ND3: le mutazioni riscontrate su ND1 e ND3 non hanno compromesso la funzionalità della struttura ma sono state riscontrate in prossimità dei confini della membrana, in linea con un meccanismo che coinvolge la modellazione dello spessore, della fluidità e della composizione lipidica della membrana, un processo noto come adattamento omeoviscoso già ampiamente documentato nei vertebrati poichilotermi.

2.5 Scopo della tesi

Lo scopo di questa tesi, che estende lo studio effettuato in precedenza, è quello di identificare tutte le regioni che ospitano polimorfismi nell'intero mitogenoma delle tartarughe marine *Caretta caretta* campionate nel Mediterraneo centrale e di studiare gli effetti delle varianti codificanti sulla struttura e funzione proteica per valutarne il significato biologico. A questo scopo, l'Università degli studi di Firenze ha campionato 74 individui da diversi siti italiani e ne ha eseguito il sequenziamento tramite tecniche NGS Illumina *paired-end*.

3 Materiali e Metodi

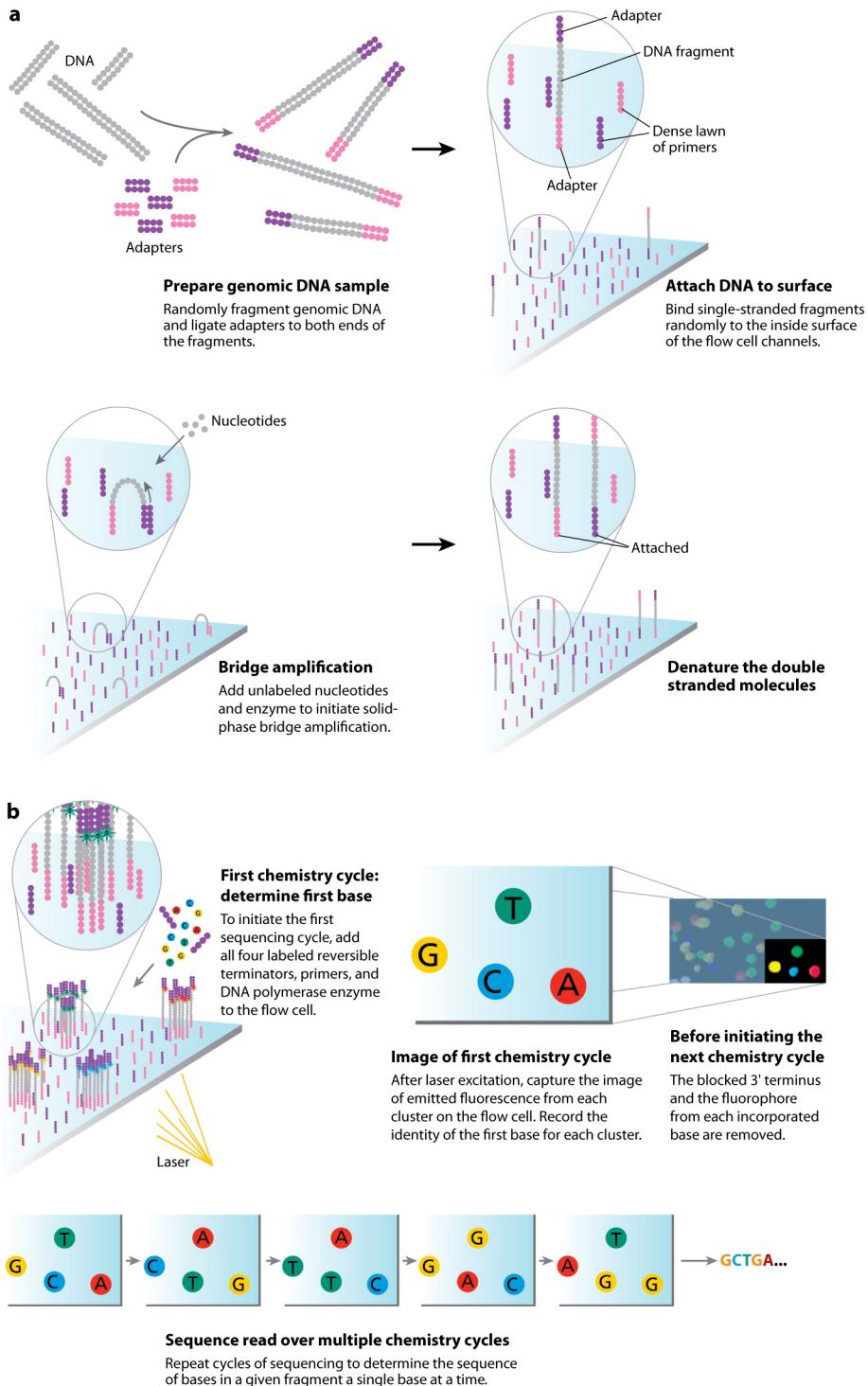
3.1 Sequenziamento Illumina

La tecnologia di sequenziamento Illumina è oggi la più utilizzata, nonché quella relativamente più economica e con la maggiore capacità di sequenziamento relativamente al volume di *read* prodotte (Muzzey et al., 2015; Mardis, 2008; Shendure and Ji, 2008).

Il supporto fisico sul quale viene effettuato il sequenziamento è un dispositivo micro-fabbricato in vetro sigillato a 8 canali, chiamato *flow cell*, che consente l'amplificazione di frammenti sulla sua superficie grazie alla presenza di oligonucleotidi ancorati con sequenza nota, che sono sfruttati come *primer* nella successiva reazione di amplificazione. Il DNA da sequenziare è sottoposto ad un processo di frammentazione fisica o enzimatica, in seguito alla quale i frammenti ottenuti vengono selezionati sulla base della loro lunghezza: i frammenti della lunghezza desiderata vengono legati a opportuni adattatori complementari agli oligonucleotidi presenti sulla superficie della *flow cell* e sono poi denaturati. Le sequenze di DNA a singolo filamento così ottenute sono inoculate sulla *flow cell* e la complementarietà tra le sequenze presenti sulla superficie e gli adattatori assicura il loro ancoraggio sul supporto.

Le sequenze ancorate alla superficie del vetrino sono amplificate tramite PCR con la tecnica della *bridge amplification*: i frammenti si ripiegano a ponte grazie all'ibridazione con gli oligo complementari sulla *flow cell* e la reazione di amplificazione sfrutta questi oligomeri come *primer*. Terminata la reazione, il DNA viene denaturato e si procede con il ciclo di amplificazione successivo. Dopo diversi cicli di amplificazione si ottengono dei *cluster* ognuno dei quali contiene circa un milione di copie del frammento originale, necessarie per ottenere una buona intensità di segnale richiesta per il rilevamento durante il sequenziamento.

Il sistema Illumina utilizza un approccio di *sequencing-by-synthesis* in cui i 4 nucleotidi vengono aggiunti simultaneamente ai canali della *flow cell*, insieme alla DNA polimerasi, per essere incorporati nei *cluster*: i nucleotidi portano un'etichetta fluorescente unica per ogni base che blocca chimicamente il gruppo 3' OH in modo che ogni incorporazione sia un evento unico. La reazione di sequenziamento di per sé è una variazione del metodo di sequenziamento Sanger, in cui l'innovazione principale consiste nell'utilizzo



A Mardis ER. 2008.
R Annu. Rev. Genomics Hum. Genet. 9:387–402

Figura 1: Fasi dell’approccio di sequenziamento illumina: preparazione e ancoraggio delle librerie, *bridge amplification*, sequenziamento e *base-calling* (Mardis, 2008)

di terminatori fluorescenti reversibili: durante i diversi cicli di reazione, questi dNTP vengono aggiunti in quantità equimolare sulla *flow cell* e, per ogni ciclo, in ciascuno dei cluster avviene una ed una sola incorporazione di base, cui corrisponde un’emissione luminosa fluorescente ad una specifica lunghezza d’onda. Una fase di *imaging* segue ogni fase di incorporazione della base, durante la quale un apparato ottico registra un’immagine istantanea. Dopo ogni fase di *imaging*, il gruppo bloccante viene rimosso chimicamente, i reagenti in eccesso sono rimossi dalla superficie del vetrino e il ciclo si ripete. Il sequenziamento Illumina può essere di due tipi a seconda se i frammenti sono sequenziati da una sola estremità (sequenziamento *single-end*) o da entrambe le estremità (sequenziamento *paired-end*).

Al termine del processo di sequenziamento, i segnali chimico-fisici ottenuti nel corso della reazione devono essere decodificati allo scopo di convertirli, tramite l’utilizzo di opportuni algoritmi, nelle sequenze che essi rappresentano. Questo processo prende il nome di *base-calling* nel quale i ”file primari”, le immagini scattate dagli apparati di sequenziamento durante la reazione, sono convertiti in ”file secondari”, che contengono le stringhe in formato *human readable* di tutte le read sequenziate.

Gli algoritmi di *base-calling*, inoltre, assegna dei punteggi di qualità, o *quality score*, per ciascuna base determinata nel sequenziamento, denominato anche *Phred score*: è un punteggio numerico (Q), solitamente compreso tra 1 e 40, che rappresenta in scala logaritmica la probabilità di errore (P) associata alla chiamata di ciascuna base, secondo la formula:

$$Q = -10 \log_{10} P \quad (1)$$

Il formato fastq (Cock et al., 2010) è attualmente il formato di riferimento adottato per rappresentare i risultati di una reazione di sequenziamento NGS: è un file di tipo testuale in cui i dati associati con ciascuna sequenza vengono riportati su 4 righe separate:

- La prima riga è un’intestazione caratteristica che inizia con il carattere @ e contiene un identificativo alfa-numerico univoco della *read*; questo identificativo ha lo scopo di rendere ciascuna *read* unica e riconoscibile ma anche di incorporare informazioni che consentono di risalire alla posizione fisica della *flow cell* da cui la *read* stessa deriva.

- La seconda riga contiene la sequenza ordinata delle basi ottenute dall'algoritmo di *base-calling* in cui le basi sono rappresentate secondo la covenzione IUPAC con l'aggiunta del simbolo N che rappresenta una chiamata non definibile.
- La terza riga contiene il carattere separatore + che introduce la riga successiva con i punteggi di qualità.
- La quarta riga contiene i punteggi di qualità associati alla chiamata delle basi indicizzato allo stesso modo della sequenza presente nella seconda riga; per motivi di spazio i punteggi non vengono rappresentati su base numerica ma utilizzando un'apposita convenzione basata sui caratteri ASCII: ciascun numero viene rappresentato dalla lettera che nella tabella ASCII corrisponde al numero stesso aumentato da una costante definita, +33 nel caso del Phred+33 e +64 nel caso del punteggio Phred+64, consentendo in questo modo di rappresentare un numero a due cifre con un solo carattere.

```

@M01527:39:000000000-DCTFV:1:1101:15249:7408 1:N:0:6
CCCCCCCCTCACACACAAACACACTCTCCTGCTGGTAATAGCT
TATCTAAAGTGACCACTCTCCTTGTATGGGGG
+
BCC3BBCCDFDFGGGGGGGGGGHHHHHHHHHHHHHHHHHHH
HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHG

```

Figura 2: Esempio di una *read* in un file fastq

Nel caso di un sequenziamento *paired-end* sono prodotti due file FASTQ uno per il filamento *forward* e uno per il filamento *reverse*: i due file contengono lo stesso numero di righe e quindi lo stesso numero di *read* e nello stesso ordine, le prima *read* del primo file corrisponde alla prima *read* del secondo file e così via.

3.2 Analisi dei dati di sequenziamento NGS

Non tutte le basi che compongono una *read* hanno lo stesso livello di qualità, ma solitamente tende a diminuire più si avvicina al 3'. È quindi necessario verificare in

ogni esperimento di sequenziamento come varia la qualità al variare della posizione sulla *read* come valutazione qualitativa della procedura sperimentale effettuata (Long et al., 2018).

L'analisi di qualità delle *read* si divide in due passaggi:

- Controllo di Qualità, nella quale si va ad valutare la qualità delle *read* tramite parametri statistici e la presenza di eventuali problematiche come la presenza degli adattatori;
- *Trimming*, che permette di migliorare i risultati scartando le *read* di bassa qualità.

Dopo il controllo di qualità, le *read* sono sottoposte al *read mapping* una strategia nella quale si ricostruisce la sequenza iniziale allineando le *read* ad un genoma di riferimento noto a priori che ci permette l'identificazione di SNP o INDEL nel campione sequenziato rispetto al genoma di riferimento e successivamente le varianti identificate sono annotate.

3.2.1 Quality Control

I moderni metodi di sequenziamento possono generare milioni di sequenze in una singola *run*. Prima di analizzare queste sequenze per trarre conclusioni biologiche bisogna effettuare un controllo della qualità per assicurare che i dati di *output* non presentano alcun problema che si potrebbe ripercuotere sulle analisi a valle.

Il *software* utilizzato per il controllo di qualità è FastQC che crea un *report* della qualità in grado di individuare eventuali problemi avvenuti nel corso del sequenziamento.

L'analisi in FastQC è eseguita tramite una serie di moduli che producono dei grafici (Fig. 3) in grado di fornire una panoramica generale su vari parametri riguardo il sequenziamento e le *read* ottenute, e una valutazione qualitativa del risultato che viene espressa tramite una spunta verde, un punto esclamativo arancione o una croce rossa. FastQC può essere eseguito in due modalità: come un'applicazione *stand-alone* con interfaccia grafica per l'analisi immediata di file fastq, o può essere usato in modalità non interattiva, tramite linea di comando, adattata per *pipeline* di analisi nel caso di grandi quantità di dati.

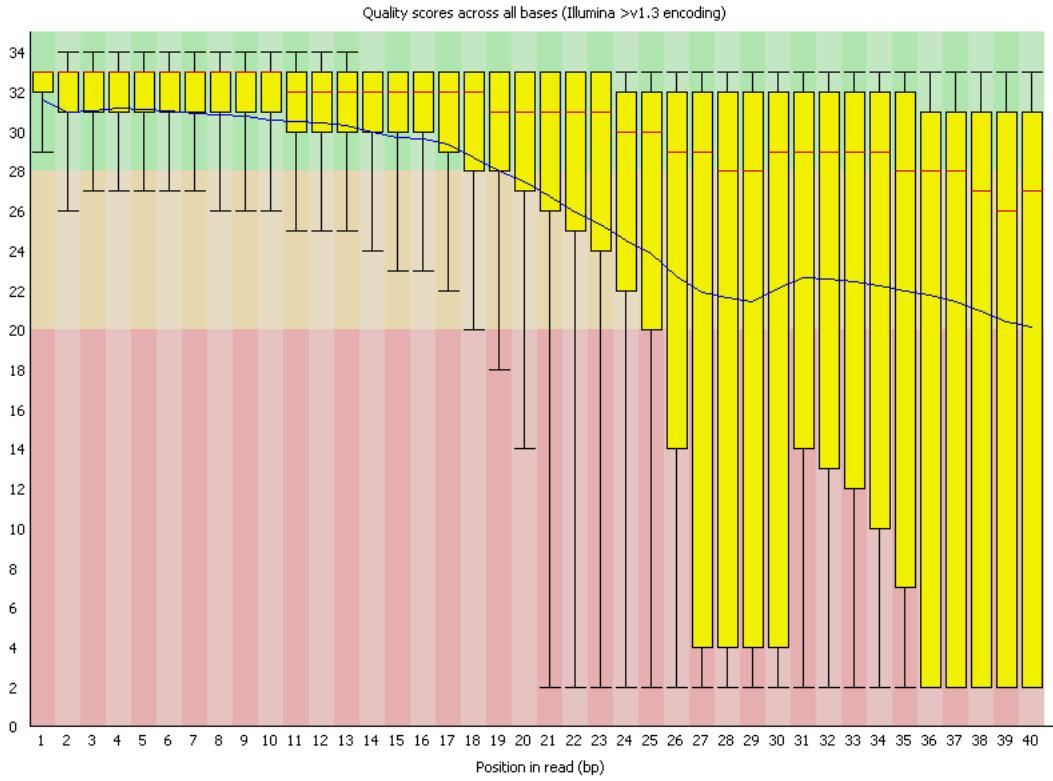


Figura 3: Barplot generato dal modulo di analisi *per base sequence quality* di FastQC

Il software mira a fornire un modo semplice per eseguire controlli di qualità sui dati di sequenziamento procedendo a:

- valutare la qualità dei dati generati
- fornire statistiche riassuntive e basilari dei dati grezzi
- analizzare diverse caratteristiche come la presenza di adattatori
- esportazione dei risultati in un report basato su HTML
- consentire generazione automatica di report senza applicazione interattiva

3.2.2 Trimming

Il *trimming* è il processo che va ad eliminare tutte le *read* all'interno dei file fastq con bassa qualità migliorando la qualità del sequenziamento effettuato.

Questo procedimento è stato effettuato con il *software* TRIMMOMATIC (Bolger et al, 2014) con l'approccio *paired-end* che permette di mantenere la corrispondenza delle

coppie di *read* nei due file fastq e di utilizzare le informazioni aggiuntive contenute nelle *read* accoppiate per agevolare la ricerca di eventuali adattatori o di frammenti di *primer* PCR che sono stati introdotti nel processo di preparazione della libreria e sequenziati erroneamente.

Come *input* sono specificati due file fastq, uno per le *read forward* e uno per le *read reverse* e il programma restituisce in *output* quattro file, di cui due fanno riferimento ai risultati *paired* in cui sono sopravvissute entrambe le *read* nei due file fastq e i restanti due inglobano l'*output unpaired* in cui è sopravvissuta solo una delle due *read*.

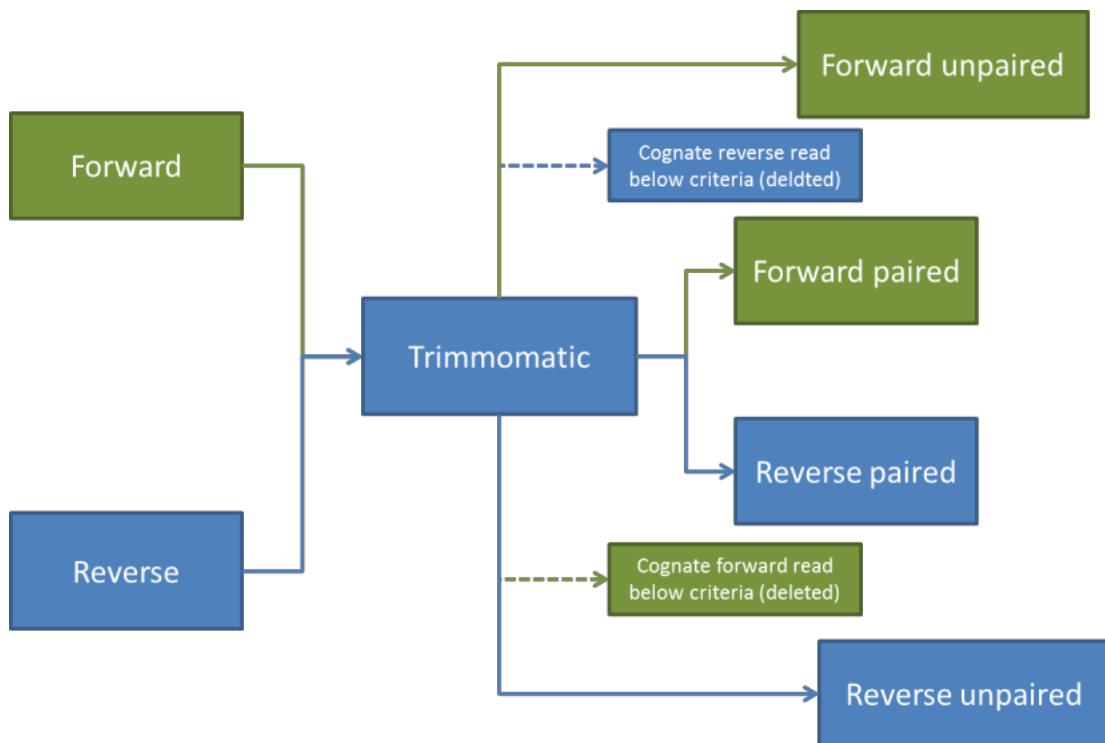


Figura 4: Schematizzazione della modalità *paired-end* di TRIMMOMATIC

TRIMMOMATIC offre due principali approcci di filtraggio della qualità delle *read*, *sliding window quality filtering* e *Maximum Information quality filtering*, che sfruttano i punteggi di qualità Illumina di ciascuna posizione di base, determinando in tal modo il punto in cui la *read* necessita di essere tagliata. La selezione delle fasi di *trimming*, l'ordine e i parametri sono forniti tramite riga di comando: TRIMMOMATIC implementa degli *step* ai quali vengono forniti in *input* le condizioni selezionate per il *trimming* dei dati in *input*.

Per i nostri dati in *input*, il modulo *per base quality content* di fastqc ha messo in

evidenza un anormale contenuto delle quattro basi nelle prime 10 posizioni tipico del sequenziamento Illumina, quindi come primo *step* sono state rimosse le prime 10 basi di ogni *read* tramite la funzione HEADCROP:10; come approccio di filtraggio è stato usato lo *sliding window* applicato sia al 5' sia al 3' con un *quality score* di 28, tramite le funzioni TRAILING:28 e LEADING:28 e infine sono state scartate tutte le *read* con una lunghezza inferiore di 15 paia di basi tramite MINLEN:15.

Questo passaggio è stato automatizzato tramite il seguente *script* in linguaggio di programmazione *shell bash*:

```
#!/bin/bash

in_dir=fastq_files
trim_dir=trim_data

for forward in `ls $in_dir/*R1*`; do
    reverse=${forward/R1/R2}
    echo $forward
    echo $reverse

    basename_file=$(basename $forward)

    trimlog=$trim_dir/${basename_file/_R1_*/_trimlog.txt}
    for_pai=$trim_dir/${basename_file/_R1_*/_R1_pai.fastq.gz}
    for_unp=$trim_dir/${basename_file/_R1_*/_R1_unp.fastq.gz}
    rev_pai=$trim_dir/${basename_file/_R1_*/_R2_pai.fastq.gz}
    rev_unp=$trim_dir/${basename_file/_R1_*/_R2_unp.fastq.gz}

    java -jar Trimmomatic-0.39/trimmomatic-0.39.jar PE -phred33 -thread 4 \
        -trimlog $trimlog \
        $forward $reverse \
        $for_pai $for_unp $rev_pai $rev_unp \
        HEADCROP:10 TRAILING:28 LEADING:28 MINLEN:15
```

done

3.2.3 Read Mapping

Il *read mapping* è il primo step in molte pipeline di genomica comparativa, in particolare per le pipeline di *variant calling* (McKenna et al., 2010), che consiste nell'allineamento delle *read* ad un genoma di riferimento: il sequenziamento del genoma di molti individui per i quali esiste un genoma di riferimento consente di studiare la relazione tra le variazioni di sequenza e tipi di malattia o di normalità (Stratton, 2008).

Il *read mapping* si svolge in 4 step ognuno dei quali è mediato da uno specifico *software* (Van der Auwera et al., 2013):

- Allineamento delle *read* al genoma di riferimento
- Compressione del file SAM in BAM
- Chiamata delle varianti
- Filtraggio e annotazione delle varianti

I programmi di allineamento più utilizzati per i dati di sequenziamento NGS sono stati progettati esplicitamente per l'allineamento delle *read* che, a differenza di programmi di allineamento come BLAST, deve risolvere il problema computazionale di mappare piccole sequenze di DNA su un genoma di riferimento della specie di interesse. Gli attuali metodi di allineamento delle *read* si basano sulla trasformata di Burrows-Wheeler, un algoritmo utilizzato come metodo di compressione dei dati che crea un indice efficiente dell'assemblaggio del genoma di riferimento in modo da facilitare la ricerca rapida con basso utilizzo di memoria (Flicek et al., 2009).

La trasformata di Burrows-Wheeler, o BWT (Burrows and Wheeler, 1994), è una trasformazione reversibile che utilizza delle sotto-stringhe per rappresentare una stringa originale in modo compresso permutandone l'ordine dei caratteri. Se la stringa originale contiene molte ripetizioni di certe sottostringhe, allora nella stringa trasformata troveremo diversi punti in cui lo stesso carattere si ripete tante volte. Ciò è utile per la compressione perché diventa facile comprimere una stringa in cui compaiono lunghe sequenze di caratteri tutti uguali, come nel caso di sequenze biologiche. L'algoritmo

Input	Rotazioni	Ordine Lessicografico	Traformata
^BANANA@	^BANANA@ @^BANANA A@^BANAN NA@^BANA ANA@^BAN NANA@^BA ANANA@^B BANANA@^	ANANA@^B ANA@^BAN A@^BANAN BANANA@^ NANA@^BA NA@^BANA ^BANANA@ @^BANANA	BNN^AA@A

Tabella 1: Esempio della trasformata di Burrows-Wheeler della parola ”BANANA”

trasforma una stringa S di N caratteri formando N rotazioni di S (*cyclic shifts*), ordinandoli in ordine lessicografico ed estraendo l’ultimo carattere di ogni rotazione: la trasformata L si forma da questi caratteri in modo tale che l’ i -esimo carattere di L è l’ultimo carattere dell’ i -esima rotazione (Tabella 1). Inoltre, l’algoritmo va a calcolare un indice I della stringa originale S nella lista delle rotazioni ordinate chiamato *array dei suffissi*.

I moderni allineatori utilizzati nel *read mapping* sfruttano la trasformata di Burrows-Wheeler per poter comprimere e indicizzare il genoma di riferimento in quanto richiede meno memoria RAM e fornisce un metodo efficiente per l’identificazione di match tra le *read* e il genoma. Questi metodi utilizzano tipicamente la struttura dati *FM-index* nella quale si mantiene un *array* dei suffissi basato sulla trasformata e non sulla sequenza originale: l’*FM-index* mantiene quindi la capacità dell’*array* dei suffissi di consentire una rapida ricerca di sottosequenze (Chae Na et al., 2016). La creazione dell’indice richiede due fasi: in una prima fase, l’ordine di sequenza del genoma di riferimento viene modificato con la trasformata di Burrows-Wheeler e successivamente viene creato l’indice finale che viene poi utilizzato per il posizionamento rapido delle *read* sul genoma.

Bowtie2 è un allineatore che sfrutta BWT e FM-index usando un approccio per identificare match imperfetti tra la sequenza *input* e il genoma di riferimento basato su *backtracking*, uno strumento efficace nella ricerca delle varianti rispetto al genoma di

riferimento (Langmead and Salzberg, 2012): si provano ad inserire delle sostituzioni nucleotidiche nella *read* fino a trovare una possibile sostituzione che consenta di poterla allineare al riferimento. Per ogni *read*, Bowtie2 procede in 4 *step*: estraе le sottostringhe, allinea le sottostringhe tramite FM-index, sono calcolate le posizioni dell'allineamento e estende le sottostringhe tramite programmazione dinamica.

Come genoma di riferimento, è stato utilizzato il genoma mitocondriale di *Caretta caretta* depositato su RefSeq (Maglott et al., 2000) con *accession number* NC_016923.1; questo è stato indicizzato con il comando *bowtie2-build*, per poi allineare le *read* con il comando *bowtie2*, che a partire dal genoma indicizzato e dai due file fastq restituisce in *output* un file di allineamento in formato SAM.

Il formato SAM (*Sequence Alignment Map*) è un formato di testo in cui viene rappresentato l'allineamento delle *read* contro un genoma di riferimento supportando sia lunghe sequenze sia sequenze più corte prodotte dalle diverse piattaforme di sequenziamento (Li et al., 2009). Il formato SAM consiste di due sezioni:

- *Header section*, una sezione con la funzione di sommario che contiene informazioni riguardanti l'allineamento. Ogni riga inizia con il carattere '@' seguito da un *tag* di due lettere che identifica il tipo di informazione contenuta nella riga
- *Alignment section*, sono rappresentati tutti gli allineamenti delle *read* contro il genoma di riferimento. Ogni linea va a rappresentare l'allineamento di una *read* ed è composta da 11 campi obbligatori che descrivono l'allineamento effettuato

Il file SAM (Fig. 5) ha il vantaggio di avere le informazioni rappresentate in formato coinciso, ma con lo svantaggio che la ricerca di una **read** in questo file non è computazionalmente efficiente in quanto è un file di elevate dimensioni. Per incrementare le performance, è stato progettato il formato BAM (*Binary Alignment Map*) che è la rappresentazione binaria del file SAM che contiene le stesse informazioni ma in formato compresso. I file BAM, inoltre, hanno il vantaggio di poter essere ordinati in funzione delle coordinate: questo permette sia una possibile indicizzazione del file BAM (formato *bai*) sia una ricerca efficiente delle *read* in un determinato locus.

La compressione dei file SAM è stata effettuata con samtools, una libreria e un pacchetto *software* per l'analisi e la manipolazione di allineamenti in formato SAM e BAM (Li

Figura 5: Esempio di un file sam

et al., 2009); questo *software* permette di convertire i file nei due formati, ordinare e unire gli allineamenti, rimuovere duplicati, generare informazioni per ogni posizione nel formato *pileup* e mostrare l'allineamento con un visualizzatore testuale. Il file SAM generato da bowtie2 è stato convertito in BAM con il comando *samtools view*, ordinato con *samtools sort* e infine indicizzato *samtools index*. Il risultato, infine, è stato visualizzato con Tablet (Milne et al., 2013), un visualizzatore grafico di allineamenti.

L'identificazione delle varianti (o *variant calling*) è la fase di analisi del *read mapping* che va ad individuare tutti i punti nei quali i dati di sequenziamento differiscono dal riferimento, identificando polimorfismi a singolo nucleotide (SNP) e piccole inserzioni e delezioni (*indels*). Uno dei problemi maggiori di questa operazione è data dalla difficoltà nel riuscire a distinguere le varianti vere dai falsi positivi: i due fattori principali che complicano il *variant calling* sono gli errori di sequenziamento, dovuti ad artefatti della PCR, e la variabilità della qualità delle *read*. Fortunatamente, questi problemi sono ridotti dagli algoritmi di *variant calling* che svolgono valutazioni della credibilità delle varianti rilevate tramite calcoli bayesiani o calcoli di linkage.

Per la chiamata delle varianti è stato utilizzato BCFtools, una libreria e un pacchetto software utilizzato per la manipolazione di file BAM per generare file VCF (*Variant Calling Format*), un formato file di testo che contiene, per ogni linea, informazioni su

tutte le varianti presenti nel campione sequenziato rispetto al genoma di riferimento, con vari parametri che indicano la qualità della chiamata, come ad esempio la profondità, *depth*, che va ad indicare il numero di *read* mappate che identificano la variante. BCFtools implementa i due comandi da utilizzare in *pipeline* per la chiamata delle varianti *bcftools mpileup* e *bcftools call* che prendono in *input* il file formato BAM e restituiscono in *output* il file VCF.

Una volta ottenute i file VCF, come ultimo passaggio del *read mapping*, le varianti individuate sono state filtrate in funzione del punteggio di qualità e della profondità, tramite *bcftools filter*, e annotate per osservare la conseguenze delle mutazione a valle dell'espressione. L'annotazione è stata effettuata tramite il *software* SnpEff, un programma per la categorizzazione degli effetti di SNP e indels (Cingolani et al., 2012). SnpEff annota e classifica i polimorfismi sulla base dei geni annotati, identificando varianti sinonime o varianti missenso, perdita o creazione di codoni di inizio, perdita o creazione di codoni di stop. SnpEff ha implementato un database di genomi annotati pronti per la classificazione delle varianti ma permette anche la creazione di database personali nel caso di organismi non presenti: nel file di configurazione di SnpEff, *snp_eff.config*, è possibile inserire il genoma annotato presente in Genbank e associare un codice genetico.

È stato creato il database di *Caretta caretta* a partire dal report su RefSeq con codice NC_016923.1 ed è stato configurato il codice genetico mitocondriale dei vertebrati.

Tutto l'analisi di *read mapping* è stata automatizzata tramite il seguente script in linguaggio di programmazione *shell bash*:

```

#!/bin/bash

in_dir=trim_data
ref=reference/FR694649.1.fasta
out_dir=output

for for_pai in `ls $in_dir/*R1*pai*`; do
    rev_pai=${for_pai/R1/R2}

    name=$(basename $for_pai)

```

```

base=$out_dir/${name/_R1*/}

bowtie2 --no-unal -p 4 -x $ref -1 $for_pai -2 $rev_pai -S $base.sam
samtools view -bt $ref -o $base.bam $base.sam
samtools sort $base.bam -o $base.sorted.bam
samtools index $base.sorted.bam
bcftools mpileup -f $ref $base.sorted.bam | bcftools call -mv > $base.snp.vcf
java -jar snpEff/snpEff.jar eff Caretta $base.snp.vcf > $base.annotato.vcf

done

```

3.3 Metodi di modellazione molecolare

La struttura terziaria/quaternaria di una proteina può essere determinata tramite metodi sperimentali o computazionali. Tra i metodi sperimentali troviamo la cristallografia ai raggi X, la spettroscopia NMR e la microscopia elettronica (principalmente microscopia crioelettronica). Questi metodi di risoluzione di strutture proteiche permettono di ottenere strutture proteiche molto accurate, ma ognuna presenta limiti intrinseci principalmente correlati con il protocollo della metodologia utilizzata, la dimensione delle macromolecole e con la completezza delle strutture determinate (Nwanochie and Uversky, 2019). A livello computazionale la predizione della struttura terziaria e quaternaria delle proteine rappresenta invece uno degli obiettivi principali della bioinformatica strutturale (Deng et al., 2018). Non esistono ancora metodi computazionali efficienti in grado di predire la struttura tridimensionale di una proteina basandosi unicamente su proprietà chimico-fisiche e sull'esplorazione sistematica di tutte le possibili conformazioni da essa assunte. Negli anni sono però stati sviluppati metodi computazionali che permettono la predizione della struttura tridimensionale di una sequenza proteica attraverso il confronto con strutture proteiche note (*template*). I metodi di modellazione si dividono in metodi di modellazione per omologia, metodi di *fold recognition* e metodi *ab initio*. La scelta del metodo da utilizzare dipende dall'identità di sequenza stimata tra la proteina *query* e uno o più *template*. I metodi di modellazione per omologia sono basati sul presupposto che sequenze simili tenderanno ad adottare strutture tridimen-

sionali simili, e dipendono quindi criticamente dall’identità di sequenza tra la *query* ed il/i *template* ($\geq 25\%$) (Deng et al., 2018). Nella modellazione per omologia regioni conservate tra *query* e *template* vengono direttamente utilizzate per la costruzione della nuova struttura, mentre regioni di inserzione o delezione vengono di solito modellate come regioni di *loop*, costruite tramite ricerca per similarità di struttura in database di *loop* noti oppure tramite modellazione diretta, ottimizzando una funzione energetica per trovare la struttura più favorevole per quel segmento (Muhammed and Aki-Yalcin, 2019).

Quando non esistono sequenze proteiche con una similarità significativa con una certa sequenza *query* vengono invece utilizzati i metodi di *fold recognition* (Jones et al., 1992). Questi metodi sono basati sull’assunzione che le strutture vengono conservate in maniera molto più stringente rispetto alle sequenze durante l’evoluzione e sull’evidenza che il numero di *fold* proteici unici presenti in natura ad oggi conosciuti è abbastanza limitato (Chothia, 1992). Il database SCOPe (Fox et al., 2014) riporta infatti 1257 fold unici nel 2022, con una percentuale di nuovi fold scoperti che diminuisce di anno in anno. Nei metodi di *fold recognition* come il *protein threading* vengono costruiti tanti possibili allineamenti di sequenza tra la sequenza *query* e un database di *fold* conosciuti utilizzando la programmazione dinamica (Khor et al., 2015). Viene valutata la fitness dei residui della sequenza *query* in ognuno dei *fold* considerati tramite funzioni statistiche ed energetiche. Queste funzioni tengono conto della compatibilità di ogni residuo nell’ambiente tridimensionale del *fold* considerato, dell’energia di interazione di residui spazialmente vicini e dell’energia dovuta all’introduzione e/o estensione di *gap* nella struttura. I metodi di modellazione per omologia e di *fold recognition* sono classificati come metodi comparativi basati su *template*. Quando non si ottengono risultati significativi tramite questi metodi, di solito a causa di similarità di sequenza troppo basse, si ricorre ad una modellazione *ab initio* basata sul presupposto che la struttura di una proteina nel suo stato nativo deve trovarsi al minimo globale della propria energia libera (Khor et al., 2015). Non viene utilizzato alcun *template* e viene effettuata una ampia ricerca nello spazio conformazionale utilizzando principalmente il metodo Monte Carlo (Hansmann and Okamoto, 1999). Metodi *ab initio* più avanzati sono basati sull’assemblaggio combinatoriale di una serie di frammenti della sequenza

query per i quali vengono derivate singolarmente molte strutture possibili tramite comparazione con proteine a struttura nota (Khor et al., 2015). Le coordinate strutturali del sistema di partenza generato vengono continuamente modificate in maniera casuale portando ad un’ampia esplorazione dello spazio conformazionale. Tramite i metodi *ab initio* vengono quindi generati molti possibili modelli al minimo di energia libera.

Per la modellazione molecolare sono state prese in considerazione tutte le subunità dei complessi proteici mitocondriali sui cui geni sono state annotate le varianti missenso trovate nel *read mapping*: l’elevata conservazione della struttura delle proteine mitocondriali permette l’utilizzo di metodi di *homology modeling*.

3.3.1 Ricerca dei *template*

La ricerca dei *template* per la modellazione delle proteine *target* è stata effettuata tramite BLAST (Altschul et al., 1990) sul PDB (*Protein Data Bank*), e per ciascuna sequenza è stata selezionata la struttura del miglior templato in base ai valori di *coverage* e *identity*. Un *template* adatto per la modellazione per omologia dovrebbe possedere una catena aminoacidica completa e avere una buona risoluzione cristallografica.

BLAST (*Basic Local Alignment Search Tool*) è un algoritmo usato per la ricerca per similarità in banche dati di sequenze: una ricerca in BLAST permette di confrontare una sequenza di interesse con un database di sequenze note e di identificare quelle che presentano somiglianze con la sequenza di interesse.

Data una sequenza *query* da utilizzare per la ricerca e una banca dati, l’algoritmo di BLAST si suddivide in 3 fasi principali:

- Creazione dei *W-mer*, l’algoritmo inizialmente suddivide la sequenza *query* in sottostringhe sovrapposte di una lunghezza data W, che per le sequenze di aminoacidi tipicamente assume valori di 2, 3 o 6, dalle quali viene generata una lista di parole affini ad esse, chiamate *W-mer*.
- Ricerca dei *W-mer* in banca dati, l’algoritmo seleziona dalla banca dati solo quelle sequenze che contengono almeno un frammento di lunghezza W uguale ad uno dei *W-mer* prodotti da tutta la sequenza *query*.

- Elongazione delle sequenze hit, le corrispondenze trovate tra *W-mer* della sequenza *query* e frammenti di lunghezza W delle sequenze della banca dati (dette *hit*) vengono estesi sia a monte che a valle senza inserire *gap*. Per ogni coppia di aminoacidi aggiunta all'allineamento viene determinato il relativo punteggio ottenuto dalla matrice di sostituzione utilizzata.

Ogni allineamento è valutato tramite il parametro HSP (*high-scoring segment pair*) che definisce una soglia massima di perdita di punteggio: nel caso in cui l'estensione dell'allineamento porti all'inclusione di coppie di aminoacidi il cui relativo punteggio nella matrice di sostituzione è negativo, l'algoritmo verifica che il punteggio complessivo non cada al di sotto della soglia HSP.

3.3.2 Modellazione con Modeller

Modeller, rilasciato per la prima volta nel 1989 da Andrej Sali, è un software che permette di effettuare una modellazione comparativa di strutture tridimensionali basata su *restraint* strutturali a partire da uno o più allineamenti di sequenza (Fiser et al., 2000; Šali and Blundell, 1993; Webb and Sali, 2016). Il metodo utilizzato da Modeller viene chiamato *satisfaction of spatial restraints* ed utilizza una serie di vincoli geometrici strutturali per determinare una funzione di densità di probabilità, che viene ottimizzata per individuare la struttura più favorevole assunta da ognuno dei residui della sequenza *query*. Questa funzione dipende dalle coordinate cartesiane di ognuno degli atomi, da parametri geometrici (distanze, angoli e diedri tra i singoli atomi) e da parametri che aiutano a descrivere i *restraint*. I vincoli da applicare alle strutture possono essere ottenuti automaticamente tramite allineamenti di sequenza tra la proteina *query* e uno o più *template* a struttura nota, oppure possono essere forniti tramite strutture NMR o altri dati sperimentali eventualmente conosciuti. Modeller incorpora inoltre delle funzioni che permettono la predizione ab initio di regioni di loop, che a causa della loro ampia variabilità anche tra proteine omologhe sono spesso difficili da predire tramite metodi comparativi (Webb and Sali, 2016).

Il software Modeller può essere installato ed utilizzato da linea di comando su sistemi Unix tramite *script* in Python, oppure può essere utilizzato tramite un'interfaccia automatizzata implementata nel software UCSF Chimera (Pettersen et al., 2004). Nel-

la versione *stand-alone* Modeller richiede in *input* una sequenza *query* da modellare e uno o più *template* strutturali (Webb and Sali, 2016). I *template* vengono allineati alla sequenza *query* tramite la funzione *align2d*. Sulla base di questi allineamenti il programma estrae dei *restraints* spaziali (distanze, angoli, angoli diedri, ecc.) che vengono utilizzati per ottimizzare una funzione di densità di probabilità che permette di effettuare una modellazione sulla base dei *restraints* forniti.

Modeller è stato utilizzato tramite UCSF Chimera che offre un’interfaccia grafica nella quale forniamo al software la struttura del *template*, la sequenza da modellare e vari parametri, e come *output* restituisce il modello della proteina: il programma allinea la sequenza *query* da modellare al templato strutturale e calcola un modello tridimensionale della proteina target con il modulo *automodel*. Per ogni sequenza target sono stati generati cinque modelli differenti e tra questi è stato selezionato il migliore in base ai valori di DOPE score ottenuti per ogni struttura tridimensionale generata. Il *Discrete Optimized Protein Energy* è un potenziale statistico che viene utilizzato per valutare la qualità del modello per omologia: più il DOPE score è basso e migliore sarà il modello.

3.3.3 Mutazioni con FoldX

Foldx è un campo di forza empirico sviluppato per valutare l’effetto di mutazioni sulla stabilità, il *folding* e la dinamica di proteine e acidi nucleici (Schymkowitz et al., 2005; Buß et al., 2018): la funzionalità principale di FoldX è calcolare l’energia libera di una macromolecola sulla base della sua struttura 3D tramite una combinazione lineare di termini empirici.

$$\begin{aligned} \Delta G = & a \cdot \Delta G_{vdw} + b \cdot G_{solvH} + c \cdot \Delta G_{solvP} + d \cdot \Delta G_{wb} + e \cdot \Delta G_{hbond} \\ & + f \cdot \Delta G_{el} + g \cdot \Delta G_{kon} + h \cdot T \Delta S_{mc} + k \cdot T \Delta S_{sc} + l \cdot \Delta G_{clash} \end{aligned} \quad (2)$$

Questa equazione include termini per la desolvatazione polare e idrofobica o l’energia del legame a idrogeno ΔG_{wb} di una proteina che interagisce con il solvente e all’interno della catena proteica. I termini da a-l sono i pesi relativi, calcolati empiricamente, dei diversi termini energetici utilizzati per il calcolo dell’energia libera, in questo modo FoldX fornisce una stima rapida e quantitativa dell’importanza delle interazioni che contribuiscono alla stabilità delle proteine e dei complessi proteici.

Il pacchetto *software* Foldx implementa diverse funzioni come, ad esempio, la funzione

RepairPDB che minimizza l’energia di una struttura proteica riorganizzando le catene laterali mentre la funzione *BuildModel* introduce mutazioni e ottimizza la struttura della nuova proteina mutata. La funzione di energia di Foldx è in grado di calcolare la differenza di energia tra il *wild-type* e una variante della proteina:

$$\Delta\Delta G = \Delta G_{wild-type} - \Delta G_{variant} [kcal \cdot mol^{-1}] \quad (3)$$

Prima di procedere alla mutazione dei modelli, FoldX richiede che la struttura sia prima minimizzata per ottenere risultati affidabili: i modelli sono stati riparati mediante il comando *RepairPDB* di FoldX che avvia una minimizzazione dell’energia ottimizzando la posizione delle catene laterali degli amminoacidi per ottenere una minore energia libera della proteina e rimuovendo i *clashes* di Van Der Waals. Come *output* restituisce il PDB del modello minimizzato e un file fxout nel quale sono riportate le energie dei residui riparati.

I modelli minimizzati sono stati mutati con il comando *BuildModel* che per ogni mutazione effettuata sulla proteina ottimizza la posizione dei residui prossimali vicini nel *wild-type* producendo per ogni struttura un PDB mutante. La funzione richiede in *input* un *mutant file* nel quale è riportata sulla prima riga la sequenza originale da mutare e nelle righe sottostanti le sequenze mutate. Per ogni subunità da modellare è stato creato il relativo *mutant-file* e sono state effettuate cinque *run* di *BuildModel*: questa procedura è necessaria per la convergenza dell’algoritmo e quindi per ottenere dei risultati affidabili dal punto di vista biologico. Come *output* il programma restituisce i file PDB dei cinque modelli e i file Dif.fxout e Average.fxout nelle quali sono riportate, rispettivamente, le energie di mutazione dei modelli e l’energia media.

3.4 Dinamica molecolare

La dinamica molecolare classica è una tecnica computazionale che permette di studiare l’evoluzione dinamica a livello atomico di macromolecole biologiche, come proteine ed acidi nucleici, attraverso l’integrazione delle equazioni del moto. Le simulazioni di dinamica molecolare vengono generalmente utilizzate per interpretare dati sperimentali, misurare grandezze all’equilibrio e seguire l’evoluzione temporale di un sistema. Negli ultimi anni, grazie all’aumento delle capacità di calcolo e al miglioramento dei

software e degli algoritmi utilizzati, è possibile studiare sistemi sempre più complessi su scale temporali che vanno dai picosecondi (ps) fino ai millisecondi (ms) nel caso di sistemi molto piccoli. Tra i programmi più utilizzati per effettuare simulazioni di dinamica molecolare troviamo AMBER (Case et al., 2005), NAMD (Phillips et al., 2005), CHARMM (Brooks et al., 2009) e GROMACS (Berendsen et al., 1995). Il punto di partenza di una simulazione di dinamica molecolare è la struttura tridimensionale di una macromolecola. Questa struttura viene inserita in un box di simulazione opportunamente solvatato e neutralizzato aggiungendo dei controioni (Na^+ e Cl^-) prima di essere simulato.

3.4.1 Principi di dinamica molecolare classica

La dinamica molecolare si basa sulla meccanica molecolare classica e consente, partendo da velocità iniziali e da un set di coordinate assegnate a tutti gli N atomi che compongono il sistema, di esplorare un sistema molecolare e la sua evoluzione. Le velocità iniziali vengono selezionate in maniera casuale dalla distribuzione gaussiana di Maxwell-Boltzmann, in grado di descrivere la distribuzione delle velocità di un atomo in funzione della temperatura, attraverso la formula:

$$p(v) = \frac{m_i}{2\pi k_b T}^{\frac{1}{2}} e^{-\frac{mv^2}{2k_b T}} \quad (4)$$

dove p rappresenta la probabilità della componente velocità per la particella i -esima, k_b equivale alla costante di Boltzmann e T alla temperatura del sistema.

L'evoluzione del sistema viene monitorata mediante integrazione della legge di Newton, su cui si basa il secondo principio della dinamica:

$$F_i = m_i \cdot a_i = \frac{m_i \cdot \delta v}{\delta t} = \frac{m_i \cdot \delta^2 r_i(t)}{\delta t^2} \quad (5)$$

dove i rappresenta la particella i -esima su cui agisce F , ovvero la forza, al tempo t ; m_i è la massa della particella e r_i è il vettore posizionale della particella i -esima, ottenuto delle coordinate tridimensionali della particella al tempo t .

La meccanica molecolare calcola, per ogni molecola parametrizzata, un potenziale (V) di interazione tra le particelle del sistema; da questo potenziale è possibile ottenere, per derivazione, la forza esercitata su ogni atomo in ogni punto dello spazio:

$$F = -\frac{\delta V}{\delta f_i} \quad (6)$$

da cui si può ricavare l'accelerazione e la velocità di ogni atomo nel sistema:

$$a_i = -\frac{1}{m} \cdot \frac{\delta V}{\delta r_i} \quad (7)$$

$$v_i = \int \frac{-\frac{\delta V}{\delta r_i}}{m} dt \quad (8)$$

Questa tipologia di equazione e, in generale, le equazioni del moto sono molto complesse per poter essere risolte in forma analitica per i sistemi oggetto dell'indagine di dinamica molecolare.

Per questi motivi, le equazioni del moto possono essere integrate utilizzando il metodo delle differenze finite che consiste nel calcolare l'evoluzione di posizioni e velocità nell'arco di piccoli intervalli discreti di tempo Δt , definiti *timestep*. È necessario, di conseguenza, considerare il tempo come una variabile discreta e non continua, sostituendo le derivate con i rispettivi rapporti incrementali. Il *timestep* deve essere abbastanza breve per poter integrare adeguatamente le equazioni del moto ma abbastanza lungo da descrivere le proprietà di interesse nel sistema. Se si conoscono le posizioni, le velocità iniziali degli atomi nel sistema e la forza che agisce su ogni atomo, è possibile ricavare il valore dell'accelerazione di ogni atomo e derivare posizioni e velocità nell'istante $t + \Delta t$, considerando la forza costante durante il Δt .

L'algoritmo più utilizzato nell'ambito della dinamica molecolare per consentire l'integrazione delle equazioni del moto con il metodo delle differenze finite è l'algoritmo di Verlet (Verlet, 1967): una volta suddiviso l'intervallo d'interesse in piccoli intervalli di tempo Δt , l'algoritmo fa uso della serie di Taylor per espandere il vettore composto dall'insieme delle coordinate cartesiane del sistema, negli intorni $t + \Delta t$ e $t - \Delta t$. Un altro algoritmo utilizzato per l'integrazione delle equazioni del moto attraverso il metodo delle differenze finite è il leap frog (Van Gunsteren and Berendsen, 1988), una variante del Verlet.

3.4.2 Force Field

I *force field* rappresentano un set di parametri in grado di descrivere le interazioni presenti nel sistema attraverso potenziali empirici i cui parametri vengono calcolati sulla base di dati sperimentali e calcoli di quanto-meccanica. La serie di termini che costituiscono i campi di forze, se sommati, sono in grado di determinare l'energia potenziale

del sistema:

$$E_{tot} = E_{bond} + E_{angle} + E_{tors} + E_{vdw} + E_{elec} \quad (9)$$

Si tratta, dunque, della sommatoria di tutte le interazioni di legame, nell'equazione date dai primi tre termini, e delle interazioni di non legame, elettrostatiche e di Van Der Waals, a partire dalle coordinate del sistema per ogni Δt .

I force field sono in continuo aggiornamento e miglioramento, i più utilizzati sono: AMBER (Weiner et al., 1984), principalmente per gli acidi nucleici; CHARMM (Brooks et al., 2009), per le membrane; GROMOS (Oostenbrink et al., 2004) e OPLS (Jorgensen and Tirado-Rives, 1988).

Il *software* di simulazione GROMACS, utilizzato per le simulazioni di questa tesi, implementa numerosi *force field* in particolare è stato utilizzato GROMOS 53A6 i cui file sono stati modificati per effettuare la simulazione, oltre che di proteine, anche dei lipidi che compongono le membrane biologiche simulate.

3.4.3 Rappresentazione del solvente

Le simulazioni per essere più veritieri possibile non possono essere condotte nel vuoto, ma devono tener conto del contributo apportato dall'acqua sul sistema in analisi. A tale scopo, nel corso degli anni, sono stati sviluppati diversi modelli in grado di rappresentare le molecole d'acqua (Wallqvist and Mountain, 2007) che inglobano un set di parametri specifici in grado di riprodurre le proprietà fisiche e termodinamiche dell'acqua come la densità, l'entalpia di vaporizzazione e il momento di dipolo. Il modello più utilizzato è il TIP3P (Jorgensen et al., 1983) e considera tre siti di interazione con carica puntiforme, con geometria rigida, i quali rappresentano i tre atomi che compongono la molecola d'acqua; l'atomo di ossigeno possiede un set di parametri per le interazioni di Lennard-Jones. Altri modelli più recenti, vedi il TIP4P o il TIP5P (Mahoney and Jorgensen, 2000), aggiungono dei miglioramenti in termini di distribuzione della carica elettrostatica della molecola a scapito del tempo di simulazione, dato l'incremento delle interazioni da calcolare.

3.4.4 Calcolo delle interazioni

All'interno del sistema soggetto a simulazione è necessario, come detto, avere una panoramica circa tutte le tipologie di interazioni tra gli atomi all'interno del sistema. Le interazioni di legame si basano sulla meccanica molecolare in cui gli atomi vengono considerati come sfere solide legate da molle. In base a questa considerazione è possibile calcolare le funzioni di potenziale.

$$U_{bonded} = U_{bond-stretch} + U_{bond-bend} + U_{dihedrals} \quad (10)$$

dove:

- $U_{bond-stretch}$ corrisponde al potenziale di Morse associato allo stiramento del legame covalente che unisce due atomi; per descriverla si ricorre alla legge di Hooke.
- $U_{bond-bend}$ rappresenta il piegamento dell'angolo ovvero la deviazione degli angoli rispetto a un angolo di riferimento; anche questo termine può essere descritto in funzione della legge di Hooke.
- $U_{dihedrals}$ descrive la funzione di potenziale associata agli angoli di torsione in presenza di barriere steriche tra gli atomi separati da tre legami covalenti. Tale potenziale periodico è espresso attraverso una funzione coseno.

Anche le funzioni di potenziale che riguardano le interazioni di non legame, si basano su leggi di fisica classica che aiutano a semplificare la descrizione delle interazioni stesse:

$$U_{non-bonded} = U_{LJ} + U_{Coulomb} \quad (11)$$

dove:

- U_{LJ} rappresenta il potenziale di Lennard-Jones, modello matematico che tiene conto di un contributo attivo e di uno repulsivo.
- $U_{Coulomb}$ descrive le interazioni elettrostatiche tra due atomi, ricorre alla legge di Coulomb.

Il caso più difficoltoso di una simulazione è rappresentato dal calcolo delle interazioni a lungo raggio, molto gravoso e pesante da un punto di vista computazionale soprattutto se consideriamo che il quantitativo di forze da calcolare aumenta in modo esponenziale all'incremento del numero di atomi nel sistema. Per agevolare questo aspetto, sono state effettuate delle approssimazioni (Brooks, 1989), quella più semplice è il *cut-off* che impone un limite entro il quale calcolare le interazioni a lungo raggio, una volta superato questo limite qualsiasi forza di interazione tra due atomi viene ignorata. Questa approssimazione è buona se si considerano le forze di Lennard-Jones a corto raggio ma non agevola il calcolo delle forze di Coulomb a lungo raggio che, invece, sfruttano metodi come le somme di Ewald (Allen and Tildesley, 1989) o il PME (*Particle Mesh Ewald*) (Darden et al., 1993). Utilizzando tali metodiche è possibile separare il calcolo delle interazioni elettrostatiche a corto raggio da quelle a lungo raggio che sfruttano per il calcolo una trasformata di Fourier veloce.

3.4.5 *Ensemble* di simulazione e controllo delle condizioni del sistema

I risultati ottenuti dalle simulazioni di dinamica molecolare il più delle volte necessitano di essere confrontati con dati sperimentali ottenuti da condizioni di temperatura e pressione ambientale. Per il mantenimento di tali parametri a livello simulativo sono stati introdotti diversi algoritmi. Il controllo della temperatura, ad esempio, viene effettuato con l'utilizzo del termostato di Berendsen (Berendsen et al., 1994) o con quello di Nosè-Hoover (Evans and Holian, 1985) o, ancora, con il termostato di Langevin (Allen and Tildesley, 1998). Per il controllo della pressione vi sono degli algoritmi, detti barostati, tra cui, ad esempio, si annoverano: il barostato di Berendsen (Berendsen, 1994) o quello di Nosè-Hoover Langevin (Quigley and Probert, 2004).

3.4.6 Simulazione di dinamica molecolare delle subunità ND2

Come descritto nel paragrafo 3.2 dei risultati, le precedenti analisi di modellazione e mutazione, hanno messo in evidenza che la mutazione presente sulla subunità ND2 ha un effetto destabilizzante sulla struttura. Per studiare l'effetto della mutazione sulla

funzione della proteina, sono state effettuate due simulazioni di dinamica molecolare classica:

1. nella prima simulazione è stata prodotta una traiettoria di 100 ns della subunità ND2 di fenotipo *wild-type* modellato precedentemente, in presenza del doppio strato fosfolipidico, solvente e ioni per portare il sistema alla neutralità
2. nella seconda simulazione è stata prodotta una traiettoria di 100 ns della subunità ND2 di fenotipo mutato modellato precedentemente, in presenza del doppio strato fosfolipidico, solvente e ioni per portare il sistema alla neutralità

Le due subunità sono state impacchettate nel doppio strato fosfolipidico, in questo caso è stato usato un doppio strato di dipalmitoilfosfatidilcolina (DPPC), tramite la *InfilateGRO methodology* (Kandt et al., 2007) implementata in uno script in perl: questo metodo consiste nel posizionare la proteina e i lipidi all'interno di una griglia per poi restringerla fino a che i lipidi che impacchettano la proteina non presentano la densità desiderata. Sono state effettuate 26 iterazioni ognuna delle quali ha ridotto la griglia di 0.95 Å ottenendo un'area per lipide di circa 71 Å². Il sistema è stato solvatato con molecole di acqua spc216 e un secondo script perl ha eliminato tutte le molecole di acqua in collisione con il sistema, al quale sono stati aggiunti gli ioni per raggiungere la neutralità.

Allo scopo di stabilizzare il sistema prima di procedere con la fase di simulazione, è stata eseguita una prima minimizzazione per verificare che i valori di energia potenziale e delle forze massime convergessero prima di iniziare le simulazioni.

Le strutture minimizzate sono state termalizzate utilizzando l'*ensemble* canonico NVT ed un timestep di 2.0 fs, partendo da una temperatura di 0 K ed aumentandola gradualmente di 10 K ogni 30 ps, fino ad arrivare ad una temperatura finale di 323 K. A questo punto è stata aggiunta pressione al sistema tramite l'*ensemble* NPT, il timestep è stato mantenuto a 2 fs e ha avuto inizio la dinamica di produzione. Le interazioni elettrostatiche sono state calcolate ogni 2 passi (4 fs) attraverso il metodo Particle-Mesh Ewald (Darden et al., 1993), utilizzando una griglia con una spaziatura di 1,6 Å. Per le interazioni di non legame è stato impostato un raggio di cut-off di 12 Å, aggiornando la lista dei vicini ogni 20 passi e con una distanza di 14,5 Å per l'inclusione degli atomi nella lista. La temperatura delle simulazioni è stata mantenuta costante a 323 K

utilizzando il termostato di Nose-Hoover, con un damping coefficient di accoppiamento di 0,5 ps da applicare a tutti gli atomi mentre Per il controllo della pressione è stato utilizzato il metodo Parrinello-Rahman.

3.4.7 Analisi delle componenti principali del moto

La *Principal Component Analysis* (PCA) è una tecnica statistica utilizzata per estrarre un modello a partire da una grande quantità di dati (Pearson, 1901). La PCA utilizza una procedura matematica che trasforma un certo numero di variabili, verosimilmente correlate, in un numero inferiore di variabili non correlate, chiamate componenti principali (o autovettori). Nell'analisi di traiettorie MD di macromolecole biologiche, la PCA viene generalmente utilizzata per identificare le direzioni e le ampiezze dei moti principali che caratterizzano la struttura analizzata (Amadei et al., 1993). Durante un'analisi di PCA vengono rimossi i moti roto-traslazionali della proteina attraverso una procedura di *mass-weighted least squared fitting* e viene poi calcolata la covarianza di ogni coppia di atomi A e B nel sistema. La covarianza è una misura statistica che permette di calcolare il rapporto tra più dimensioni dello stesso insieme di dati:

$$Cov(X, Y) = \frac{1}{N} \cdot \sum (X_i - \bar{X}) \cdot (Y_i - \bar{Y}) \quad (12)$$

dove N è il numero di frame della traiettoria, \bar{X} è la posizione media dell'atomo X, \bar{Y} è la posizione media dell'atomo Y e X_i e Y_i sono le posizioni degli atomi X e Y al frame i.

I valori di covarianza ottenuti per ogni coppia di atomi vengono raggruppati in una matrice simmetrica composta da N righe ed N colonne (dove N è il numero di atomi). La matrice di covarianza viene diagonalizzata per calcolare i relativi autovettori, ottenendo così $3N$ autovettori ed i corrispondenti autovalori. Gli autovettori vengono ordinati in maniera decrescente partendo dall'autovettore con l'autovalore maggiore. Il primo autovettore viene chiamato prima componente principale del moto e descrive la maggior parte della variabilità dei dati in esame, mentre le componenti successive rappresentano una certa quantità della variabilità residua. Nel caso della dinamica molecolare, ciascun autovettore descrive una direzione del moto, pertanto è possibile scomporre il movimento totale in tutte le sue componenti e considerare così solo gli autovettori che danno il massimo contributo al movimento complessivo di una molecola (moti a bassa

frequenza), escludendo movimenti meno informativi come le vibrazioni atomiche (moti ad alta frequenza).

L'analisi PCA è stata effettuata sulle subunità ND2 *wild-type* e mutata per confrontare i cambiamenti conformazionali della proteina in funzione della mutazione. La matrice di covarianza è stata calcolata utilizzando il *tool covar* di GROMACS (Berendsen et al., 1995), ed utilizzando uno script in Python è stata prodotta la corrispondente matrice di cross-correlazione. I primi due autovettori generati sono stati poi scelti per essere analizzati tramite il *tool anaeig* di GROMACS. In questo modo è stata ottenuta una proiezione degli autovettori in uno spazio mono- e bi-dimensionale, ed è stato inoltre possibile proiettare il moto da essi descritto sulla rispettiva struttura PDB di riferimento, permettendo di osservare i movimenti effettuati tramite il software grafico VMD (Humphrey et al., 1996).

4 Risultati e Conclusioni

4.1 Risultati del *read mapping*

I risultati relativi ai 74 campioni, costituiti da 148 file fastq, sono stati processati attraverso una pipeline di *trimming* da una pipeline di *read mapping*, come descritto in Materiali e Metodi; in Tabella 2 sono riportati il numero delle *read* sopravvissute alle procedure durante queste fasi di analisi.

Individuo	n° <i>read</i> fastq	n° <i>read</i> trimmed	n° <i>read</i> mappate
1112-36_S35	2508698	1769015	12717
1124-43_S75	1254135	901368	29858
1137-62_S71	2038897	1448746	16120
1201-1_S64	2098481	1489637	13329
1234-2_S33	2936885	2081404	14527
1303-1_S15	2911669	2067894	14820
1303-2_S2	1536252	1118079	54244
1315-16_S63	3464134	2436935	16600
1315-17_S54	1475091	1057643	13843
1319-22_S17	2024186	1440037	9834
1319-55_S65	1978685	1395679	35315
1320-24_S4	2193024	1537986	14010
1323-27_S66	2622137	1878363	11582
1323-28_S25	3542958	2482937	20525
1325-31_S7	1635032	1183592	12927
1328-38_S44	2344035	1683112	4319
1335-41_S43	2215906	1467658	4011
1346-42_S53	3120803	2181612	2573
1362-45_S34	2432941	1731009	19216
1370-16_S22	1484816	1065998	154007
1384-51_S46	1204693	868980	10148
1385-13_S56	2073070	1409928	77430
20150445_S10	11723	8490	53
21002840_S79	2718795	1939488	88380
21070877_S72	1345965	973281	10673
54985_S42	4647047	3285781	43012
55243_S3	2208572	1578359	51135
55376_S74	3496119	2469709	174191
57284_S52	4038702	2836492	17508
Baratti2_S45	3673470	2604679	23937
Baratti5_S11	3519895	2510284	25736
Billy_S27	2234762	1586330	13103
Cassius_S23	3463830	2457410	29728
Cerveteri_S76	2046648	1438760	18

Chiccosan_S37	2723335	1946764	17583
Ciri_S24	3836163	2709930	32216
Cocoon_S49	3052520	2215483	17312
Costier_S19	3724917	2633472	22896
Diana_S62	2481199	1760185	34179
Flamy_S57	3617190	2564625	25182
Fondi_S61	4826866	3227854	81700
Gea_S68	6240251	4390038	38930
Leucosia_S18	2323790	1656989	18299
Lilu_S5	2522565	1801608	26277
Macchia_S14	4299415	3015304	42543
Maite_S39	2766071	1961668	25322
Matilda_S6	3654219	2567919	22085
Molly_S48	2339002	1665388	16305
MOR04_S41	3904242	2603787	107500
MOR08_S55	3812507	2585782	47035
nido1_2006_S12	2847478	2041437	38962
nido3_2008_S13	2591681	1859270	14364
Onda_S67	3386435	2408528	28450
Pan_S9	2802010	2015655	19733
Perla_S29	2111036	1511205	13653
Polli_S36	3445714	2443910	26822
Rivadelsole1bis_S77	4395859	3118259	12
Rivadelsole1_S20	11314	9560	4
Rocchette03_S1	4124933	2900831	48496
Rocchette04_S59	4327566	2884012	55869
Salvita_S8	2265744	1621967	13862
Sampei_S58	3533597	2514878	23741
Schiaffo_S78	3471564	2455266	32407
Simba_S26	2909172	2065586	15056
S_Lucia_15_S69	4653522	3320519	17713
S_Lucia_3_S21	7106847	4919667	7226
Sofy_S32	2585873	1841582	14605
S_Severa_gialla1_S51	5718262	3968330	325
S_Severa_gialla2_S31	5310606	3710894	369
Tempesta_S47	1988318	1419104	13751
Tramia_S38	161778	112315	10520
Trovatella_S28	1511070	1089985	9954
Ventotene_S73	4077039	2687745	17402
Willy_S16	3358239	2378863	10244

Tabella 2: Elenco del numero di *read*: per ogni individuo sono riportati il numero di *read* che sono state sequenziate, che sono sopravvissute al *trimming* e che si sono allineate al genoma mitocondriale di riferimento

L'analisi del numero delle *read* ha messo in evidenza che, dei 74 campioni sequenziati, 6 campioni hanno presentato anomalie del numero di *read* sequenziate o allineate (*20150445-S10*, *Cerveteri-S76*, *Rivadelsole1bis-S77*, *Rivadelsole1-S20*, *S-Severa_gialla1-S51*, *S-Severa_gialla2-S31*) rivelando problematiche sorte durante la fase di sequenziamento dei campioni.

Una volta esclusi i suddetti campioni dalle analisi successive, i file prodotti dal *read mapping* sono stati analizzati mediante BCFtools in modo da selezionare tutte le varianti identificate sui 68 campioni, le quali sono state filtrate sulla base dei punteggi di qualità e profondità.

Posizione	Riferimento	Mutato	Gene
1631	ATT	AT	16S rRNA
2936	C	T	ND1
3045	C	T	ND1
4277	G	A	ND2
4671	G	A	ND2
5303	A	G	tRNA Cys
7433	C	T	COX2
7900	C	T	ATP8
8398	C	T	ATP6
8661	T	C	ATP6
9102	G	A	COX3
9440	T	A	COX3
9482	A	G	tRNA Gly
9610	T	C	ND3
9974	T	C	ND4L
10591	T	C	ND4
13287	G	A	ND5
13373	C	T	ND5
14373	G	A	CYTB
15562	T	C	D-loop
15645	C	T	D-loop
15731	A	G	D-loop
15800	G	A	D-loop
15970	A	G	D-loop

Tabella 3: Elenco di SNP e indels identificati

L’analisi ha portato all’identificazione di 24 varianti uniche, 23 SNP e 1 INDEL, riportate in Tabella 3: 5 varianti mappano sul D-loop (una struttura tipica del DNA mitocondriale in cui due filamenti di DNA sono separati da un terzo filamento di DNA), 2 varianti mappano su geni codificanti tRNA (Cys e Gly), la delezione è stata mappata nel gene dell’rRNA 16S e 16 varianti sono state mappate all’interno di geni codificanti. Inoltre, su tutti i campioni sequenziati sono state riscontrate le due varianti in posizione 1631 e 9974 rispetto al genoma di riferimento, già segnalate in precedenza.

Le varianti associate ai geni codificanti sono state successivamente annotate per associare alla mutazione l’effetto che ha sul prodotto proteico finale; lo *step* di annotazione è stato effettuato tramite il software SnpEff (Cingolani et al., 2012) nel quale abbiamo inserito l’annotazione di *Caretta caretta* depositata su RefSeq (Maglott et al., 2000), che ha portato all’identificazione di 6 varianti missenso riportate in Tabella 4.

Posizione Genoma	Gene	Nucleotide Riferimento	Nucleotide Mutato	Residuo Riferimento	Residuo Mutato	Posizione Proteina
4277	ND2	G	A	Ala	Thr	103
8398	ATP6	C	T	Pro	Ser	135
9102	COX3	G	A	Val	Met	142
9610	ND3	T	C	Leu	Ser	27
13287	ND5	G	A	Ala	Thr	487
14373	CYTB	G	A	Ala	Thr	53

Tabella 4: Elenco e annotazione delle varianti missenso

4.2 Risultati del modeling per omologia

Per lo studio dell’effetto delle varianti sulle subunità proteiche, sono state modellate le 6 subunità (ATP6, COX3, CYTB, ND2, ND3 e ND5), sui cui geni sono state mappate le varianti missenso.

I *template* sono stati selezionati dal PDB (Berman et al., 2000) tramite BLAST (Altschul et al., 1999) e per ciascuna sequenza *query* è stata selezionata la struttura del miglior *template* in base ai valori di *coverage* e *identity*. In tabella 5 sono riportati i codici PDB dei *template* utilizzati.

Proteina	PDB tempiato	Subunità	Descrizione
ATP6	6ZBB	a	ATP sintasi bovina
COX3	1V54	C	Citocromo C ossidasi bovina
CYTB	1BCC	C	Citocromo BC1 di Pollo
ND2	5LDW	N	Complesso I bovino
ND3	6G2J	A	Complesso I murino
ND5	6Q9B	D	Complesso I ovino

Tabella 5: Lista dei *template* selezionati dal PDB per l'*homology modeling*: sono riportati per ogni subunità, il PDB del *template* scelto e una breve descrizione

La modellazione è stata effettuata tramite *homology modeling* attraverso il *software* Modeller (Webb and Sali, 2016), utilizzando l’interfaccia grafica di UCSF Chimera (Pettersen et al., 2004).

Per ogni sequenza target sono state effettuate cinque *run* di Modeller, generando cinque differenti modelli di cui è stato selezionato il migliore in base ai valori di DOPE *score* che il programma genera per ogni struttura tridimensionale modellata. Il *Discrete Optimized Protein Energy* (DOPE) è un potenziale statistico che viene utilizzato per valutare la qualità del modello ottenuto tramite *homology modeling*, è una valutazione dell’energia del modello generato: un basso valore di DOPE *score* è associato ad un buon modello.

Per valutare l’effetto della mutazione sulle strutture, queste sono state introdotte tramite il software FoldX (Schymkowitz et al., 2005), un algoritmo in grado di determinare l’effetto di mutazioni puntiformi sulla stabilità della proteina. La mutazione viene introdotta utilizzando una libreria di rotameri che sono valutati attraverso l’esplorazione delle conformazioni con lo spazio tridimensionale circostante. Come valutazione dell’effetto della mutazione, FoldX calcola la differenza delle energie libere ($\Delta\Delta G$), valutata attraverso un *force field*, tra la struttura mutante e la struttura *wild-type*: un’energia di mutazione positiva è associata ad un effetto destabilizzante della mutazione sulla struttura tridimensionale mentre un’energia di mutazione negativa è associata ad un effetto stabilizzante della mutazione sulla struttura.

Inoltre, per valutare il grado di conservazione dei residui sulle proteine *wild-type*, le

sequenze delle proteine per cui sono state identificate le mutazioni missenso sono state allineate tramite BLAST sul *non-redundant protein sequences (nr) database*, selezionando le prime 5000 sequenze per *identity*, e riallineate tramite Clustal Omega (Slevers and Higgins, 2014) per ottenere un allineamento multiplo.

In figura 6 sono rappresentati tutti i modelli delle proteine *wild-type* mentre in Tabella 6 sono riportate le energie di mutazione calcolate da FoldX e il grado di conservazione dei residui ottenute dal multiallineamento.

Proteina	Mutazione	Convervazione	Energia Mutazione
ATP6	Pro135Ser	99.96%	1.15939
COX3	Val142Met	98.98%	-1.19221
CYTB	Ala53Thr	99.32%	0.573959
ND2	Ala103Thr	80.5%	3.04807
ND3	Leu27Ser	10.04%	0.002351
ND5	Ala487Thr	99.48%	1.2062

Tabella 6: Varianzione di energia associata alle mutazioni e conservazione dei residui

Tutte le mutazioni, oltre ad essere mappate in posizioni con residui molto conservati (tranne nel caso della subunità ND3 in quanto la mutazione cade su una regione di *loop*), presentano piccole differenze di energia libera in negativo o in positivo che suggeriscono uno scarso impatto sulla struttura tridimensionale o sulla stabilità intrinseca delle proteine. L'unica mutazione con energia anomala è quella mappata sulla proteina ND2, con un valore di $\Delta\Delta G$ pari a 3.04807 Kcal/mol: questo valore di energia libera suggerisce che la mutazione sulla subunità ND2 potrebbe avere un effetto destabilizzante, interferendo sulla struttura proteica tridimensionale.

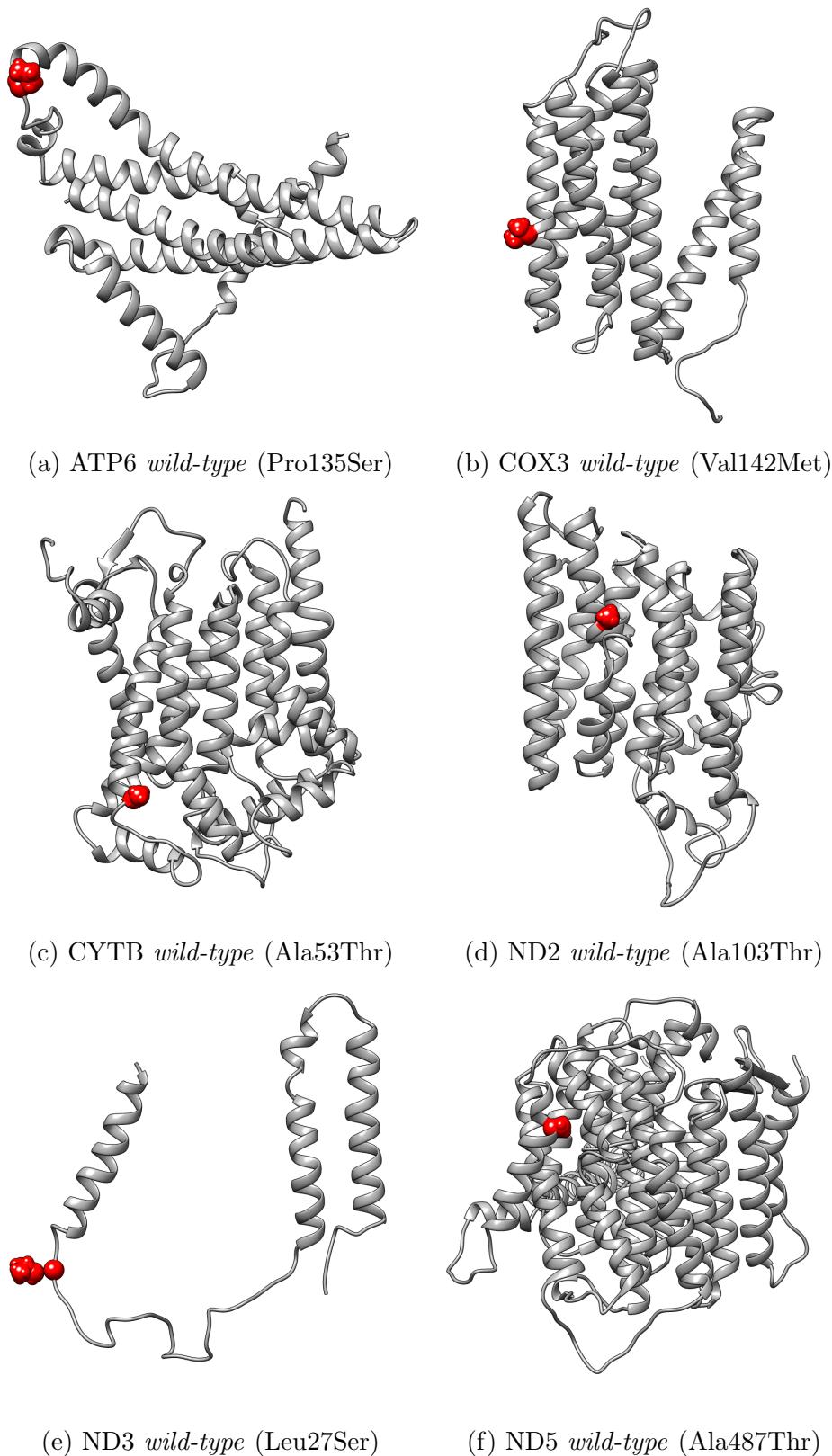


Figura 6: Rappresentazione grafica dei modelli tridimensionali delle proteine *wild-type* modellate. Per ogni struttura è stato evidenziato il residuo che subisce la mutazione e nelle didascalie sono riportate le mutazioni

4.3 Risultati della dinamica molecolare

4.3.1 Analisi del calcolo delle strutture secondarie

Poiché da un punto di vista strutturale e dinamico il solo FoldX non permette di ottenere informazioni dettagliate sugli effetti della mutazione, l'effetto della mutazione sulla subunità ND2 è stato analizzato mediante simulazioni di dinamica molecolare classica. In particolare, sono state effettuate due simulazioni di dinamica molecolare classica della subunità ND2 attraverso il *software* di dinamica molecolare GROMACS (Berendsen et al., 1995), una nella forma *wild-type* e una nella forma mutata.

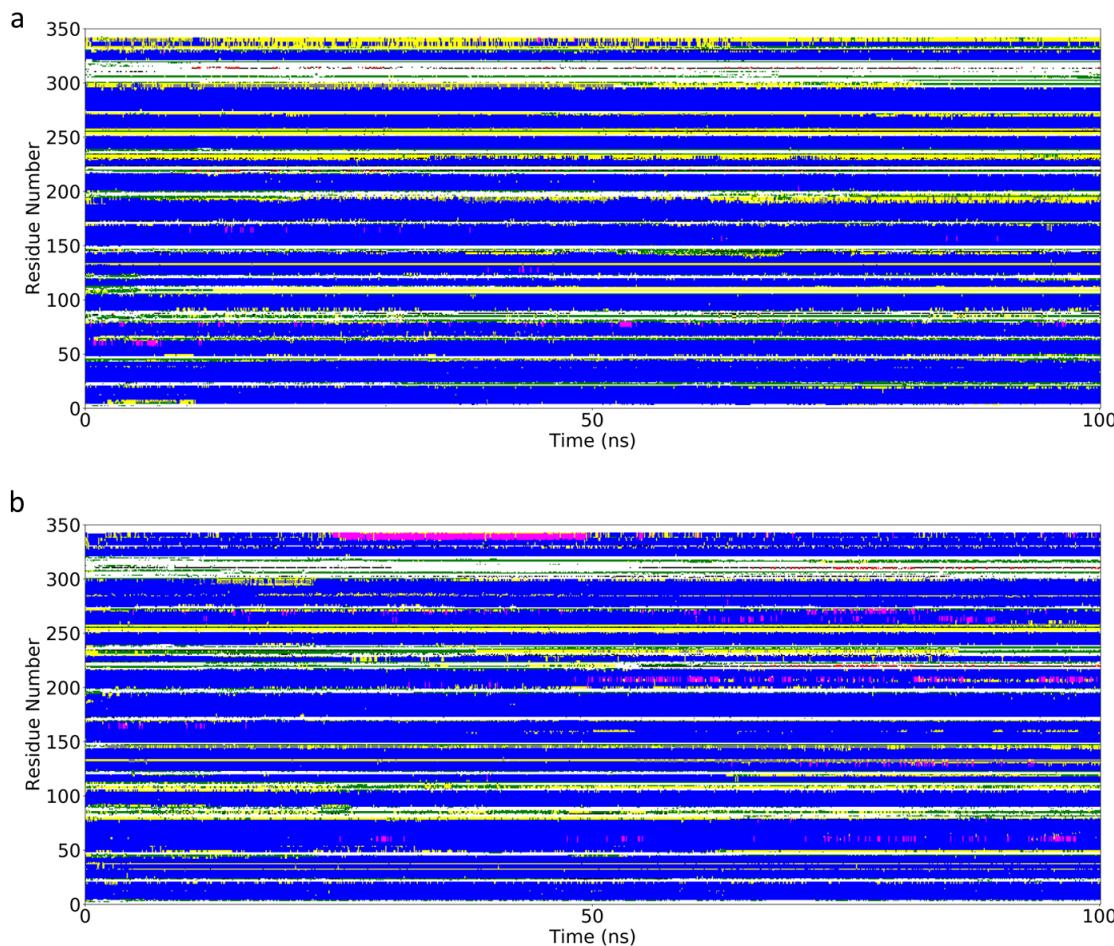


Figura 7: Grafici sull'evoluzione della struttura secondaria. In blu sono rappresentate le α -eliche, in giallo i *turn*, in magenta le eliche 3_{10} , in bianco i *coil* e in verde i *bend*

Per valutare l'effetto della mutazione sulla struttura tridimensionale, sono state analizzate e valutate le evoluzioni temporali delle strutture secondarie delle due proteine

nel corso dei 100 ns di simulazione. Questa analisi è stata effettuata tramite il *tool* *do_dssp* di GROMACS che effettua il calcolo della composizione delle strutture secondarie delle proteine analizzando il *pattern* dei legami a idrogeno per assegnare ad ogni residuo la corretta struttura secondaria. Tramite uno script in Python, che sfrutta le librerie di *matplotlib*, l'evoluzione della struttura secondaria è stata graficata in funzione del tempo di simulazione: ad ogni residuo è associato un colore a cui corrisponde una determinata struttura secondaria (Figura 7).

Dall'analisi del grafico relativo alla struttura mutata (Figura 7b), è possibile riscontrare poche differenze rispetto alla subunità *wild-type* (Figura 7a), le quali sono limitate principalmente alla porzione C-terminale e solamente per un tempo di simulazione limitato, coerentemente con dei riarrangiamenti che possono verificarsi nelle porzioni terminali di proteine che sono notoriamente più flessibili. Pertanto, da questa prima analisi emerge che la variante Ala103Thr di ND2 non sembra influenzare in maniera particolare la struttura tridimensionale della proteina.

4.3.2 Analisi delle componenti principali

Data l'assenza di effetti sulla struttura tridimensionale, per verificare se la mutazione potesse influenzare i moti della proteina è stata effettuata una *Principal Component Analysis* (PCA) sulle traiettorie delle due subunità.

Utilizzando il *tool* *gmx covar* del *software* GROMACS 2022 è stata generata una matrice di covarianza, che è poi stata trasformata in una matrice di cross-correlazione tramite uno *script* in Python: valori positivi indicano una correlazione positiva, ovvero i due residui presi in considerazione si muovono nella stessa direzione, mentre valori negativi rappresentano una correlazione negativa, ovvero i residui si stanno muovendo in direzione opposta.

Questa analisi (Figura 8) ha messo in evidenza che la mutazione sulla subunità ND2 contribuisce in maniera limitata all'alterazione di moti correlati e anti-correlati: tra la matrice di cross-correlazione *wild-type* (Figura 8a) e quella mutata (Figura 8b) si osservano solo piccole riorganizzazioni nel pattern dei moti correlati (punti marroni e verdi) e anti-correlati (punti grigi e celesti) in alcune regioni della proteina.

La diagonalizzazione delle matrici di covarianza ha generato, in entrambi i casi, un

totale di 1032 autovettori che vanno a descrivere tutti i moti all'interno del sistema. Il *tool* di diagonalizzazione della matrice di covarianza restituisce in *output* come indice di mobilità anche la *trace of the covariance matrix*, una misura della fluttuazione quadratica media della proteina. Le fluttuazioni (Tabella 7) indicano che la proteina mutata mostra una mobilità maggiore rispetto alla *wild-type*.

Proteina	Fluttuazione
ND2 <i>wild-type</i>	4.82567
ND2 mutato	8.28909

Tabella 7: *Trace of the covariance matrix*

Per confermare questo risultato, è stato proiettato il moto principale descritto dal primo autovettore di entrambe le subunità sulla struttura tridimensionale della proteina tramite il *tool gmx ana eig* che analizza gli autovettori ricavati dalla matrice di covarianza. Come *flag* per l'*input*, è stato utilizzato *-extr* che permette di calcolare due proiezioni estreme lungo una traiettoria sulla struttura media e interpola un numero di frame. in questo caso abbiamo interpolato 10 frame e abbiamo osservato il risultato tramite il software di grafica VMD (Figura 9).

Nonostante le fluttuazioni indichino una maggior mobilità della subunità mutata, l'analisi di dinamica essenziale ha permesso di osservare che questa subunità in realtà presenta una maggiore rigidità nella struttura rispetto al *wild-type* mostrando che la mobilità è confinata solo nella regione del *loop* esposto al solvente, al di fuori della membrana lipidica.

In conclusione, è possibile affermare che la struttura della subunità ND2 è riuscita ad assorbire l'effetto della mutazione Ala103Thr senza inficiare sulla funzionalità anche se con una lieve alterazione della sua mobilità.

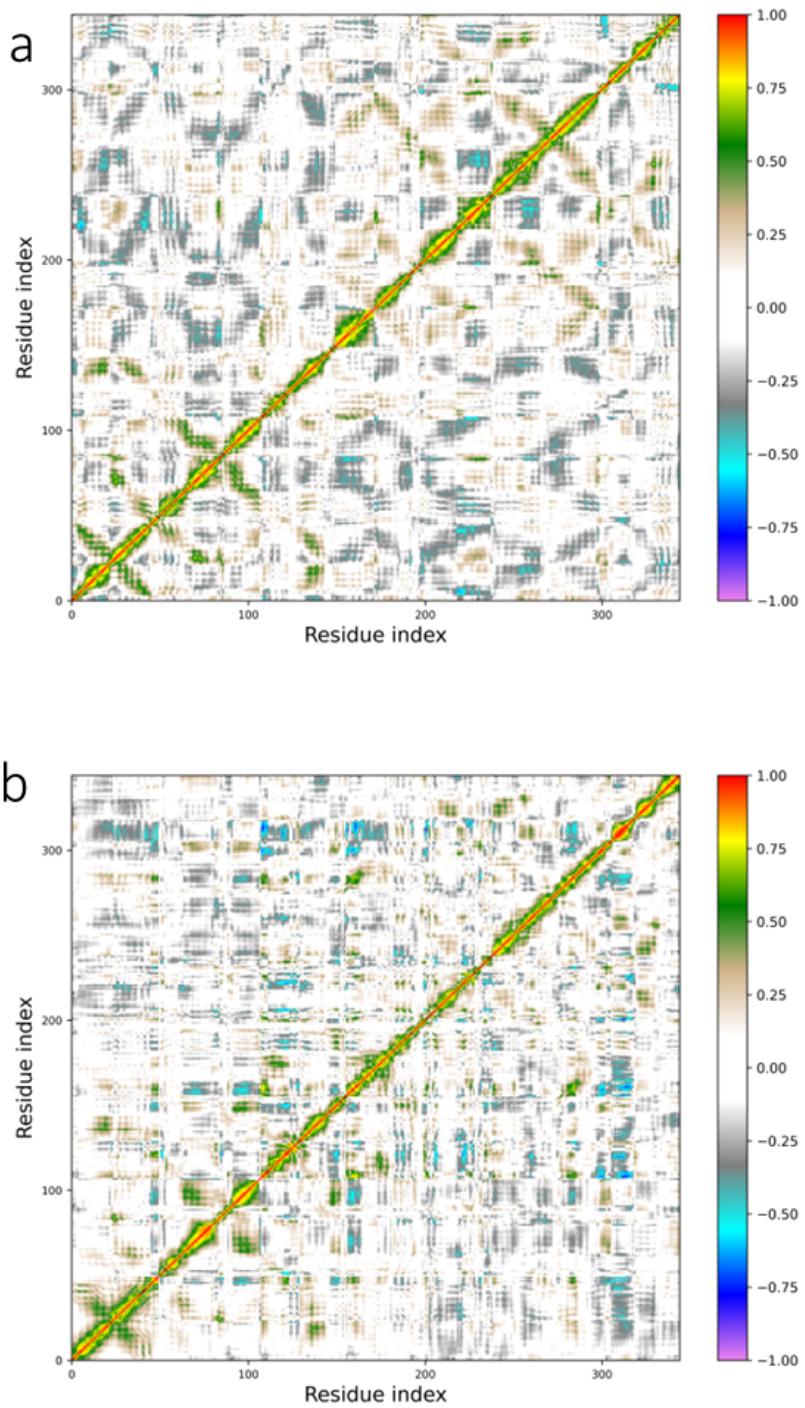


Figura 8: Matrici di cross-correlazione della subunità ND2 *wild-type* (a) e mutata (b). La correlazione di un residuo rispetto agli altri viene indicata dai punti colorati identificati dall’intersezione tra il numero dei residui in ordinata. Il codice dei valori della correlazione è indicato a destra del grafico.

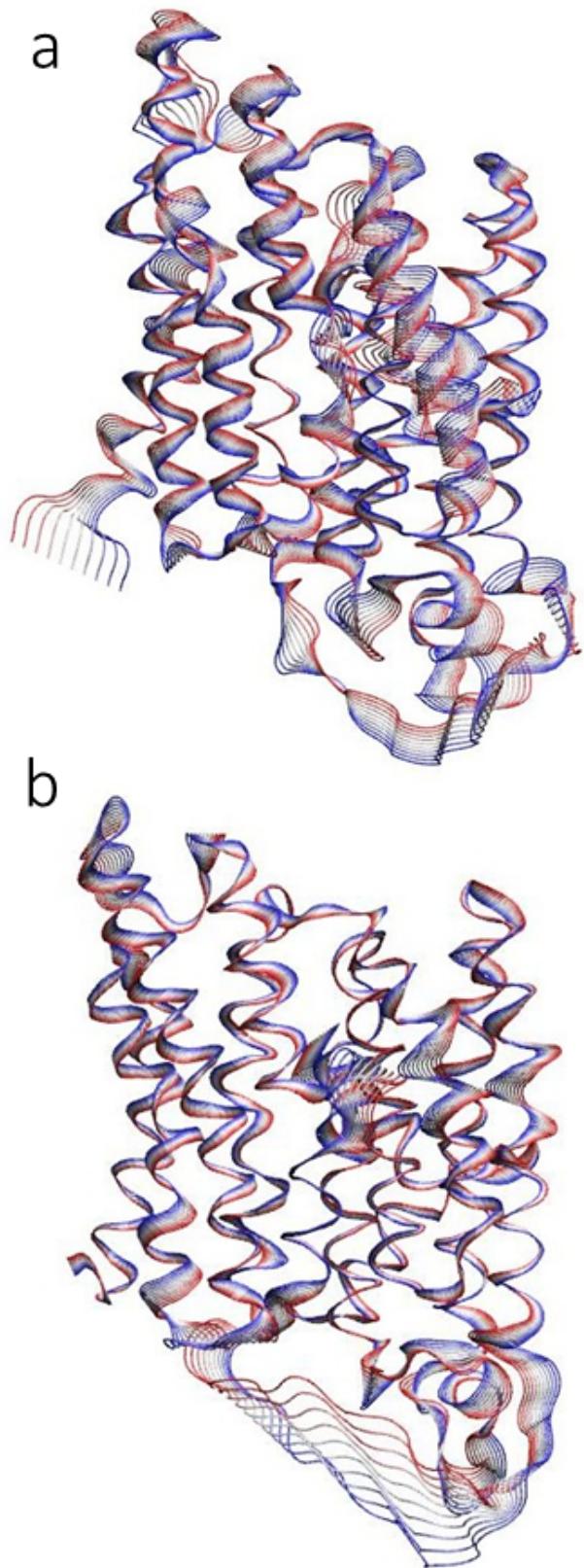


Figura 9: Proiezioni del moto principale descritto dal primo autovettore della PCA della subunità ND2 *wild-type* (a) e mutata (b)

5 Conclusioni

Nelle sezioni precedenti è stata descritta tutta la pipeline di analisi che ci ha permesso di condurre un'indagine sulle variazioni nel mitogenoma delle tartarughe marine del Mediterraneo e la successiva analisi dell'effetto delle varianti missenso sulla struttura proteica. Nello studio precedente, nel quale sono stati caratterizzati i polimorfismi identificati nei geni ND1 e ND3 di esemplari di *Caretta caretta* campionati nel Mediterraneo (Novelletto et al., 2016), l'analisi computazionale delle varianti osservate trasferite sui modelli tridimensionali ha portato all'osservazione che tutte le mutazioni intra-specifiche osservate non compromettono la funzionalità della struttura ma la maggior parte si trovava in prossimità dei confini della membrana in linea col fenomeno di adattamento omeoviscoso (Hazel, 1995), un meccanismo che coinvolge la ristrutturazione dello spessore, della fluidità e della composizione lipidica della membrana. Questo progetto di tesi ha avuto la finalità di identificare e studiare i polimorfismi nell'intero mitogenoma ottenuti attraverso l'analisi dei dati NGS e la successiva ricostruzione tridimensionale delle proteine su cui sono state mappate le varianti missenso.

In una prima fase sono stati ricostruiti i mitogenomi degli individui di *Caretta caretta* tramite la procedura di *read mapping* dei dati NGS ed ha portato all'individuazione di 24 varianti di cui:

- 5 mappate nel D loop
- 2 mappate in geni codificanti tRNA
- 1 mappata nel gene codificante rRNA 16S
- 16 mappate in geni codificanti

Queste varianti (riportate in Tabella 3), oltre a permetterci di rivelare nuovi nodi nella filogenesi intraspecifica e nella migrazione grazie all'analisi di polimorfismi presenti sul D-loop, sono state studiate al fine di correlare le mutazioni missenso ad effetti strutturali sul prodotto proteico. È stata quindi eseguita una modellazione strutturale per comprendere la significatività delle sostituzioni osservate a livello fenotipico. I risultati della modellazione tramite tecniche di *homology modeling* (Figura 6) e l'introduzione delle mutazioni attraverso il *software* Foldx (Tabella 6) suggeriscono che la maggior

parte delle mutazioni non compromettono la funzionalità della struttura, in quanto la variazione di energia libera tra la struttura *wild-type* e quella mutata è minima, mentre la mutazione Ala103Thr della subunità ND2, caratterizzata da un $\Delta\Delta G$ positivo, suggeriva che la mutazione potesse avere un effetto destabilizzante sulla struttura.

Per validare la predizione di Foldx su come la mutazione influisce sulla subunità ND2, sono state effettuate due simulazioni di dinamica molecolare classica della proteina nella forma *wild-type* e mutata. Il confronto delle due traiettorie (Figura 8) e il calcolo della *Principal Component Analysis* (Figura 9) hanno messo in evidenza che la mutazione ha scarsi effetti sulla struttura e sui moti interni della proteina, compatibilmente con la vita dell’animale.

In conclusione, non possiamo ricondurre le varianti missenso osservate ad un adattamento dovuto alle differenti condizioni climatiche come nel caso dell’adattamento omeoviscoso del lavoro del 2016, e possiamo inoltre affermare che non hanno un ruolo destabilizzante sulla funzionalità della proteina. Anche nel caso della subunità ND2, nonostante la mutazione sia non conservativa, la struttura riesce ad assorbire gli effetti della mutazione mantenendo la sua funzionalità.

6 Bibliografia

Allen, M. and Tildesley, D. (1989). Computer simulation of liquids (New York: Clarendon Press).

Altschul, S.F., Gish, W., Miller, W., Myers, E.W. and Lipman, D.J. (1990). Basic local alignment search tool. *Journal of molecular biology*, 215(3), 403–410.

Bagda E., Bardakci, F. and Turkozan., O. (2012). Lower genetic structuring in mitochondrial DNA than nuclear DNA among the nesting colonies of green turtles (*Chelonia mydas*) in the Mediterranean. *Biochem Syst Ecol* 43:192–199.

Berendsen, H.J.C., van der Spoel, D. and van Drunen, R. (1995). GROMACS: A message-passing parallel molecular dynamics implementation. *Comput. Phys. Commun.* 91, 43–56.

Berendsen, H.J.C., van der Spoel, D. and van Drunen, R. (1994). Gromacs- A message-passing parallel molecular-dynamics implementation. *Comput. Phys. Commun.* 91, 43-56.

Bolger, A.M., Lohse, M. and Usadel, B. (2014). Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics* (Oxford, England), 30(15), 2114–2120.

Brooks, B.R., Brooks, C., Mackerell, A.D., Nilsson, L., Petrella, R.J., Roux, B., Won, Y., Archontis, G., Bartels, C., Boresch, S. et al. (2009). CHARMM: Molecular dynamics simulation package. *J. Comput. Chem.* 30, 1545–1614.

Burrows, M., and Wheeler, D.J. (1994). A Block-sorting lossless data compression algorithm. Research Report

Buß, O., Rudat, J. and Ochsenreither, K. (2018). FoldX as Protein Engineering Tool: Better Than Random Based Approaches?. Computational and structural biotechnolo-

gy journal, 16, 25–33.

Case, D.A., Cheatham, T.E., Darden, T., Gohlke, H., Luo, R., Merz, K.M., Onufriev, A., Simmerling, C., Wang, B. and Woods, R.J. (2005). The Amber biomolecular simulation programs. *J. Comput. Chem.* 26, 1668–1688.

Chae Na, J., Kim, H., Park, H., Lecroq,T., Leonard, M., Mouchard, L. and Park, K. (2016). FM-index of alignment: A compressed index for similar strings. *Theoretical Computer Science*, 638, 159-170.

Chothia, C. (1992). One thousand families for the molecular biologist. *Nature* 357, 543–544.

Cingolani, P., Platts, A., Wang, I., Coon, M., Nguyen, T., Wang, L., Land, S.J., Lu, X. and Ruden, D.M. (2012). A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff: SNPs in the genome of *Drosophila melanogaster* strain w1118; iso-2; iso-3. *Fly*, 6(2), 80–92.

Cock, P.J., Fields, C.J., Goto, N., Heuer, M.L. and Rice, P.M. (2010). The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic acids research*, 38(6), 1767–1771.

Darden, T., York, D. and Pedersen, L. (1993). Particle mesh Ewald: An $N \log(N)$ method for Ewald sums in large systems. *J. Chem. Phys.* 98, 10089-10092.

Deng, H., Jia, Y. and Zhang, Y. (2018). Protein structure prediction. *Int. J. Mod. Physics. B* 32, 1840009.

Evans, D. and Holian, B. (1985). The Nose-Hoover Termostat. *J. Chem. Phys.* 83, 4069-4074.

Fiser, A., Do, R.K. and Sali, A. (2000). Modeling of loops in protein structures. *Protein Sci.* 9, 1753–1773.

Flicek, P. and Birney, E. (2009). Sense from sequence reads: methods for alignment and assembly. *Nature methods*, 6(11 Suppl), S6–S12.

Fox, N.K., Brenner, S.E. and Chandonia, J.M. (2014). SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Res.* 42, D304-9.

Hansmann, U.H.E. and Okamoto, Y. (1999). New Monte Carlo algorithms for protein folding. *Curr. Opin. Struct. Biol.* 9, 177–183.

Hawkes L.A., Broderick, A.C., Godfrey, M.H. and Godley B.J. (2007). Investigating the potential impacts of climate change on a marine turtle population. *Glob Change Biol* 13:1–10.

Hazel J. R. (1995). Thermal adaptation in biological membranes: is homeoviscous adaptation the explanation?. *Annual review of physiology*, 57, 19–42.

Horner, D.S., Pavesi, G., Castrignanò, T., De Meo, P.D., Liuni, S., Sammeth, M., Picardi, E. and Pesole, G. (2010). Bioinformatics approaches for genomics and post genomics applications of next-generation sequencing. *Briefings in bioinformatics*, 11(2), 181–197.

Huang, X., Feng, Q., Qian, Q., Zhao, Q., Wang, L., Wang, A., Guan, J., Fan, D., Weng, Q., Huang, T., Dong, G., Sang, T. and Han, B. (2009). High-throughput genotyping by whole-genome resequencing. *Genome research*, 19(6), 1068–1076.

Imelfort, M., Duran, C., Batley, J. and Edwards, D. (2009). Discovering genetic polymorphisms in next-generation sequencing data. *Plant biotechnology journal*, 7(4),

312–317.

Jones, D.T., Taylor, W.R. and Thornton, J.M. (1992). A new approach to protein fold recognition. *Nature* 358, 86–89.

Jorgensen, W.L. and Tirado-Rives, J. (1988). The OPLS [optimized potenzials for liquid simulations] potential function for proteins, energy minimizations for crystals of cyclic peptides and crambin. *J. Am. Chem. Soc.* 110, 1657–1666.

Jorgensen, W.L., Chandrasekhar, J., Madura, J.D., Impey, R.W. and Klein, M.L. (1983). Comparison of Simple Potential Functions for Simulating Liquid Water. *J. Chem. Phys.* 79, 926–935.

Kandt, C., Ash, W.L. and Tieleman, D.P. (2007). Setting up and running molecular dynamics simulations of membrane proteins. *Methods* (San Diego, Calif.), 41(4), 475–488.

Khor, B.Y., Tye, G.J., Lim, T.S. and Choong, Y.S. (2015). General overview on structure prediction of twilight-zone proteins. *Theor. Biol. Med. Model.* 12, 15.

Krief, M. and Ashkenazy, Y. (2021). Calculation of elastic constants of embedded-atom-model potentials in the NVT ensemble. *Physical review. E*, 103(6-1), 063307

Langmead, B. and Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nature methods*, 9(4), 357–359.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R. and 1000 Genome Project Data Processing Subgroup (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics* (Oxford, England), 25(16), 2078–2079.

Long, K., Cai, L. and He, L. (2018). DNA Sequencing Data Analysis. Methods in molecular biology (Clifton, N.J.), 1754, 1–13.

Maglott, D.R., Katz, K.S., Sicotte, H. and Pruitt, K.D. (2000). NCBI's LocusLink and RefSeq. *Nucleic acids research*, 28(1), 126–128.

Mahoney, M.W. and Jorgensen, W.L. (2001). Diffusion constant of the TIP5P model of liquid water. *J. Chem. Phys.* 114, 363.

Mardis E. R. (2008). Next-generation DNA sequencing methods. *Annual review of genomics and human genetics*, 9, 387–402.

Mazaris A.D., Kornarali, E., Matsinos, Y.G. and Margaritoulis D. (2004). Modeling the effect of sea surface temperature on sea turtle nesting activities by investigating seasonal trends. *Nat Res Model* 17:445–465.

McKenna, A., Hanna, M., Banks, E., Sivachenko, A., Cibulskis, K., Kernytsky, A., Garimella, K., Altshuler, D., Gabriel, S., Daly, M. and DePristo, M.A. (2010). The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome research*, 20(9), 1297–1303.

Milne, I., Stephen, G., Bayer, M., Cock, P.J.A., Pritchard, L., Cardle, L., Shaw, P.D. and Marshall D. (2013). Using Tablet for visual exploration of second-generation sequencing data. *Briefings in Bioinformatics*, 14(2), 193-202

Mooney S. (2005). Bioinformatics approaches and resources for single nucleotide polymorphism functional analysis. *Briefings in bioinformatics*, 6(1), 44–56.

Muhammed, M.T. and Aki-Yalcin, E. (2019). Homology modeling in drug discovery: Overview, current applications, and future perspectives. *Chem. Biol. Drug Des.* 93, 12–20.

Muzzey, D., Evans, E. A. and Lieber, C. (2015). Understanding the Basics of NGS: From Mechanism to Variant Calling. *Current genetic medicine reports*, 3(4), 158–165.

Novelletto, A., Testa, L., Iacobelli, F., Blasi, P., Garofalo, L., Mingozzi, T. and Falconi, M. (2016). Polymorphism in Mitochondrial Coding Regions of Mediterranean Loggerhead Turtles: Evolutionary Relevance and Structural Effects. *Physiological and biochemical zoology : PBZ*, 89(6), 473–486.

Nwanochie, E. and Uversky, V.N. (2019). Structure Determination by Single-Particle Cryo-Electron Microscopy: Only the Sky (and Intrinsic Disorder) is the Limit. *Int. J. Mol. Sci.* 20, 4186.

Oosternbrink, C., Villa, A., Mark, A.E. and van Gunsteren, W.F. (2004). A biomolecular force field based on the free enthalpy of hydration and solvation: the GROMOS force-field parameter sets 53A5 and 53A6. *J. Comput. Chem.* 25, 1656-1676.

Pearson, K. (1901). On Lines and Planes of Closest Fit to Systems of Points in Space. *Philos. Mag.* 2, 559-572.

Pettersen, E.F., Goddard, T.D., Huang, C.C., Couch, G.S., Greenblatt, D.M., Meng, E.C. and Ferrin, T.E. (2004). UCSF Chimera - A visualization system for exploratory research and analysis. *J. Comput. Chem.* 25, 1605–1612.

Phillips, J.C., Braun, R., Wang, W., Gumbart, J., Tajkhorshid, E., Villa, E., Chipot, C., Skeel, R.D., Kalé, L. and Schulten, K. (2005). Scalable molecular dynamics with NAMD. *J. Comput. Chem.* 26, 1781–1802.

Quigley, D. and Probert, M.I.J. (2004). Langevin dynamics in constant pressure extended system. *J. Chem. Phys.*, 120, 11432.

Rebbeck, T. R., Spitz, M. and Wu, X. (2004). Assessing the function of genetic variants in candidate gene association studies. *Nature reviews. Genetics*, 5(8), 589–597.

Šali, A. and Blundell, T.L. (1993). Comparative Protein Modelling by Satisfaction of Spatial Restraints. *J. Mol. Biol.* 234, 779–815.

Schymkowitz, J., Borg, J., Stricher, F., Nys, R., Rousseau, F. and Serrano, L. (2005). The FoldX web server: an online force field. *Nucleic acids research*, 33(Web Server issue), W382–W388.

Shendure, J. and Ji, H. (2008). Next-generation DNA sequencing. *Nature biotechnology*, 26(10), 1135–1145.

Sievers, F. and Higgins, D. G. (2014). Clustal Omega, accurate alignment of very large numbers of sequences. *Methods in molecular biology* (Clifton, N.J.), 1079, 105–116.

Smith, D.R., Quinlan, A.R., Peckham, H.E., Makowsky, K., Tao, W., Woolf, B., Shen, L., Donahue, W.F., Tusneem, N., Stromberg, M.P., Stewart, D.A., Zhang, L., Ranade, S.S., Warner, J.B., Lee, C.C., Coleman, B.E., Zhang, Z., McLaughlin, S.F., Malek, J.A., Sorenson, J.M. and Richardson, P.M. (2008). Rapid whole-genome mutational profiling using next-generation sequencing technologies. *Genome research*, 18(10), 1638–1642.

Somero G.N. (2010). The physiology of climate change: how potentials for acclimatization and genetic adaptation will determine 'winners' and 'losers'. *The Journal of experimental biology*, 213(6), 912–920.

Stratton M. (2008). Genome resequencing and genetic variation. *Nature biotechnology*, 26(1), 65–66.

Sunyaev, S., Hanke, J., Brett, D., Aydin, A., Zastrow, I., Lathe, W., Bork, P. and Reich, J. (2000). Individual variation in protein-coding sequences of human genome.

Advances in protein chemistry, 54, 409–437.

Sunyaev, S., Lathe, W. and Bork, P. (2001). Integration of genome data and protein structures: prediction of protein folds, protein interactions and "molecular phenotypes" of single nucleotide polymorphisms. *Current opinion in structural biology*, 11(1), 125–130.

Van der Auwera, G.A., Carneiro, M.O., Hartl, C., Poplin, R., Del Angel, G., Levy-Moonshine, A., Jordan, T., Shakir, K., Roazen, D., Thibault, J., Banks, E., Garimella, K.V., Altshuler, D., Gabriel, S. and DePristo, M.A. (2013). From FastQ data to high confidence variant calls: the Genome Analysis Toolkit best practices pipeline. *Current protocols in bioinformatics*, 43(1110), 11.10.1–11.10.33.

Van Gunsteren, W.F. and Berendsen, H.J.C (1988). A Leap-Frog Algorithm for Stochastic Dynamics. *Mol. Simul.*, 1, 173-185.

Verlet, L. (1967). Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules. *Phys. Rev.* 159, 98–103.

Wallqvist, A. and Mountain, R.D. (1999). Molecular Models of Water: Derivation and Description. In *Review in Computational Chemistry*, pp.183-247.

Wang, Z. and Moult, J. (2001). SNPs, protein structure, and disease. *Human mutation*, 17(4), 263–270.

Webb, B. and Sali, A. (2016). Comparative Protein Structure Modeling Using MO-DELLER. *Curr. Protoc. Bioinforma.* 54, 5.6.1-5.6.37.

Weiner, S.J., Kollman, P.A., Singh, U.C., Case, D.A., Ghio, C., Alagona, G., Profeta, S. and Weiner, P. (1984). A New Force Field for Molecular Mechanical Simulation of Nucleic Acids and Proteins. *J. Am. Chem. Soc.* 106, 765–784.