Laboratorio di Elementi di Bioinformatica

Laurea Triennale in Informatica
(codice: E3101Q116)

AA 2016/2017

Il formato SAM per memorizzare allineamenti

Docente del laboratorio: Raffaella Rizzi

L’allineamento tra due sequenze

In generale, l’allineamento tra due sequenze è una matrice di due righe che descrive la relazione tra i simboli delle due sequenze evidenziando come essi si appaiano.

L’allineamento tra due sequenze

S_1=AGCTAGGCGGTGACAAATG
S_2=TGCTTGGGCGGACAAATG

A	G	C	T	A	G	G	C	G	G	T	-	G	A	C	A	A	A	T	G
T	G	C	T	T	G	G	-	G	-	C	G	G	A	C	A	A	A	T	G

Un allineamento tra due sequenze può essere definito come una matrice M di due righe

L’allineamento tra due sequenze

S_1=AGCTAGGCGGTGACAAATG
S_2=TGCTTGGGCGGACAAATG

A	G	C	T	A	G	G	C	G	G	T	-	G	A	C	A	A	A	T	G
T	G	C	T	T	G	G	-	G	-	C	G	G	A	C	A	A	A	T	G

Un allineamento tra due sequenze può essere definito come una matrice M di due righe:

- ✓ ➡ la prima riga corrisponde a S_1

L’allineamento tra due sequenze

S_1=AGCTAGGCGGTGACAAATG
S_2=TGCTTGGGCGGACAAATG

➡	A	G	C	T	A	G	G	C	G	G	T	-	G	A	C	A	A	A	T	G
➡	T	G	C	T	T	G	G	-	G	-	C	G	G	A	C	A	A	A	T	G

Un allineamento tra due sequenze può essere definito come una matrice M di due righe:

- ✓ ➡ la prima riga corrisponde a S_1
- ✓ ➡ la seconda riga corrisponde a S_2

L’allineamento tra due sequenze

S_1=AGCTAGGCGGTGACAAATG
S_2=TGCTTGGGCGGACAAATG

A	G	C	T	A	G	G	C	G	G	T	-	G	A	C	A	A	A	T	G
T	G	C	T	T	G	G	-	G	-	C	G	G	A	C	A	A	A	T	G

Il simbolo trattino - rappresenta cancellazione/inserimento

L’allineamento tra due sequenze

S_1=AGCTAGGCGGTGACAAATG
S_2=TGCTTGGGCGGACAAATG

A	G	C	T	A	G	G	C	G	G	T	-	G	A	C	A	A	A	T	G
T	G	C	T	T	G	G	-	G	-	C	G	G	A	C	A	A	A	T	G

La concatenazione dei simboli della prima riga (escluso il -), presi in ordine, produce la sequenza S_1

L’allineamento tra due sequenze

S_1=AGCTAGGCGGTGACAAATG
S_2=TGCTTGGGCGGACAAATG

A	G	C	T	A	G	G	C	G	G	T	-	G	A	C	A	A	A	T	G
T	G	C	T	T	G	G	-	G	-	C	G	G	A	C	A	A	A	T	G

La concatenazione dei simboli della seconda riga (escluso il -), presi in ordine, produce la sequenza S_2

L’allineamento tra due sequenze

S_1=AGCTAGGCGGTGACAAATG
S_2=TGCTTGGGCGGACAAATG

A	G	C	T	A	G	G	C	G	G	T	-	G	A	C	A	A	A	T	G
T	G	C	T	T	G	G	-	G	-	C	G	G	A	C	A	A	A	T	G

Ogni colonna rappresenta un appaiamento tra un simbolo della prima sequenza e un simbolo della seconda, e prende il nome di coppia appaiata

L’allineamento tra due sequenze

S_1=AGCTAGGCGGTGACAAATG
S_2=TGCTTGGGCGGACAAATG

A	G	C	T	A	G	G	C	G	G	T	-	G	A	C	A	A	A	T	G
T	G	C	T	T	G	G	-	G	-	C	G	G	A	C	A	A	A	T	G

Coppia appaiata (A,T):
sostituzione del simbolo A con il simbolo T: *mismatch (o sostituzione)*

L’allineamento tra due sequenze

S_1=AGCTAGGCGGTGACAAATG
S_2=TGCTTGGGCGGACAAATG

A	G	C	T	A	G	G	C	G	G	T	-	G	A	C	A	A	A	T	G
T	G	C	T	T	G	G	-	G	-	C	G	G	A	C	A	A	A	T	G

Coppia appaiata (G,G):
simboli uguali: *match*

L’allineamento tra due sequenze

S_1=AGCTAGGCGGTGACAAATG
S_2=TGCTTGGGCGGACAAATG

A	G	C	T	A	G	G	C	G	G	T	-	G	A	C	A	A	A	T	G
T	G	C	T	T	G	G	-	G	-	C	G	G	A	C	A	A	A	T	G

Coppia appaiata (C,-):
inserimento del simbolo C in S_1
cancellazione del simbolo C in S_2

L’allineamento tra due sequenze

S_1=AGCTAGGCGGTGACAAATG
S_2=TGCTTGGGCGGACAAATG

A	G	C	T	A	G	G	C	G	G	T	-	G	A	C	A	A	A	T	G
T	G	C	T	T	G	G	-	G	-	C	G	G	A	C	A	A	A	T	G

Coppia appaiata (-,G):
cancellazione del simbolo G in S_1
inserimento del simbolo G in S_2

L’allineamento tra due sequenze

S_1=AGCTAGGCGGTGACAAATG
S_2=TGCTTGGGCGGACAAATG

A	G	C	T	A	G	G	C	G	G	T	-	G	A	C	A	A	A	T	G
T	G	C	T	T	G	G	-	G	-	C	G	G	A	C	A	A	A	T	G

indel

Il formato SAM

Il formato SAM (Sequence Alignment Map) permette di memorizzare gli allineamenti di sequenze (*query sequences*) rispetto ad altre prese come riferimento (*reference sequences*).

Reference sequences (o *references*)

- ✓ tipicamente lunghe (ad esempio un cromosoma)
- ✓ sequenze annotate (studi precedenti)

Query sequences (o *queries*)

- ✓ tipicamente corte
- ✓ sequenza prodotte in esperimenti di sequenziamento (prendono anche il nome di *read*)

Il formato SAM (e BAM)

Il formato SAM è di puro testo:

- ✓ facile da produrre
- ✓ facile da leggere e da sottoporre a *parsing*

La versione binaria di SAM è il formato BAM:

- ✓ veloce da sottoporre a *parsing*
- ✓ indicizzabile
- ✓ usato per processamenti intensivi

Il formato SAM (e BAM)

Il formato SAM è di puro testo:

- ✓ facile da produrre
- ✓ facile da leggere e da sottoporre a *parsing*

La ver

- ✓ vel
- ✓ ind
- ✓ usa

I Samtools costituiscono un insieme di utilità che manipolano allineamenti in formato SAM/BAM per:

- ✓ effettuare ordinamenti, unioni e indicizzazioni
- ✓ rintracciare allineamenti in qualsiasi regione della *reference* sequence in modo veloce

Tipi di allineamento

I tipi di allineamento *query/reference* che il formato SAM permette di rappresentare sono:

- ✓ *mapping*
- ✓ *clipped* alignment
- ✓ *spliced* alignment
- ✓ *multiple* alignment
- ✓ *multi-part* alignment
- ✓ *padded* alignment

Mapping

Un *mapping* descrive la relazione tra una *query* sequence (read) che si allinea a una sottostringa della *reference* sequence.

Ad esempio:

Reference: `acgtgtgacgatgcaaaatgatgctgaccgtaaaccatgacgtag`

Query: `caaatgatgc`

```
acgtgtgacgatgcaaaatgatgctgaccgtaaaccatgacgtag
------------caaa-tgatgc--------------------
```

Clipped alignment

Un *clipped alignment* descrive la relazione tra una *query* sequence che ha una sottostringa che si allinea ad una sottostringa della *reference* sequence (lasciando fuori un prefisso e/o un suffisso).

Ad esempio:

Reference: `acgtgtgacgatgcaaaatgatgctgaccgtaaaccatgacgtag`

Query: `gggggcaaatgatgcggggg`

```
acgtgtgacgatgcaaaatgatgctgaccgtaaaccatgacgtag
--------ggggg caaa-tgatgc ggggg----------------
```

In rosso sono evidenziate le *clipped sequences*

Spliced alignment

Uno *spliced alignment* descrive la relazione tra una *query* sequence che è una concatenazione di fattori che si allineano a regioni della *reference* sequence.

Ad esempio:

Reference: `acgtgtgacgatgcaaagtccatgatgctgaccgtaaaccatgac`

Query: `tgacgatatgatgct`

```
acgtgtgacgatgcaaagtccatgatgctgaccgtaaaccatgac
-----tgacgat---------atgatgct---------------
```

Multiple alignment

Un *multiple alignment* si ha quando la *query* si allinea alla *reference* in “luoghi” diversi della *reference*.

Ad esempio:

Reference: `acgtgtgacgatgcaaagtccatgatgctgaccgtaaaccatgac`

Query: `tgacgtg`

```
acgtg tgagcgta acgtggcaaag tgacgta atgctgaccgtaaacc
-----tga-cgta------------------------------------
----------------------tgacgta---------------
```

I due allineamenti vengono rappresentati (nel formato SAM) con record diversi, ma il loro legame logico viene mantenuto. Di solito uno degli allineamenti di un *multiple alignment* viene etichettato come *primary alignment*.

Multi-part alignment

Un *multi-part alignment* si ha quando la *query* si allinea alla *reference* in posti diversi ma viene spezzata in parti.

Ad esempio:

Reference: `acgtgtgagcgtaacgtggcaaagtgacgtaatgctgaccgtaaacc`

Query: `tgacgtagacgtaa`

```
acgtgtgagcgtaacgtggcaaagtgacgtaatgctgaccgtaaacc
-----tgacgtagacgtaa----------------------------
------------------tgacgtagacgtaa--------------
```

In rosso sono evidenziate le *clipped* sequences relative agli allineamenti delle due parti del read.

Padded alignment

Un *padded alignment* descrive gli inserimenti nella *reference* sequence relativi agli allineamenti di un set di *query* sequences.

Ad esempio:

Reference: `acgtgtgagcgtaacgtggcaaa`

Query 1: `gtgattgcg`, *Query 2*: `gtgatgcg`, *Query 3*: `gtgagcg`

```
    acgtgtga--gcgtaacgtggcaaa
Q1  ----gtgattgcg-----------

    acgtgtga-gcgtaacgtggcaaa
Q2  ----gtgatgcg-----------

    acgtgtgagcgtaacgtggcaaa
Q3  ----gtgagcg-----------
```

Padded alignment

Un *padded alignment* descrive gli inserimenti nella *reference* sequence relativi agli allineamenti di un set di *query* sequences.

Ad esempio:

Reference: `acgtgtgagcgtaacgtggcaaa`

Query 1: `gtgattgcg`, *Query 2*: `gtgatgcg`, *Query 3*: `gtgagcg`

```
      acgtgtga--gcgtaacgtggcaaa
Q1    ----gtgattgcg------------

      acgtgtga-gcgtaacgtggcaaa
Q2    ----gtgatgcg------------

      acgtgtgagcgtaacgtggcaaa
Q3    ----gtgagcg------------
```

```
acgtgtga--gcgtaacgtggcaaa
----gtgattgcg------------
----gtga*tgcg------------
----gtga**gcg------------
```

* = delezione “silente”

Il formato SAM

Scopo: memorizzare gli allineamenti di un set *S* di *queries* (reads) rispetto ad un set *R* di sequenze di riferimento (*references*)

Composto da due sezioni:

1. Header Section
2. Alignment Section

Il formato SAM

Scopo: memorizzare gli allineamenti di un set *S* di *queries* (reads) rispetto ad un set *R* di sequenze di riferimento (*references*)

Composto da due sezioni:

1. **Header Section**
2. Alignment Section

Header Section

La *Header Section* è composta da un insieme di record (righe) che iniziano con il simbolo @

Il significato di ogni record è specificato da un codice a due lettere che segue immediatamente il simbolo @

Header Section

La *Header Section* è composta da un insieme di record (righe) che iniziano con il simbolo @

Il significato di ogni record è specificato da un codice a due lettere che segue immediatamente il simbolo @

Ogni record contiene una lista di attributi (separati da tabulazione), dove ogni attributo segue il formato:

TAG:VALUE

Header Section

La *Header Section* è composta da un insieme di record (righe) che iniziano con il simbolo @

Il significato di ogni record è specificato da un codice a due lettere che segue immediatamente il simbolo @

Ogni record contiene una lista di attributi (separati da tabulazione), dove ogni attributo segue il formato:

TAG:VALUE

dove TAG è una stringa di due caratteri che identifica l'attributo e VALUE è il suo valore

Record @HD (*Header Section*)

Record @HD (uno solo e deve essere il primo):

```
@HD   VN:[version]     SO:[sorting]
```

Il record @HD rappresenta l'*header* della Header Section.

L'attributo `VN` è obbligatorio è il suo valore `[version]` è la versione del formato (ad esempio 1.0)

L'attributo `SO` indica come sono ordinati gli allineamenti nella *Alignment Section;* i valori possibili sono:

- ✓ `unsorted`
- ✓ `queryname`
- ✓ `coordinate`
- ✓ `unknown`

Record @SQ (*Header Section*)

Record @SQ (uno o più) :

```
@SQ   SN:[ref_name]    LN:[length]
```

Ogni record @SQ rappresenta una *reference* sequence.

Record @SQ (*Header Section*)

Record @SQ (uno o più) :

```
@SQ   SN:[ref_name]    LN:[length]
```

Ogni record @SQ rappresenta una *reference* sequence.

L'attributo `SN` è obbligatorio è il suo valore `[ref_name]` è l'identificatore della *reference* sequence

Record @SQ (*Header Section*)

Record @SQ (uno o più) :

```
@SQ   SN:[ref_name]    LN:[length]
```

Ogni record @SQ rappresenta una *reference* sequence.

L'attributo `SN` è obbligatorio è il suo valore `[ref_name]` è l'identificatore della *reference* sequence

L'attributo `LN` è obbligatorio è il suo valore `[length]` è la lunghezza dela *reference* sequence

Record @SQ (*Header Section*)

Record @SQ (uno o più) :

```
@SQ   SN:[ref_name]    LN:[length]
```

Ogni record @SQ rappresenta una *reference* sequence.

L'attributo `SN` è obbligatorio è il suo valore `[ref_name]` è l'identificatore della *reference* sequence

L'attributo `LN` è obbligatorio è il suo valore `[length]` è la lunghezza dela *reference* sequence

Il record @SQ prevede altri attributi (opzionali) tra i quali `SP`, il cui valore è il nome della specie della sequenza.

Record @RG (*Header Section*)

Record @RG (uno o più):

```
@RG   ID:[group_id]    SM:[sample]
```

Ogni record @RG rappresenta un gruppo di reads.

Record @RG (*Header Section*)

Record @RG (uno o più):

```
@RG   ID:[group_id]    SM:[sample]
```

Ogni record @RG rappresenta un gruppo di reads.

L'attributo `ID` è obbligatorio è il suo valore `[group_id]` è l'identificatore del gruppo.

Record @RG (*Header Section*)

Record @RG (uno o più):

```
@RG   ID:[group_id]    SM:[sample]
```

Ogni record @RG rappresenta un gruppo di reads.

L'attributo `ID` è obbligatorio è il suo valore `[group_id]` è l'identificatore del gruppo

L'attributo `SM` è obbligatorio è il suo valore `[sample]` è l'identificatore del campione da cui sono stati sequenziati i reads del gruppo.

Record @RG (*Header Section*)

Record @RG (uno o più):

```
@RG   ID:[group_id]    SM:[sample]
```

Tra gli altri attributi (opzionali):

```
LB:[library]
DS:[description]
PU:[platform_unit]
DT:[date]
```

Record @PG (*Header Section*)

Record @PG (uno o più record):

`@PG  ID:[program_id]  PN:[program_name]`

Ogni record @PG rappresenta un software (di allineamento).

Record @PG (*Header Section*)

Record @PG (uno o più record):

`@PG  ID:[program_id]  PN:[program_name]`

Ogni record @PG rappresenta un software (di allineamento).

L'attributo `ID` è obbligatorio è il suo valore `[program_id]` è l'identificatore associato al software.

Record @PG (*Header Section*)

Record @PG (uno o più record):

`@PG  ID:[program_id]  PN:[program_name]`

Ogni record @PG rappresenta un software (di allineamento).

L'attributo `ID` è obbligatorio è il suo valore `[program_id]` è l'identificatore associato al software.

L'attributo `PN` è opzionale è il suo valore `[program_name]` è il nome del software.

Record @PG (*Header Section*)

Record @PG (uno o più record):

```
@PG   ID:[program_id]      PN:[program_name]
```

Tra gli altri attributi (opzionali):

```
VN:[program_version]
CL:[command_line]
```

Record @CO (*Header Section*)

Record @CO (uno o più record):

```
@CO  [comment]
```

`[comment]` è un commento on-line

Il formato SAM

Scopo: memorizzare gli allineamenti di un set *S* di *queries* (reads) rispetto ad un set *R* di sequenze di riferimento (*references*)

Composto da due sezioni:

1. Header Section
2. **Alignment Section**

Alignment Section

La sezione degli allineamenti fornisce gli allineamenti tra un set S di *queries* e il set R delle *references* specificate nella *header section*.

Alignment Section

La sezione degli allineamenti fornisce gli allineamenti tra un set S di *queries* e il set R delle *references* specificate nella *header section*.

Ogni record rappresenta l'allineamento tra una *query* (o *read*) di S e una *reference* di R.

Alignment Section

La sezione degli allineamenti fornisce gli allineamenti tra un set S di *queries* e il set R delle *references* specificate nella *header section*.

Ogni record rappresenta l'allineamento tra una *query* (o *read*) di S e una *reference* di R.

Ogni record è composto dai seguenti undici campi obbligatori (TAB-separated):

Alignment Section

La sezione degli allineamenti fornisce gli allineamenti tra un set S di *queries* e il set R delle *references* specificate nella *header section*.

Ogni record rappresenta l'allineamento tra una *query* (o *read*) di S e una *reference* di R.

Ogni record è composto dai seguenti undici campi obbligatori (TAB-separated):

1. QNAME ➔ identificatore della *query* (o *read*)

Alignment Section

La sezione degli allineamenti fornisce gli allineamenti tra un set S di *queries* e il set R delle *references* specificate nella *header section*.

Ogni record rappresenta l'allineamento tra una *query* (o *read*) di S e una *reference* di R.

Ogni record è composto dai seguenti undici campi obbligatori (TAB-separated):

1. QNAME ➔ identificatore della *query* (o *read*)
2. FLAG ➔ intero a 16 bit da interpretare come stringa di bit; ogni bit è un flag con un preciso significato.

   Ad esempio:

   - ✓ se il *read* è stato sottoposto a reverse&complement
   - ✓ se l'allineamento è *primario* (cioé ne esistono anche di alternativi)
   - ✓ se il *read* fa parte di un paired-end e se, in tale caso, è il primo o il secondo read dell'accoppiamento

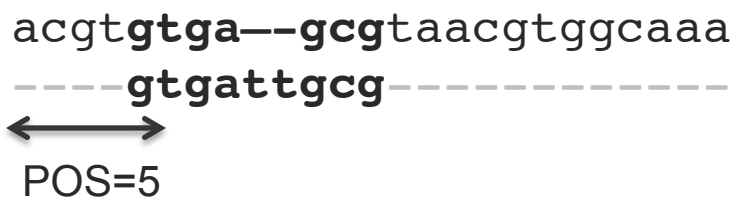
Alignment Section

2. …
3. RNAME ➔ identificatore della *reference* a cui il read si allinea

RNAME deve essere uguale al valore dell'attributo SN di uno dei record @SQ della *Header* Section (se non è disponibile si trova un '*')

Alignment Section

2. …
3. RNAME ➔ identificatore della *reference* a cui il read si allinea
4. POS ➔ posizione (sulla *reference*) di inizio dell'allineamento



Alignment Section

2. …
3. RNAME ➔ identificatore della *reference* a cui il read si allinea
4. POS ➔ posizione (sulla *reference*) di inizio dell'allineamento
5. MAPQ ➔ qualità dell'allineamento pari a $-10 \log_{10} p_w$, dove p_w è la probabilità che la posizione di allineamento sia sbagliata (se si trova un valore pari a 255 significa che MAPQ non è disponibile)

Alignment Section

2. …
3. RNAME ➔ identificatore della *reference* a cui il read si allinea
4. POS ➔ posizione (sulla *reference*) di inizio dell'allineamento
5. MAPQ ➔ qualità dell'allineamento pari a $-10 \log_{10} p_w$, dove p_w è la probabilità che la posizione di allineamento sia sbagliata (se si trova un valore pari a 255 significa che MAPQ non è disponibile)
6. CIGAR string ➔ definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole

Alignment Section

2. …
3. RNAME ➔ identificatore della *reference* a cui il read si allinea
4. POS ➔ posizione (sulla *reference*) di inizio dell'allineamento
5. MAPQ ➔ qualità dell'allineamento pari a $-10 \log_{10} p_w$, dove p_w è la probabilità che la posizione di allineamento sia sbagliata (se si trova un valore pari a 255 significa che MAPQ non è disponibile)
6. CIGAR string ➔ definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole
7. MRNM ➔ identificatore della *reference* sequence dell'allineamento primario del *mate* read (se esiste). Se MRNM è '=', allora MRNM coincide con RNAME

Alignment Section

7. …
8. MPOS ➔ posizione di inizio dell'allineamento del *mate* read

Alignment Section

7. …
8. MPOS ➔ posizione di inizio dell'allineamento del *mate* read
9. ISIZE ➔ *inferred* insertion size

Alignment Section

7. …
8. MPOS ➔ posizione di inizio dell'allineamento del *mate* read
9. ISIZE ➔ *inferred* insertion size
10. SEQ ➔ sequenza della *query* (o *read*)

Alignment Section

7. …
8. MPOS ➔ posizione di inizio dell'allineamento del *mate* read
9. ISIZE ➔ *inferred* insertion size
10. SEQ ➔ sequenza della *query* (o *read*)
11. QUAL ➔ stringa di caratteri (della stessa lunghezza di SEQ) che codifica in ASCII i *Phred Values* di SEQ

Alignment Section

7. …
8. MPOS ➔ posizione di inizio dell'allineamento del *mate* read
9. ISIZE ➔ *inferred* insertion size
10. SEQ ➔ sequenza della *query* (o *read*)
11. QUAL ➔ stringa di caratteri (della stessa lunghezza di SEQ) che codifica in ASCII i *Phred Values* di SEQ

QUAL fornisce la qualità di ogni singola base della sequenza della query

Alignment Section

La CIGAR string (sesto campo) definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole:

- ✓ M: match o mismatch
- ✓ I: inserimento nella *reference*
- ✓ D: delezione nella *reference*
- ✓ N: inserimento nella *reference* dovuto ad allineamento *spliced*
- ✓ S: *soft clipping (*della sequenza di *query*)
- ✓ H: *hard clipping (*della sequenza di *query)*
- ✓ P: "delezione" silente (*padding*)

Alignment Section

La CIGAR string (sesto campo) definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole.

Esempio 1

```
acgtgtga--gcgtaacgtggcaaa
----gtgattgcg------------
```

5

POS = 5

CIGAR=4M2D3M

SEQ=gtgattgcg

Alignment Section

La CIGAR string (sesto campo) definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole.

Esempio 1

```
acgtgtga--gcgtaacgtggcaaa
----gtgattgcg------------
```

5

POS = 5

4 match

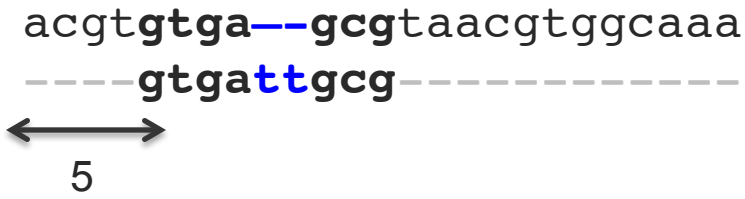
CIGAR=4M2D3M

SEQ=gtgattgcg

Alignment Section

La CIGAR string (sesto campo) definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole.

Esempio 1



2 delezioni

POS = 5

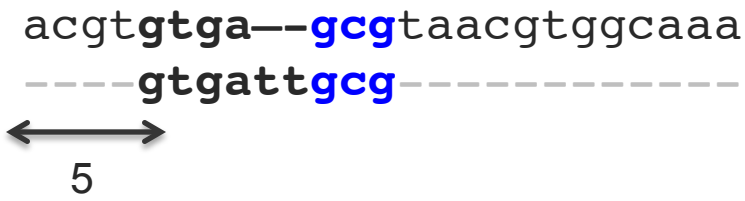
CIGAR=4M2D3M

SEQ=gtgattgcg

Alignment Section

La CIGAR string (sesto campo) definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole.

Esempio 1



POS = 5

3 match

CIGAR=4M2D3M

SEQ=gtgattgcg

Alignment Section

La CIGAR string (sesto campo) definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole.

Esempio 2

```
acgtgtgacgatgcaaaatgatgctgaccgtaaaccatgacgtag
--------gggggcaaa-tgatgcggggg----------------
```

9

POS = 9

CIGAR=4S5M1I6M5S

SEQ=gggggcaaatgatgcggggg

Alignment Section

La CIGAR string (sesto campo) definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole.

Esempio 2

```
acgtgtgacgatgcaaaatgatgctgaccgtaaaccatgacgtag
--------gggggcaaa-tgatgcggggg----------------
```

9

4 soft clipping

POS = 9

CIGAR=4S5M1I6M5S

SEQ=gggggcaaatgatgcggggg

Alignment Section

La CIGAR string (sesto campo) definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole.

Esempio 2

```
acgtgtgacgatgcaaaatgatgctgaccgtaaaccatgacgtag
--------gggggcaaa-tgatgcggggg----------------
```

9

4 match

POS = 9

CIGAR=4S5M1I6M5S

SEQ=gggggcaaatgatgcggggg

Alignment Section

La CIGAR string (sesto campo) definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole.

Esempio 2

```
acgtgtgacgatgcaaaatgatgctgaccgtaaaccatgacgtag
--------gggggcaaa-tgatgcggggg----------------
```

9

1 inserimento

POS = 9

CIGAR=4S5M1I6M5S

SEQ=gggggcaaatgatgcggggg

Alignment Section

La CIGAR string (sesto campo) definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole.

Esempio 2

```
acgtgtgacgatgcaaaatgatgctgaccgtaaaccatgacgtag
--------ggggg caaa-tgatgc ggggg----------------
```

9

6 match

POS = 9

CIGAR=4S5M1I6M5S

SEQ=gggggcaaatgatgcggggg

Alignment Section

La CIGAR string (sesto campo) definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole.

Esempio 2

```
acgtgtgacgatgcaaaatgatgctgaccgtaaaccatgacgtag
--------ggggg caaa-tgatgcggggg----------------
```

9

5 soft clipping

POS = 9

CIGAR=4S5M1I6M5S

SEQ=gggggcaaatgatgcggggg

Alignment Section

La CIGAR string (sesto campo) definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole.

Esempio 3

```
acgtgtgacgatgcaaagtccatgatgctgaccgtaaaccatgac
------tgacgat---------atgatgct----------------
```

6

POS = 6

CIGAR=7M9N8M

SEQ=tgacgatatgatgct

Alignment Section

La CIGAR string (sesto campo) definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole.

Esempio 3

```
acgtgtgacgatgcaaagtccatgatgctgaccgtaaaccatgac
------tgacgat---------atgatgct----------------
```

6

7 match

POS = 6

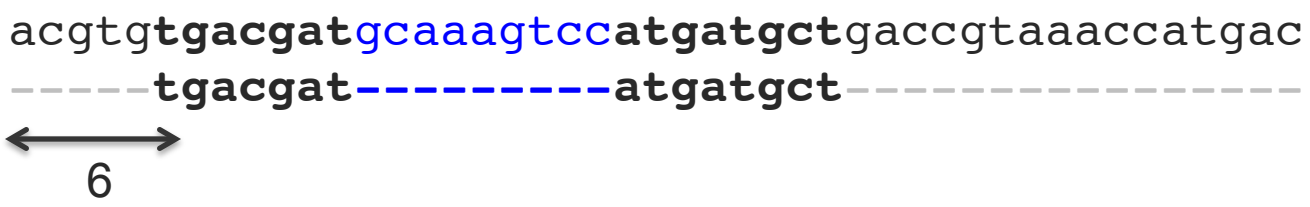
CIGAR=7M9N8M

SEQ=tgacgatatgatgct

Alignment Section

La CIGAR string (sesto campo) definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole.

Esempio 3



7 inserimenti da allineamento *spliced*

POS = 6

CIGAR=7M9N8M

SEQ=tgacgatatgatgct

Alignment Section

La CIGAR string (sesto campo) definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole.

Esempio 3

```
acgtgtgacgatgcaaagtccatgatgctgaccgtaaaccatgac
------tgacgat---------atgatgct----------------
```

6

8 match

POS = 6

CIGAR=7M9N8M

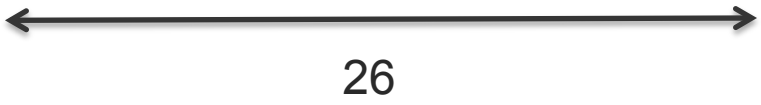
SEQ=tgacgatatgatgct

Alignment Section

La CIGAR string (sesto campo) definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole.

Esempio 4

```
acgtgtgagcgtaacgtggcaaagtgac-taatgctgaccgtaaacc
--------------------tgacgtagacgtaa----------------
```



POS = 26

CIGAR=7H3M1D3M

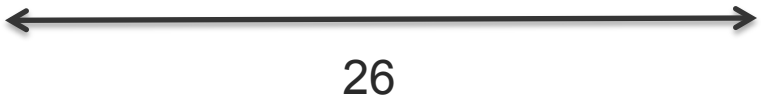
SEQ=gacgtaa

Alignment Section

La CIGAR string (sesto campo) definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole.

Esempio 4

```
acgtgtgagcgtaacgtggcaaagtgac-taatgctgaccgtaaacc
--------------------tgacgtagacgtaa----------------
```



7 hard clipping

POS = 26

CIGAR=7H3M1D3M

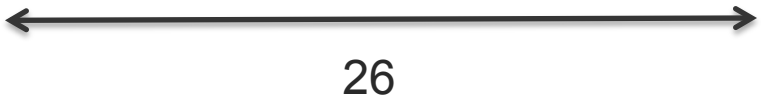
SEQ=gacgtaa

Alignment Section

La CIGAR string (sesto campo) definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole.

Esempio 4

```
acgtgtgagcgtaacgtggcaaagtgac-taatgctgaccgtaaacc
--------------------tgacgtagacgtaa---------------
```



3 match

POS = 26

CIGAR=7H3M1D3M

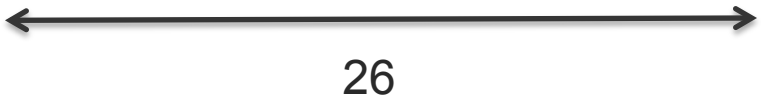
SEQ=gacgtaa

Alignment Section

La CIGAR string (sesto campo) definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole.

Esempio 4

```
acgtgtgagcgtaacgtggcaaagtgac-taatgctgaccgtaaacc
-------------------tgacgtagacgtaa---------------
```



1 delezione

POS = 26

CIGAR=7H3M1D3M

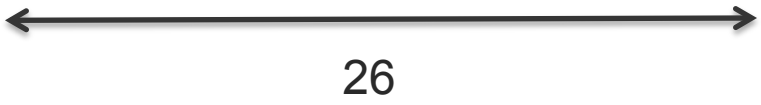
SEQ=gacgtaa

Alignment Section

La CIGAR string (sesto campo) definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole.

Esempio 4

```
acgtgtgagcgtaacgtggcaaagtgac-taatgctgaccgtaaacc
--------------------tgacgtagacgtaa---------------
```



3 match

POS = 26

CIGAR=7H3M1D3M

SEQ=gacgtaa

Alignment Section

La CIGAR string (sesto campo) definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole.

Esempio 5

```
acgtgtga--gcgtaacgtggcaaa

----gtgattgcg----------  query1

----gtga*tgcg----------  query2

----gtga**gcg----------  query3
```

CIGAR1=4M2D3M
CIGAR2=4M1P1D3M
CIGAR3=4M2P3M

Alignment Section

La CIGAR string (sesto campo) definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole.

Esempio 5

```
acgtgtga--gcgtaacgtggcaaa

----gtgattgcg----------  query1

----gtga*tgcg----------  query2

----gtga**gcg----------  query3
```

CIGAR1=4M**2D**3M
CIGAR2=4M1P1D3M
CIGAR3=4M2P3M

Alignment Section

La CIGAR string (sesto campo) definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole.

Esempio 5

```
acgtgtga--gcgtaacgtggcaaa

----gtgattgcg----------  query1

----gtga*tgcg----------  query2

----gtga**gcg----------  query3
```

CIGAR1=4M2D3M
CIGAR2=4M**1P1D**3M
CIGAR3=4M2P3M

Alignment Section

La CIGAR string (sesto campo) definisce le operazioni che forniscono l'allineamento della *query* rispetto alla *reference.* Le operazioni sono codificate in lettere maiuscole.

Esempio 5

```
acgtgtga--gcgtaacgtggcaaa

----gtgattgcg----------  query1

----gtga*tgcg----------  query2

----gtga**gcg----------  query3
```

CIGAR1=4M2D3M
CIGAR2=4M1P1D3M
CIGAR3=4M**2P**3M

Un esempio completo di SAM

```
@HD  VN:1.0        SO:coordinate
@SQ  SN:chr1       LN:247249719
@SQ  SN:chr22      LN:49691432
@RG  ID:L1         SM:NA12891
@RG  ID:L2         SM:NA12891
@PG  ID:P1         PN:BOWTIE
r001     435  chr1     30   *    8M2I4M1D3M   =    37   39   TTAGATAAAGGATACTG  *
r002     675  chr22    32   *    10M3D7M  =    37   39   TTATGAATTTGATACTGAAA   *
```

Il valore `coordinate` specifica che l'ordinamento degli allineamenti avviene prima sulla base del campo RNAME (nome della *reference* sequence) seguendo lo stesso ordine dei record @SQ) e, a parità di valore di RNAME, sulla base della posizione (campo POS).

Alignment Section

Oltre agli undici campi obbligatori, un record può anche avere campi aggiuntivi opzionali, tra i quali una serie di TAG che devono essere specificati nella forma:

TAG:VTYPE:VALUE

dove TAG è un codice a due caratteri (il primo è una lettera maiuscola e il secondo in è una lettera maiuscola o una cifra.

Ad esempio il tag **RG** permette di associare il *record* (di allineamento) a uno dei gruppi specificati in un record @RG nella Header Section. Il suo VTYPE è Z (che indica una stringa stampabile), e VALUE deve essere uguale all'attributo ID di un record @RG.

Alignment Section

Oltre agli undici campi obbligatori, un record può anche avere campi aggiuntivi opzionali, tra i quali una serie di TAG che devono essere specificati nella forma:

TAG:VTYPE:VALUE

dove TAG è un codice a due caratteri (il primo è una lettera maiuscola e il secondo in è una lettera maiuscola o una cifra.

Ad esempio il tag **PG** permette di associare il *record* (di allineamento) a uno dei software specificati in un record @PG nella Header Section. Il suo VTYPE è Z (che indica una stringa stampabile), e VALUE deve essere uguale all'attributo ID di un record @PG.

Esercizio

Scrivere un programma che prenda in input un file in formato SAM, e produca in standard output una tabella di report in cui ogni riga si riferisca all'allineamento di un *read* a una *reference*, e che abbia le seguenti colonne:

- ✓ identificatore del *read*
- ✓ identificatore della *reference* a cui il *read* si allinea
- ✓ nome del software che ha prodotto l'allineamento (se esiste)
- ✓ posizione dell'allineamento
- ✓ numero di *match/mismatch* nell'allineamento