Introduzione

Il *DNA computing* è una delle questioni che sta suscitando più interesse nell'ambito della ricerca bio-informatica negli ultimi anni. L'idea base è quella di utilizzare la tecnologia e il formalismo dell'informatica per riprodurre e simulare i fenomeni biologici in modo da catalogare e analizzare i risultati ottenuti. I problemi che molto spesso si verificano, in questo ambito, sono dovuti alla potenza e alla complessità di calcolo che i fenomeni biologici producono. Tra essi lo splicing è uno dei meccanismi di ricombinazione più studiati. È stato introdotto nel 1987 da *Tom Head*, che definì un nuovo modo di usare la teoria dei linguaggi formali per analizzare l'azione ricombinante del DNA. Successivamente altri ricercatori hanno osservato questo fenomeno e hanno fornito definizioni e proprietà alla base di questo lavoro di tesi. In breve, l'azione combinata degli enzimi di restrizione e degli enzimi di ligasi consente alle molecole di DNA di essere *tagliate* e *riassociate* per produrre ulteriori molecole. Quindi si può considerare un linguaggio I associato alle molecole di DNA e un linguaggio di regole R che formalizzano il comportamento ricombinante consentito da specifiche classi di attività enzimatiche. Il nuovo formalismo generativo viene chiamato *sistema splicing*, e il linguaggio ottenuto a partire da I per azione iterata di R e via via sulle stringhe è chiamato *linguaggio splicing*. Poiché il DNA è presente anche in forma circolare, in letteratura è stata

data una definizione di sistemi splicing circolari, in cui l'insieme iniziale è un insieme di parole circolari, ossia classi di equivalenza rispetto alla ben nota relazione di coniugazione tra parole. In breve, un sistema splicing circolare è una tripla (A, I, R) dove *A* è un alfabeto finito, *I* è il linguaggio circolare *iniziale* e *R* è l'insieme delle regole splicing. Anche se la letteratura è vasta sullo studio del potere computazionale dei sistemi splicing circolari, per lo sviluppo del progetto ci siamo basati sulla definizione di sistemi splicing circolari fatta, in ambito più generale da *Paun* (1996) che è il modello attualmente più studiato.

Lo scopo di questo lavoro di tirocinio e tesi è stato di analizzare e modellare i sistemi splicing circolari, attraverso il linguaggio di programmazione *Java* e utilizzando la piattaforma *Eclipse Europa* come ambiente di sviluppo. Oggetto della nostra implementazione è stata la classe dei *sistemi splicing circolari-(1,3)* ossia sistemi in cui le regole hanno una particolare struttura. Per far ciò si è implementato il modello *(1-3)-CSSH*, per poi estenderci in maniera naturale ai sistemi (1,3). La motivazione alla base della necessità di un' implementazione dei sistemi splicing circolari sta nella possibilità di osservare ed evidenziare anche le problematiche che le parole circolari producono a livello gestionale e computazionale. Queste problematiche sono trattate superficialmente in questa tesi, poiché nell'implementazione ci siamo serviti di un pacchetto già sviluppato ad hoc per la gestione delle parole circolari. I sistemi che noi abbiamo trattato per la nostra implementazione, invece, richiedevano una maggiore attenzione per la gestione, l'applicazione e il controllo delle regole di splicing. Il progetto sviluppato segue la definizione di linguaggio generato: quindi, dato un sistema splicing circolare del tipo *S = (A, I, R)* occorre applicare tutte le regole presenti nell'insieme *R* a

tutte le possibili coppie di stringhe che possiamo ricavare dall'insieme *I*, comprese le coppie riflessive, e unire le stringhe risultanti all'insieme *I* in modo da avere un nuovo insieme *I* per l'iterazione successiva. Dato che questa operazione è teoricamente infinita, il nostro sistema splicing comprende anche il numero di iterazioni da effettuare. Per osservare gli effetti dell'operazione splicing e quindi l'evoluzione del sistema, l'applicazione, per ogni iterazione, manda in output l'insieme delle parole circolari che sono state generate. L'implementazione ha richiesto l'uso di alcune strutture dati complesse del tipo : *Array*, *Lista concatenata*, *ArrayList*, ritenute adatte ma non ottime per l'implementazione svolta.

Si è anche ha analizzato il fatto che questo progetto con pochi adattamenti è utilizzabile pure per sistemi del tipo *(1,4)*, *(2,3)* e *(2,4)* (detti *one-sided*). Inoltre ha posto le basi per progetti successivi riguardanti la generalizzazione ad un qualsiasi sistema splicing circolare finito e una fase di ottimizzazione del codice, evitando il ricalcolo di parole già generate.

Infine analizzando i tempi di risposta del sistema sviluppato, si ipotizza un lavoro successivo per la ricerca di strutture dati che ottimizzino la gestione delle parole circolari. Tale questione non si presenta semplice poiché non si lavora su un insieme fissato di parole, ma su un insieme che viene creato dinamicamente dal sistema.

Il documento di tesi è organizzato come segue.

Dopo una breve descrizione del fenomeno dello splicing da un punto di vista biologico (Cap. 1), nel Cap. 2 vengono fornite tutte le nozioni di base di linguaggi per comprendere le definizioni date nel Cap. 3 sui

sistemi splicing lineari e circolari. L'implementazione del progetto è presentata nel Cap. 5 insieme ad una descrizione delle strutture dati usate e alla simulazione, mentre il Cap. 4 descrive il pacchetto per gestire le parole circolari.