



UNIVERSITÀ DEGLI STUDI DI MILANO - BICOCCA

Scuola di Scienze

Dipartimento di Informatica, Sistemistica e Comunicazione

Corso di laurea in Informatica

# Rilevazione di eventi di Alternative Splicing a partire da read paired-end

**Relatore:** Prof. Della Vedova Gianluca

**Correlatore:** Prof. Rizzi Raffaella

**Relazione della prova finale di:**

Francesco Porto

Matricola 816042

**Anno Accademico 2018-2019**

## **Abstract - Italiano**

L'Alternative Splicing è un meccanismo attraverso il quale diverse isoforme proteiche sono generate a partire da uno stesso gene. Si stima che oltre il 75% dei geni umani utilizzi l' Alternative Splicing, e una piena comprensione di questo meccanismo potrebbe aiutare a far luce su diversi fenomeni biologici non ancora del tutto chiari, oltre che a migliorare la capacità di rilevazione di diverse patologie di natura genetica. Con l'avvento delle tecnologie NGS (Next Generation Sequencing), l'accesso a grandi quantità di informazioni di natura biologica è diventato sempre più facile e conveniente: in questo contesto l'informatica potrebbe giocare un ruolo fondamentale nello studio dell'Alternative Splicing. Purtroppo al momento non esistono molti software in grado di rilevare eventi di Alternative Splicing a partire da read paired-end, un nuovo tipo di read ottenute da sequenziatori NGS, che potrebbero portare ad una maggiore precisione in fase di rilevazione. Si è quindi deciso di estendere ASGAL, un software sviluppato dall'Algolab in grado di rilevare eventi di Alternative Splicing, per supportare le read paired-end. In questo documento saranno evidenziate le principali modifiche apportate ad ASGAL, ponendo l'attenzione sulle differenze tra il formato single-end e quello paired-end. Sarà poi presentato un esempio di funzionamento, oltre che ad alcuni possibili sviluppi futuri.

## **Abstract - English**

Alternative Splicing is a mechanism by which different protein isoforms are produced starting from the same gene. It is estimated that over 75% of human genes use Alternative Splicing, and a full comprehension of such mechanism could help shed light on different biological phenomena which are not fully understood yet, and also to improve the ability to detect genetic diseases. With the advent of NGS (Next Generation Sequencing) technologies, access to biological data has become easier and cheaper: in this context computer science could play a key role in the study of Alternative Splicing. Unfortunately, at the moment there are not many softwares capable of detecting Alternative Splicing events starting from paired-end reads, a new format of reads obtained by NGS sequencers, which could be used to increase precision during detection. We have decided to extend ASGAL, a software developed by Algolab capable of detecting Alternative Splicing events, to support paired-end reads. In this paper the main changes applied to ASGAL will be highlighted, focusing our attention on differences between single-end and paired-end formats. An example of operation will be presented, as well as some future prospects.

# Contents

<b>1</b>	<b>Introduzione</b>	<b>1</b>
1.1	ASGAL . . . . .	1
1.2	Alternative Splicing . . . . .	2
1.3	Paired-End Reads . . . . .	3
<b>2</b>	<b>Splice-Aware Aligner</b>	<b>4</b>
2.1	Cos'è e come funziona . . . . .	4
2.2	Allineamento di entrambe le read . . . . .	5
2.3	Introduzione di read unmapped e "placeholder" nel formato MEM . . . .	5
2.4	Supporto alle fragment library types . . . . .	7
<b>3</b>	<b>Formattatore SAM</b>	<b>9</b>
3.1	Cos'è e come funziona . . . . .	9
3.2	Formato SAM . . . . .	10
3.3	Modifiche alla computazione del campo FLAG . . . . .	11
3.4	Calcolo dei campi RNEXT, PNEXT e TLEN . . . . .	12
3.5	Calcolo delle statistiche dell'allineamento . . . . .	13
<b>4</b>	<b>Rilevatore di Eventi di Alternative Splicing</b>	<b>14</b>
4.1	Cos'è e come funziona . . . . .	14
4.2	Fusione degli introni dedotti dai due sample . . . . .	15
4.3	Calcolo dell' IDMP (Inner Distance between Mate Pairs) . . . . .	16
4.4	Calcolo del TIDMP (Transcript-based IDMP) . . . . .	17
4.5	Possibile utilizzo di IDMP e TIDMP . . . . .	18
<b>5</b>	<b>Esempio di funzionamento</b>	<b>19</b>
5.1	Generazione delle read . . . . .	19
5.2	Utilizzo . . . . .	20
5.3	Risultati . . . . .	21
<b>6</b>	<b>Competenze acquisite</b>	<b>22</b>
<b>7</b>	<b>Conclusioni</b>	<b>23</b>

# 1 Introduzione

## 1.1 ASGAL

ASGAL (Alternative Splicing Graph Aligner) [1] è un tool per l'identificazione di eventi di Alternative Splicing espressi in un campione di RNA-seq a partire da un'annotazione di un gene. ASGAL si compone di tre step:

- **Costruzione dello splicing graph:** a partire dall'annotazione di un gene, ASGAL costruisce uno splicing graph, ovvero una struttura a grafo che rappresenta tutti i trascritti noti del gene in input.
- **Allineamento Splice-Aware:** ASGAL allinea le read di RNA-Seq con lo splicing graph del gene in input. L'allineamento è Splice-Aware in quanto è necessario tenere traccia della posizione di esoni ed introni per un corretto allineamento.
- **Rilevamento degli eventi di Alternative Splicing:** gli allineamenti prodotti dallo step precedente sono analizzati per rilevare gli eventi di alternative splicing indotti dalle read del campione.

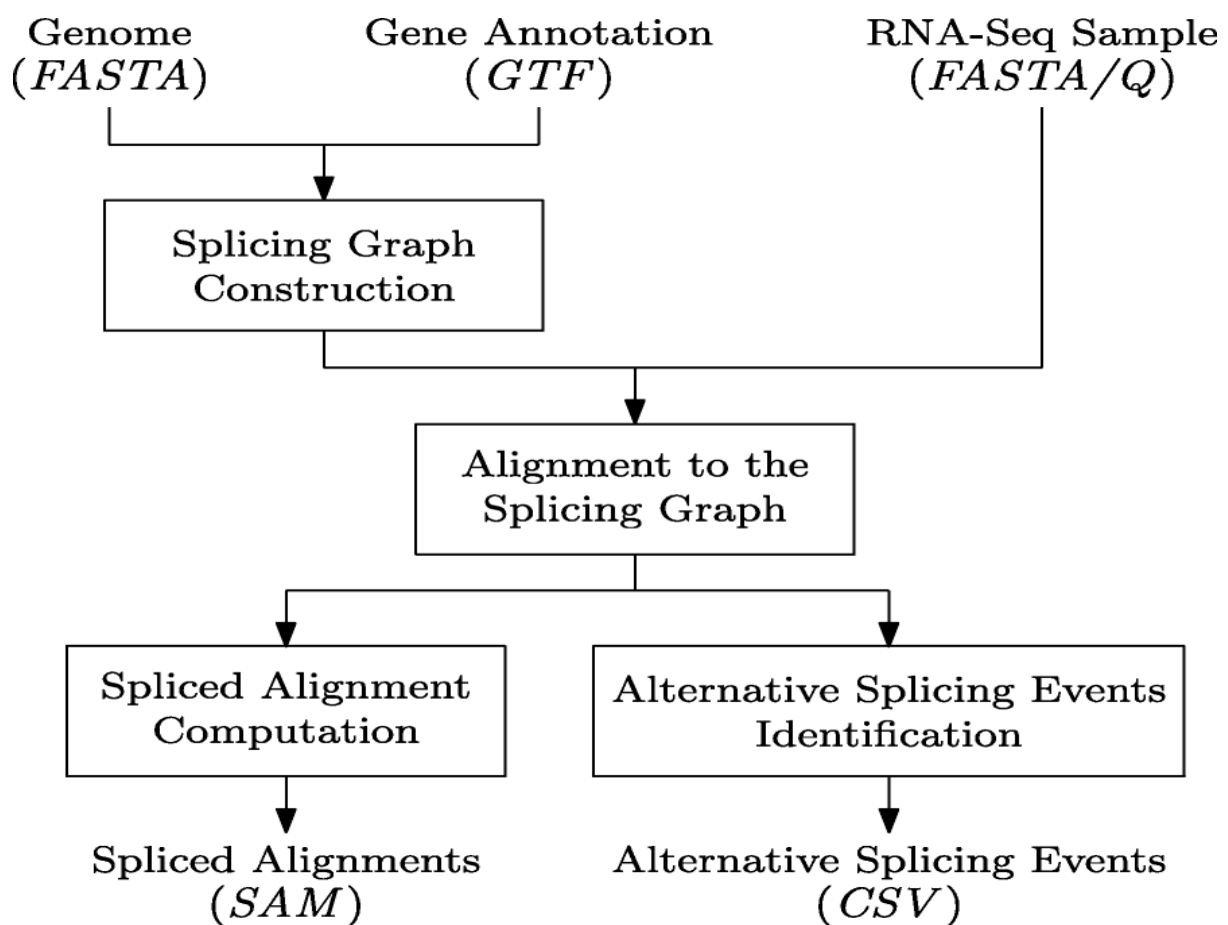


Figure 1: La pipeline di ASGAL illustrata

## 1.2 Alternative Splicing

L'Alternative Splicing è un metodo utilizzato dalle cellule per produrre proteine diverse dallo stesso frammento di DNA che viene utilizzato da oltre il 75% dei geni umani.

Considerando un generico locus, esso può essere diviso in esoni (parti codificanti) e introni (parti non codificanti). Durante la fase di Trascrizione gli introni vengono rimossi e la Timina viene trasformata in Uracile, ottenendo pre-RNA. A questo punto, in un normale processo di Splicing, tutti gli esoni vengono utilizzati, nell'ordine in cui appaiono nel pre-RNA, per ottenere una proteina. Nel caso di un evento di Alternative Splicing, questo non accade: alcuni esoni potrebbero infatti non essere utilizzati, o apparire in un ordine diverso.

Vengono riconosciuti 5 tipi di eventi di Alternative Splicing:

1. **Exon Skipping:** Almeno un esone non appare nel trascritto
2. **Mutually Exclusive Exons:** Almeno due esoni non compaiono mai in uno stesso trascritto
3. **Alternative 5' Donor Site:** Parte di un introne nel 5' diventa un esone
4. **Alternative 3' Acceptor Site:** Parte di un introne nel 3' diventa un esone
5. **Intron Retention:** Parte di un esone diventa un introne

ASGAL è in grado di rilevarli tutti tranne il caso 2.

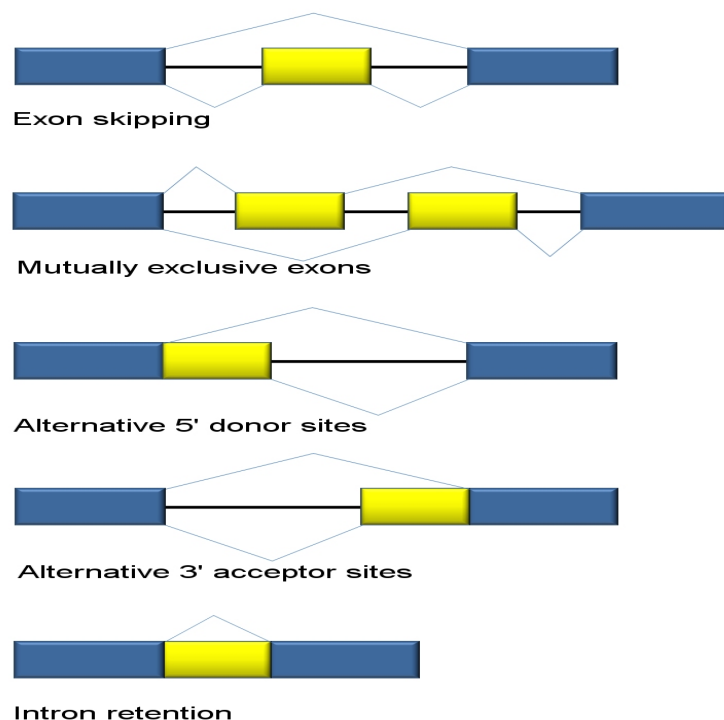


Figure 2: I diversi tipi di Alternative Splicing

### 1.3 Paired-End Reads

Le paired-end reads consistono nell'estrazione di due letture da un singolo frammento di DNA (generalmente le due estremità), contrariamente alle single-end reads che ne estraggono solo una. Sono prodotte da sistemi NGS, e la loro preparazione è molto semplice: una volta stabilita la grandezza della singola lettura, viene estratta la lettura sull'estremità sinistra, il campione viene girato, e viene estratta nuovamente l'estremità sinistra (ottenendo quindi l'estremità destra). Viene inoltre fornita la distanza tra le due letture, che permette di disambiguare alcuni casi di allineamento.

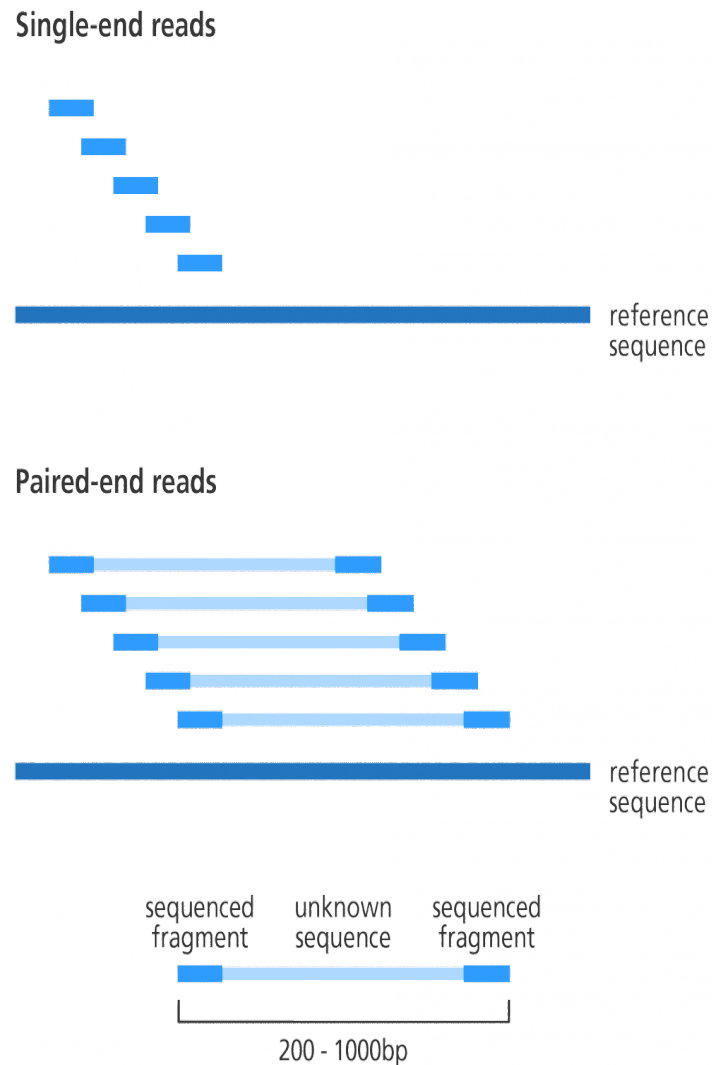


Figure 3: Esempi di read single-end e paired-end

## 2 Splice-Aware Aligner

### 2.1 Cos'è e come funziona

Lo Splice-Aware Aligner svolge due compiti:

1. Generazione dello Splicing Graph e della sua Linearizzazione
2. Allineamento delle read di RNA-Seq alla Linearizzazione dello Splicing Graph

L'allineamento avviene utilizzando il concetto di MEM (Maximum Exact Matching); l'output verrà convertito in formato SAM per permettere l'eventuale elaborazione con altri strumenti.

## 2.2 Allineamento di entrambe le read

Il primo problema da affrontare è ovviamente il fatto che sia ora necessario allineare due read e non una; fortunatamente si tratta solo di iterare il processo di allineamento su una coppia di read ad ogni ciclo, anziché su read singola. Verranno quindi generati due file contenti MEM anziché uno. Sarà poi compito della Formattazione SAM "fondere" i due file MEM per ottenere un SAM Completo.

## 2.3 Introduzione di read unmapped e "placeholder" nel formato MEM

Nei file MEM ottenuti dallo Splice-Aware Aligner vengono ora visualizzati due nuovi tipi di MEM: quelli relativi alle read unmapped e quelli relativi ai "placeholder". Il primo caso è banale, e rappresenta tutte quelle read che non hanno un matching esatto di lunghezza considerevole con il genoma dato in input.

Il secondo caso è più complesso e rappresenta un insieme di read fasulle utilizzate solo come padding per avere due file MEM della stessa lunghezza: questo facilita enormemente l'elaborazione nello step successivo (la formattazione SAM). Come detto in precedenza quando si lavora con read paired-end è sempre necessario lavorare a coppie, ma non sempre ad uno stesso pair è associato lo stesso numero di allineamenti secondari: è qui che entrano in gioco i "placeholder".

La loro implementazione è banale: si tengono due contatori (che rappresentano rispettivamente il numero di allineamenti relativi alla prima read e quelli relativi alla seconda read), si ottengono separatamente gli allineamenti relativi a ciascuna delle due read, e si controllano i contatori. Si prende il minore dei due e si aggiungono tanti placeholder quanto bastano per rendere uguali i contatori.



```

MAPPED + ENST00000623047_e_4_21_6632291 0 (5940,1,53) (8956,53,48)
GGATTGCCCCATCGCATATCTGGAGTTCGGGGTCTTAGAAAGCTTTCTTGCCCTATTTCTTTAGCAGAATGAGTGTCTGACATTTCCCAGGACTGTTTT
UNMAPPED ENST00000623047_e_90_21_6638170
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTATGAGCCACTGTACTGCGCTGTGCCTACTTCAAAGGACTGAAAATAAAAAATAAATA
PLACEHOLDER ENST00000623047_e_4_21_6632291

```

Figure 4: I diversi tipi di MEM

---

**Algorithm 1** Algoritmo per l'aggiunta dei placeholder

---

```

1: procedure ADDPLACEHOLDERS(count1, count2, file1, file2)
2:   while count1! = count2 do
3:     if count1 < count2 then
4:       addPlaceholder(file1)
5:       count1 ++
6:     else
7:       addPlaceholder(file2)
8:       count2 ++
9:     end if
10:  end while
11: end procedure

```

---

## 2.4 Supporto alle fragment library types

Ci sono diversi protocolli per la preparazione di librerie paired-end, che portano a read con caratteristiche diverse. Le **fragment library types** permettono di descrivere queste caratteristiche in modo sintetico. Le caratteristiche che possono essere descritte sono:

- **Orientamento relativo di una read rispetto all'altra:** può essere inward (I) o outward (O)
- **Se è noto o meno lo strand di appartenenza delle due read:** può essere stranded (S) o unstranded (U)
- **La direzionalità della prima read, solo nel caso stranded:** può essere first (F) o reverse (R)

La seguente immagine descrive tutte le possibili combinazioni:

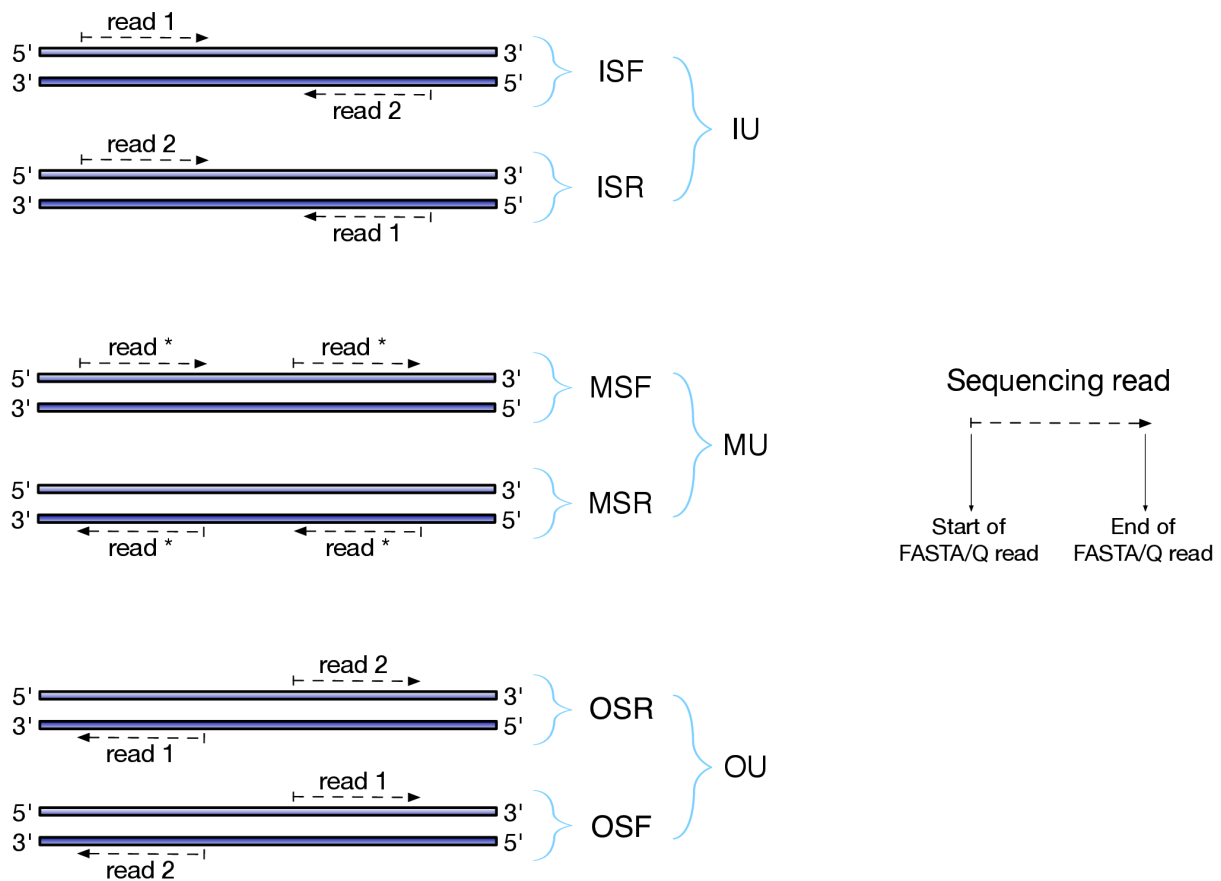


Figure 5: I diversi tipi di FTL

ASGAL utilizza queste informazioni per velocizzare il processo di allineamento. Nel formato single-end questa informazioni non esiste, quindi ogni read veniva allineata in entrambe le direzioni, e si prendeva l'allineamento migliore dei due.

Nel formato paired-end, si può utilizzare la ftl per ridurre il numero di allineamenti necessari, fino al 50% nel caso stranded. Ad esempio, se viene fornita un ftl di tipo ISF è sufficiente allineare la read 1 sullo strand + (ignorando lo strand -) e la read 2 sullo strand - (ignorando lo strand +).

Il caso unstranded richiede un po' più di attenzione: non è infatti possibile sapere a priori su quale strand allineare la prima read. Per risolvere questo problema è stata riutilizzata la vecchia procedura di allineamento della prima read in entrambe le direzioni; una volta trovata la direzione migliore per la prima, la seconda viene di conseguenza.

Supponiamo ad esempio di avere una libreria in formato MU: se la prima read allinea sullo strand +, di conseguenza anche la seconda sarà allineata sullo strand +; viceversa, se la prima read allinea sullo strand -, anche la seconda allinea sempre sullo strand -. In questo caso si ottiene solo una riduzione del 25% nel numero degli allineamenti.

Qualora il tipo di libreria non venga fornito dall'utente, rimane necessario provare ad allineare le read in entrambe le direzioni, senza alcun incremento di prestazioni.

## 3 Formattatore SAM

### 3.1 Cos'è e come funziona

Come già detto, gli allineamenti ottenuti dallo Splice-Aware aligner sono in un formato non standard chiamato MEM (nota: l'acronimo MEM si riferisce sia al formato dell'output che all'estensione del file ottenuto). L'obiettivo di questa seconda parte è quello di convertire i due file MEM in un singolo file SAM (Sequence Alignment Map), il formato standard per memorizzare gli allineamenti.

Non si tratta solo di una semplice conversione, in quanto è necessario indurre diverse informazioni aggiuntive per avere un file SAM standard, quali: la posizione di inizio dell'allineamento sulla genomica, la stringa CIGAR, i flag relativi all'allineamento, ecc. Per supportare le read paired-end è stato necessario modificare gran parte di queste funzionalità.

In questa sezione saranno descritte le principali modifiche apportate al Formattatore SAM, oltre ad alcune funzionalità aggiuntive utili per la rilevazione di eventi di Alternative Splicing.

## 3.2 Formato SAM

Il formato SAM (Sequence Alignment Map) è composto da 11 campi:

1. QNAME: Il nome identificativo dell'allineamento
2. FLAG: Una serie di flag binari che identificano le caratteristiche dell'allineamento
3. RNAME: Identificativo del gene di riferimento
4. POS: Posizione 1-based di inizio dell'allineamento sul genoma
5. MAPQ: Qualità dell'allineamento
6. CIGAR: Stringa che identifica le operazioni effettuate per ottenere l'allineamento
7. RNEXT: QNAME del mate (solo paired-end)
8. PNEXT: POS del mate (solo paired-end)
9. TLEN: Distanza tra mate (solo paired-end)
10. SEQ: Allineamento vero e proprio
11. QUAL: Valore Phread della read

```
@HD VN:1.4
@SQ SN:21 LN:46709983
ENST00000623047_e_0_21_6631916 97 21 6631917 255 100M = 6632004 187
TTCATCAAGGTTCTCCCCGAAGAATGATGCTACCCAGGCCAGGGCACACAGTGACCCGTGCACAGGATGCACTGAGCACACACGGCACTGGGTGAACCA * NM:i:0
ENST00000623047_e_0_21_6631916 145 21 6632004 255 100M = 6631917 -187
ACTGGGTGAACCAAGGAACAGAAGGAGAAGCCAGCCTGGGTCTGCAAAATATACTTTGCAGGAAAAGCAGGTAAAATGGAAAGGTCACAATTCAGCAGCAA * NM:i:0
ENST00000623047_e_1_21_6636195 81 21 6636196 255 100M = 6636276 180
CATCCACAAAGTCATAATGGGAAGCACGTTTCTTCTCACTAAGCTGTTCTTTGAGAACACAAAGAATGGGGGAATTTCTTTAAATATAGCTATTTCAGG * NM:i:0
ENST00000623047_e_1_21_6636195 161 21 6636276 255 100M = 6636196 -180
TAAATATAGCTATTTCCAGGATTACCTTCACCCACAACCTGTTCTTTCTTAGACATCTCTTTTCATTTGTCAGTTTCTGAGTTGTATTTTATAATAAAG * NM:i:0
ENST00000623047_e_2_21_6631410 81 21 6631411 255 100M = 6631500 189
AGCAGCTGTTAAAGCTGGGTGACTGGTCTGTTATACTTTTTTGTATTGTGTATGTTTTAGTGATCTGTAATAAAACACTTGTACAAAATGACAAAGTT * NM:i:0
ENST00000623047_e_2_21_6631410 161 21 6631500 255 100M = 6631411 -189
ATGACAAAATTTTATCTACACTTAGCTCTTAAGGCTTTGGTTACCTTTGGGAAGGGGAAAGTGTCAGGGGCATGAACAAATCTGATTCTTAGATACACAAG * NM:i:0
ENST00000623047_e_3_21_6634811 81 21 6634812 255 100M = 6634908 196
TATCACATCAGTTATTCTTTTTTCAAAACAGAGAACTTCTGATAGTGAGTATCTCTGTTCCATATGCTGTCATCTGGGTGTTTGAGGGTAACGCTAAGTT * NM:i:0
ENST00000623047_e_3_21_6634811 161 21 6634908 255 100M = 6634812 -196
```

Figure 6: Un esempio di file SAM prodotto da ASGAL

### 3.3 Modifiche alla computazione del campo FLAG

Il campo FLAG è il secondo del formato SAM e consiste di un valore numerico (ottenuto convertendo in decimale una serie di flag binari) che rappresenta le caratteristiche dell'allineamento preso in esame. La seguente immagine mostra il significato di ciascun bit del flag:

Binary (Decimal)	Hex	Description
00000000001 (1)	0x1	Is the read paired?
00000000010 (2)	0x2	Are both reads in a pair mapped “properly” (i.e., in the correct orientation with respect to one another)?
00000000100 (4)	0x4	Is the read itself unmapped?
00000001000 (8)	0x8	Is the mate read unmapped?
00000010000 (16)	0x10	Has the read been mapped to the reverse strand?
00000100000 (32)	0x20	Has the mate read been mapped to the reverse strand?
00001000000 (64)	0x40	Is the read the first read in a pair?
00010000000 (128)	0x80	Is the read the second read in a pair?
00100000000 (256)	0x100	Is the alignment not primary? (A read with split matches may have multiple primary alignment records.)
01000000000 (512)	0x200	Does the read fail platform/vendor quality checks?
10000000000 (1024)	0x400	Is the read a PCR or optical duplicate?

Figure 7: Il significato di ciascun bit del campo FLAG

Nei casi single-end solo due flag vengono utilizzati: quello relativo allo strand (0x16) e quello relativo al tipo di allineamento (0x100); visto che le read non sono paired, il flag 0x1 sarà sempre false, quindi tutti i flag risultanti saranno pari.

Nei casi paired-end tutti i flag vengono utilizzati. E' inoltre necessario trattare gli allineamenti a coppie, in quanto il campo FLAG esprime informazioni anche sul mate e non solo sull'allineamento preso in esame.

Supponiamo ad esempio di avere due read, la prima che mappa sullo strand positivo e la seconda che non mappa (ed è quindi *unmapped*). Sarà innanzitutto necessario mettere a true il flag relativo alle read paired-end (0x1) per entrambe le read. Considerando la prima, sarà messo a true il flag relativo al mate unmapped (0x8) e il flag relativo al first-in-pair (0x4). Considerando la seconda, sarà messo a true il flag relativo alla read unmapped (0x4) e quello relativo al second-in-pair (0x80). I flag in decimale saranno quindi 73 e 133.

Si noti che per il momento non viene tenuto conto dei flag 0x200 e 0x400, ma questo non è di alcuna rilevanza al fine di identificare eventi di Alternative Splicing.

### 3.4 Calcolo dei campi RNEXT, PNEXT e TLEN

I campi TLEN, RNEXT e PNEXT rappresentano rispettivamente il settimo, l'ottavo e il nono campo di ogni record del formato SAM; essi sono praticamente inutilizzati quando si allineano read single-end, ma nel formato paired-end assumono maggiore importanza. In particolare i campi RNEXT e PNEXT sono utilizzati da strumenti di visualizzazione degli allineamenti (come ad esempio IGV) per permettere una corretta visualizzazione di una read e del suo mate.

Il campo RNEXT contiene il nome dell'allineamento relativo al mate (ovvero il suo campo PNAME). Per semplicità, quando i due allineamenti sono consecutivi, si può lasciare il suo valore a '='.

Il campo PNEXT contiene la posizione iniziale 1-based dell'allineamento relativo al mate (ovvero il suo campo POS. Qualora il mate fosse unmapped, si utilizza il valore 0.

Il campo TLEN rappresenta la distanza la lunghezza del template osservato, ovvero la distanza (sul genoma) tra l'inizio della prima read e la fine della seconda. Per la sua computazione è sufficiente trovare la posizione finale del secondo allineamento e sottrarre la posizione iniziale del primo.

La seguente immagine mostra un esempio di allineamento visualizzato da IGV:

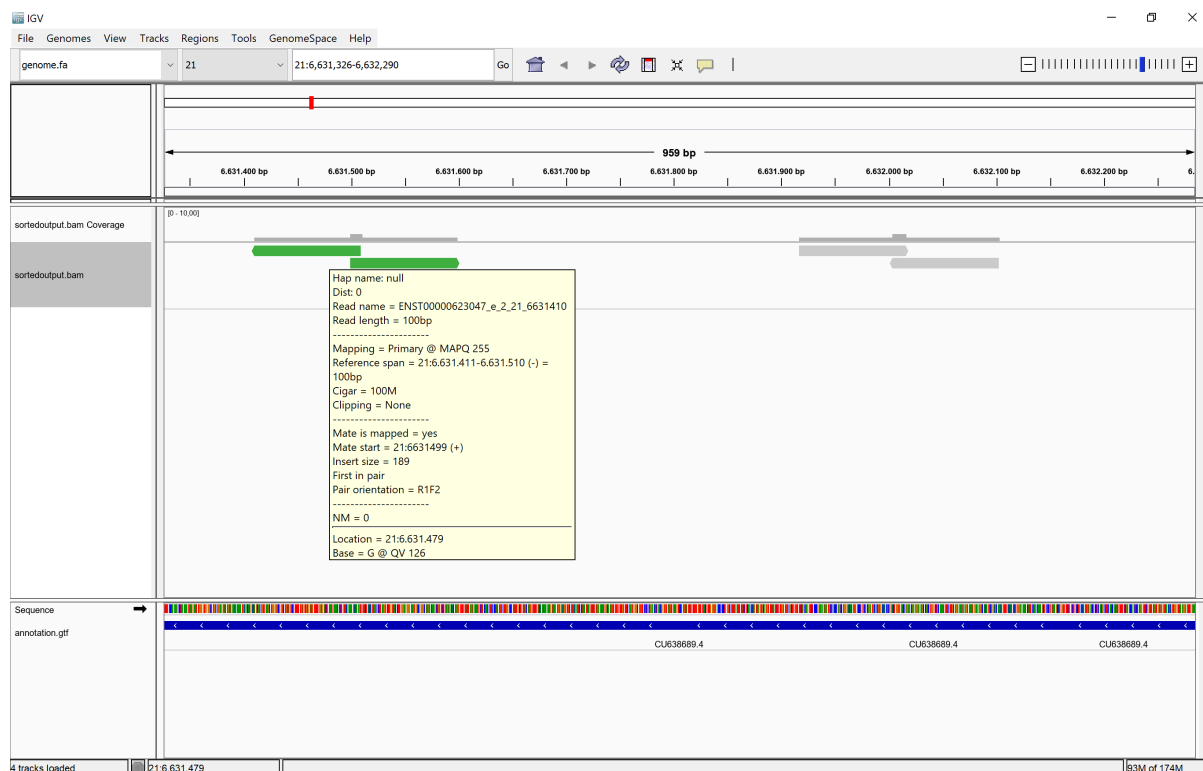


Figure 8: Informazioni su mate da SAM formato correttamente

E' importante notare che, se questi campi sono settati correttamente, lasciando il cursore del mouse su un allineamento vengono visualizzate tutte le informazioni relative al mate. Al contrario, se si prova ad indicizzare il file BAM (la versione binaria del formato SAM) per l'utilizzo con IGV, e questi campi non sono stati settati correttamente, verrà visualizzato un errore. Un esempio errore è il seguente: se il campo FLAG non contiene 0x8 (quindi il mate è mapped), e si inserisce un valore di PNEXT diverso da 0, al momento dell'indicizzazione sarà visualizzato il messaggio "mapped mate cannot have zero coordinate; treated as unmapped".

### 3.5 Calcolo delle statistiche dell'allineamento

Seguendo l'esempio di altri allineatori (come ad esempio STAR), è stato deciso di visualizzare alcune statistiche in fase di allineamento, quali:

- Numero di read mappate e non mappate
- Numero di allineamenti primari e secondari
- IDMP
- tIDMP

Questi valori sono visualizzati nel file *.alignsinfo.txt*. Anche se non hanno finalità particolari per la rilevazione di eventi di Alternative Splicing, essi forniscono uno strumento per valutare la qualità degli allineamenti effettuati da ASGAL.



## 4 Rilevatore di Eventi di Alternative Splicing

### 4.1 Cos'è e come funziona

## 4.2 Fusione degli introni dedotti dai due sample

Come già detto in precedenza, i nuovi introni rilevati dagli allineamenti vengono estratti dal file MEM e confrontati con quelli già noti al fine di rilevare eventi di Alternative Splicing. In questo senso, la versione paired rileva i nuovi introni da ciascuno dei due file MEM, e li fonde.

### 4.3 Calcolo dell' IDMP (Inner Distance between Mate Pairs)

Considerando una coppia di read, si definisce IDMP (Inner Distance between Mate Pairs) la distanza sul genoma di riferimento in termini di BP (Base Pair) tra l'ultima base della prima read e la prima della seconda. Questa informazione viene generalmente fornita dall'ente che ha effettuato il sequenziamento, e può essere confrontata con l'IDMP rilevato durante l'allineamento per rilevare nuovi eventi di Alternative Splicing.

Visto che un allineamento può essere rappresentato da più di un MEM, non è possibile semplicemente aggiungere la lunghezza dell'allineamento alla sua posizione iniziale. Prima di poter calcolare l'IDMP è quindi necessario introdurre il concetto di BitVector, ovvero una sequenza di bit che rappresenta la posizione degli esoni nella genomica. Un BitVector è dotato di due operazioni:

- Rank: data una posizione, ritorna l'esone di provenienza
- Select: dato un esone, ritorna la posizione di partenza

Queste due operazioni permettono di calcolare l'IDMP in maniera efficace. Innanzitutto si prende l'ultimo MEM relativo all'allineamento della prima read, e si utilizza l'operazione di Rank per trovare l'esone di appartenenza. A questo punto, si utilizza l'operazione di Rank per trovare la posizione iniziale dell'esone. L'offset sarà quindi dato dalla differenza tra il MEM e la posizione iniziale dell'esone. Basta quindi aggiungere questo offset alla posizione iniziale per trovare la fine del primo allineamento.

L'inizio del secondo allineamento è noto. A questo punto basterà sottrarre ad esso la fine di quello precedente, in modo da ottenere l'IDMP.

Il seguente algoritmo riassume la procedura:

## 4.4 Calcolo del TIDMP (Transcript-based IDMP)

Per TIDMP si intende la misura della distanza *sui trascritti* tra le due read. Per calcolarlo è innanzitutto necessario ottenere l'ultimo MEM relativo alla prima read e il primo MEM relativo alla seconda. Da ciascuno di essi è possibile ottenere la posizione iniziale e la lunghezza sul bitvector. A questo punto, usando l'operazione di rank, è possibile capire l'esone di provenienza di ciascun MEM.

Se i due MEM si trovano sullo stesso esone, il TIDMP è dato semplicemente dalla distanza tra la fine del primo MEM e l'inizio del secondo (sarà quindi uguale all' IDMP calcolato precedentemente).

Se i due MEM non si trovano sullo stesso esone, potrebbero essere su due esoni consecutivi o meno. Nel caso di esoni consecutivi, il TIDMP è dato dalla somma tra il suffisso non coperto dal primo esone e il prefisso non coperto dal secondo esone.

Il seguente algoritmo mostra come calcolare il TIDMP:

Il caso di esoni non consecutivi presenta diverse criticità, e per il momento non è stato calcolato.

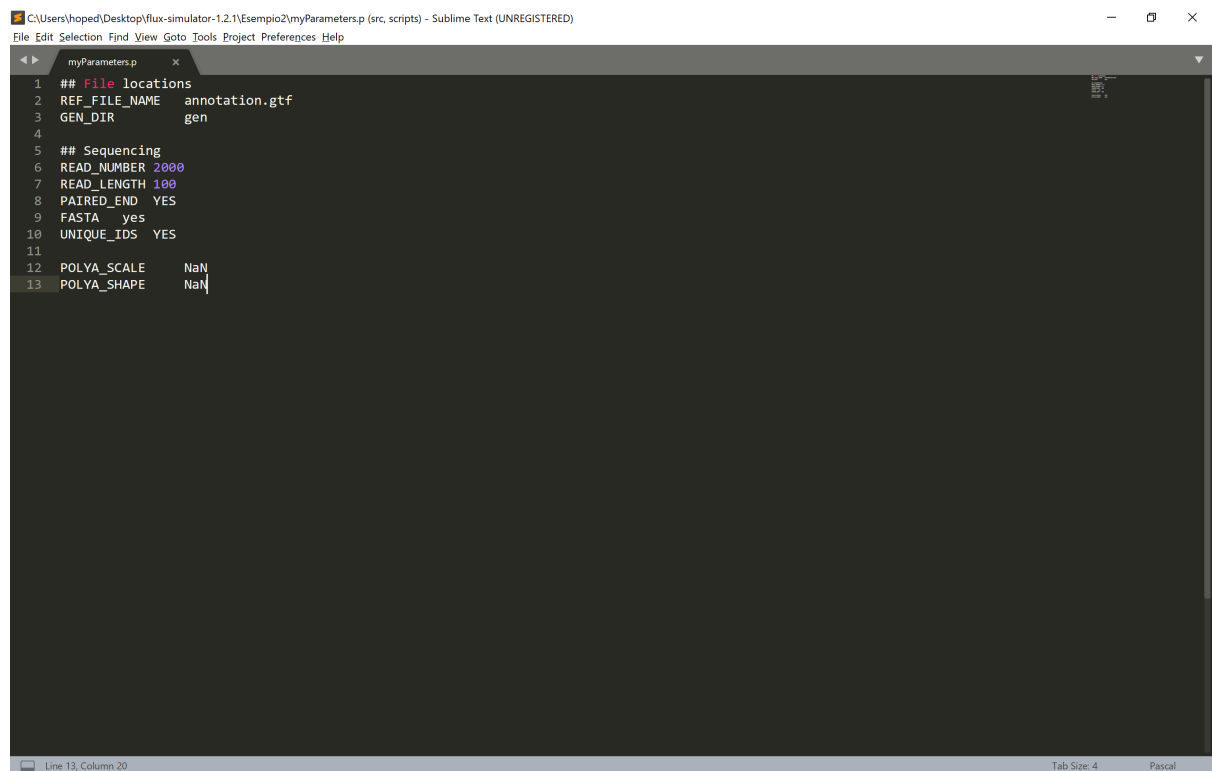
## 4.5 Possibile utilizzo di IDMP e TIDMP

## 5 Esempio di funzionamento

In questo esempio di funzionamento si utilizzerà il gene ENSG00000280145, situato nel cromosoma 21 dell'uomo sapiens (GRCh38 / hg38). E' stato prima scaricato il genoma di riferimento da ensembl in formato fasta e la relativa annotazione in formato gtf. Dal file gtf (contenente l'annotazione per l'intero genoma) è stata isolata l'annotazione relativa al gene ENSG00000280145.

### 5.1 Generazione delle read

Si è scelto di utilizzare Flux Simulator per la generazione delle read paired-end. Il suo utilizzo non è particolarmente complicato, ma è necessario passare i diversi parametri attraverso un file con estensione .p. Il file utilizzato in questa simulazione è il seguente:



```
1 ## File locations
2 REF_FILE_NAME  annotation.gtf
3 GEN_DIR        gen
4
5 ## Sequencing
6 READ_NUMBER    2000
7 READ_LENGTH    100
8 PAIRED_END     YES
9 FASTA          yes
10 UNIQUE_IDS     YES
11
12 POLYA_SCALE    NaN
13 POLYA_SHAPE    NaN
```

Figure 9: Il file contenente i parametri di Flux Simulator

Vengono così generati due file contenenti 2000 read di lunghezza 100, in formato fasta, che saranno dati in input ad ASGAL.

## 5.2 Utilizzo

ASGAL viene eseguito via linea di comando, richiamando lo script principale usando come parametri:

- Il genoma di riferimento (opzione -g)
- L'annotazione del genoma (opzione -a)
- I due file contenuti read (opzioni -s e -s2)
- La cartella di destinazione dell'output (opzione -o)
- L'indicazione delle read paired-end (opzione -paired)
- La fragment library type (opzione -f), opzionale per velocizzare la fase di allineamento

Questo script richiama nell'ordine lo Splice-Aware Aligner, il Formattatore SAM e il Rilevatore di eventi di Alternative Splicing, visualizzando alcune informazioni sul funzionamento.

Questa immagine mostra il funzionamento di ASGAL:

Sebbene sia possibile eseguire ciascuno script singolarmente, si raccomanda di usare lo script principale per un utilizzo più immediato.

### **5.3 Risultati**

Sono stati rilevati i seguenti eventi di Alternative Splicing:



## 6 Competenze acquisite

Durante lo svolgimento dello stage sono state acquisite le seguenti competenze:

- Utilizzo di dati biologici in formati diversi (fasta, gtf, SAM, ecc.) e creazione di algoritmi che li manipolano
- Utilizzo di strumenti di natura bioinformatica: è stato utilizzato SAMTools per la validazione dei file SAM, IGV per la visualizzazione degli allineamenti sul genoma, gli allineatori BWA, STAR e BMap come riferimento per lo Splice-Aware Aligner, RNASeqSim e Flux Simulator per la generazione di read
- Utilizzo di community di esperti di bioinformatica: è stato usato Biostars per chiarire alcuni dubbi inerenti al funzionamento di SAMTools, il thread si trova a questo indirizzo: <https://www.biostars.org/p/376192/>
- Approfondimento dei linguaggi di programmazione Python e C++ e utilizzo di librerie specifiche per la bioinformatica (come ad esempio kseq per il parsing di file fasta)
- Utilizzo avanzato di Linux e di strumenti di environment management specifici per la bioinformatica (Bioconda)
- Utilizzo di strumenti di version control: è stato utilizzato Github, la relativa repo si trova a questo indirizzo: <https://github.com/HopedWall/galg>
- Lavoro di gruppo: è stato utilizzato il software Slack per comunicare con i membri dell'Algolab

## 7 Conclusioni

ASGAL è ora in grado di supportare le read paired-end, rilevando correttamente gli eventi di Alternative Splicing e salvando gli allineamenti rispettando lo standard del formato SAM. Rimane da investigare l'utilizzo di IDMP e tIDMP per aumentare l'efficacia della rilevazione, oltre al possibile utilizzo delle read paired-end per rilevare gli eventi di Mutually Exclusive Exon.

## References

- [1] Denti L, Rizzi R, Beretta S, Vedova GD, Previtali M, Bonizzoni P. ASGAL: aligning RNA-Seq data to a splicing graph to detect novel alternative splicing events. BMC Bioinformatics. 2018;19(1):444. Published 2018 Nov 20. doi:10.1186/s12859-018-2436-3
- [2] Salmon Documentation, Rob Patro, Geet Duggal, Mike Love, Rafael Irizarry and Carl Kingsford. Can be found at: <https://salmon.readthedocs.io/en/latest/>
- [3] Sequence Alignment/Map Format Specification, The SAM/BAM Format Specification Working Group, 14 May 2019. Can be found at: <http://samtools.github.io/hts-specs/SAMv1.pdf>