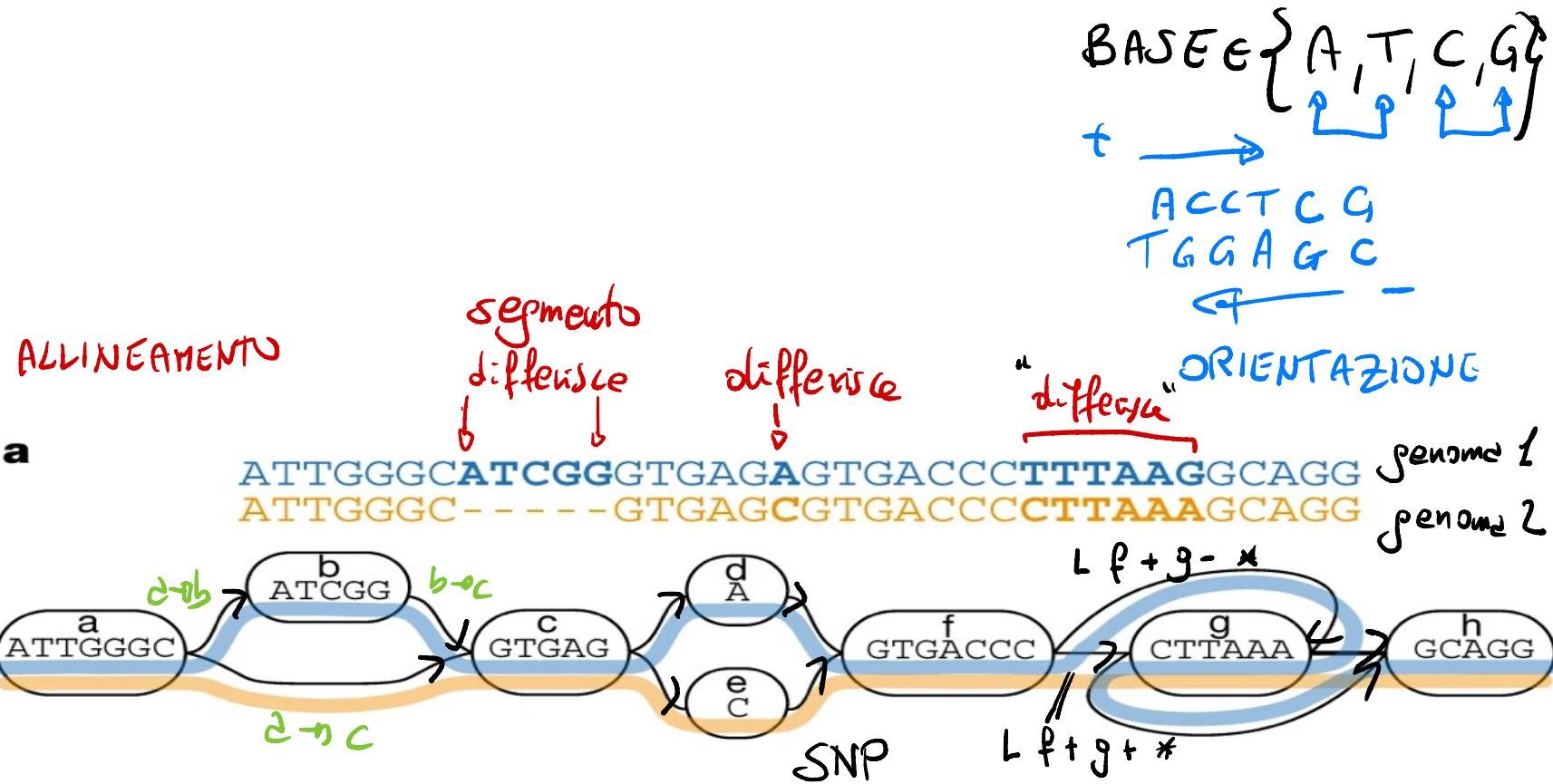
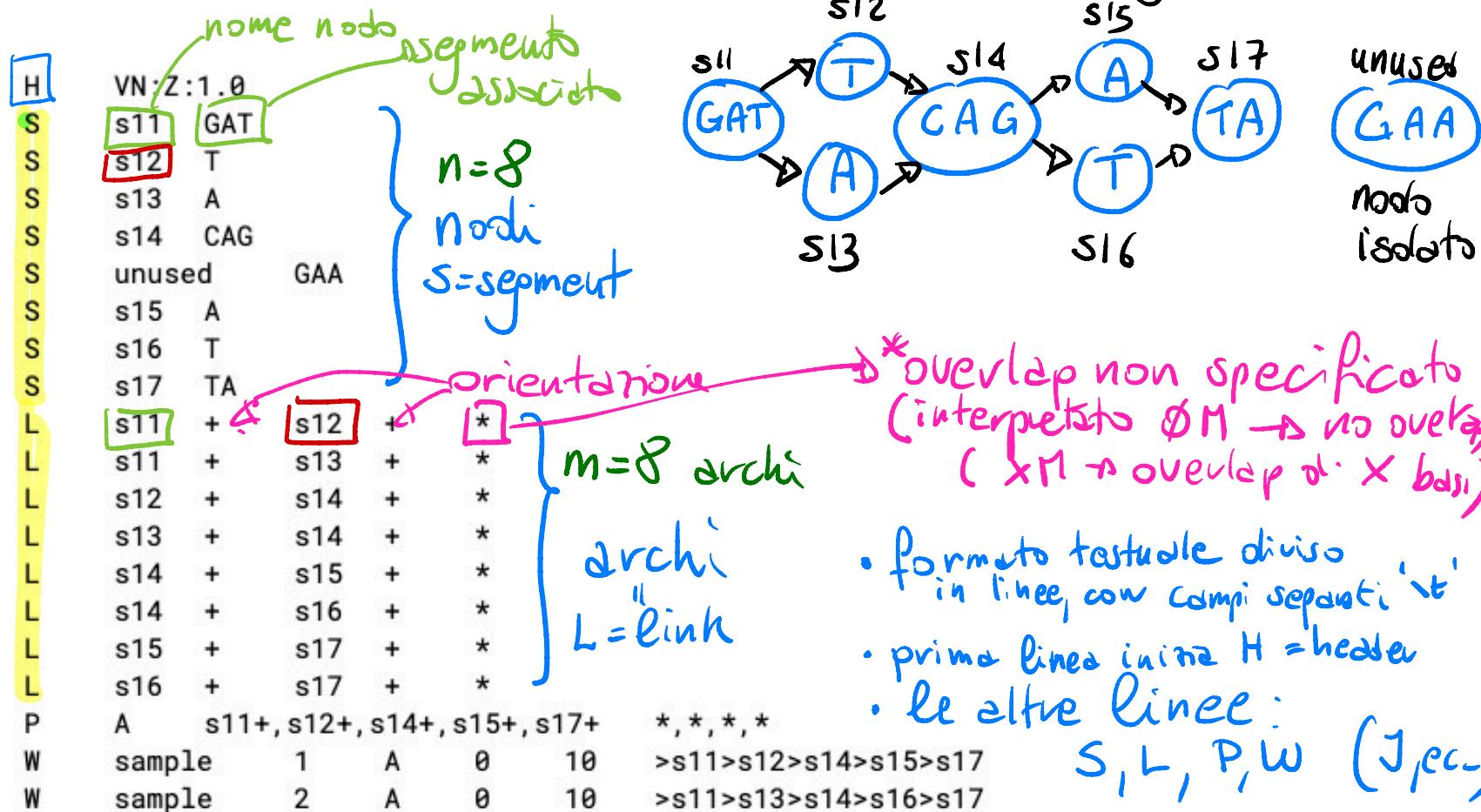
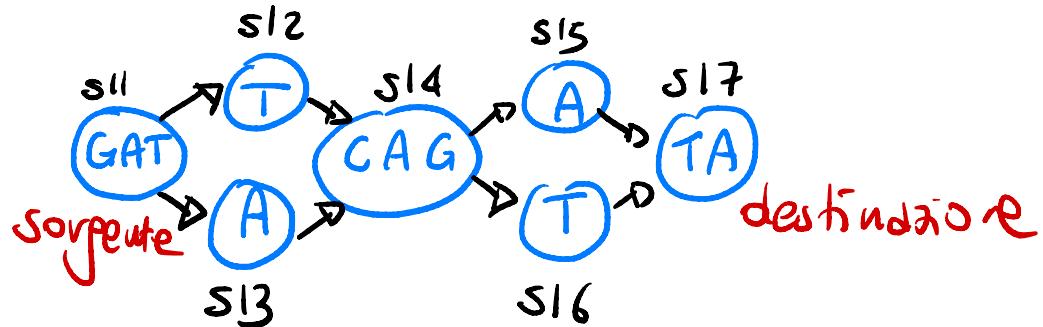


# PROGETTO: Pan genome graph (bioinformatics)



# Rappresentazione in formato GFA: Graphical Fragment Assembly

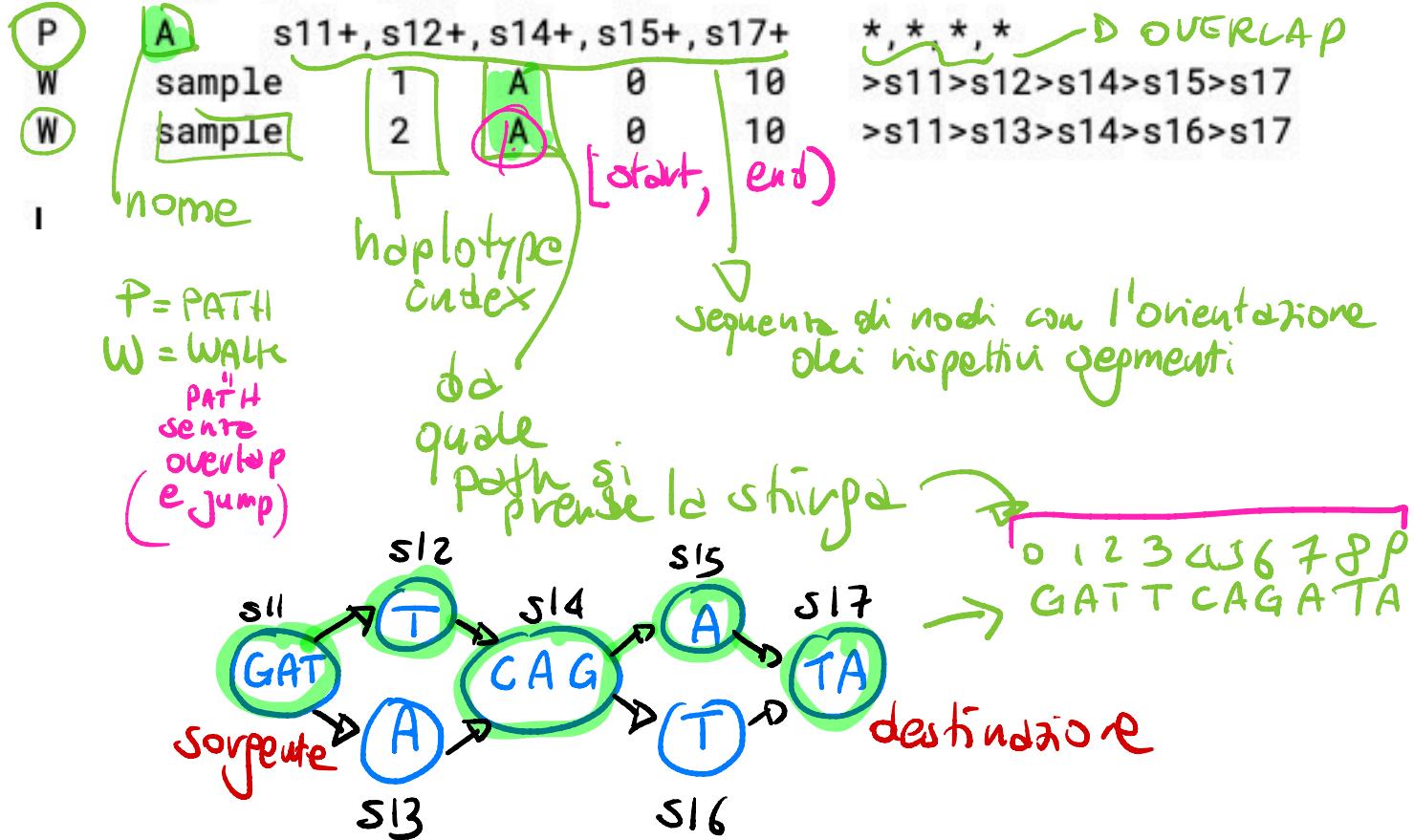




- cammini da sorgente (nodo con grado d'ingresso = 0) a destinazione (con grado d'uscita = 0)
- per ogni cammino, la sequenza corrispondente si ottiene concatenando i segmenti dei nodi attraversati.  
(tenendo conto di eventuali overlap)

- GAT I CAG A TA
- GAT A CAG A TA
- GAT I CAG I TA
- GAT A CAG I TA

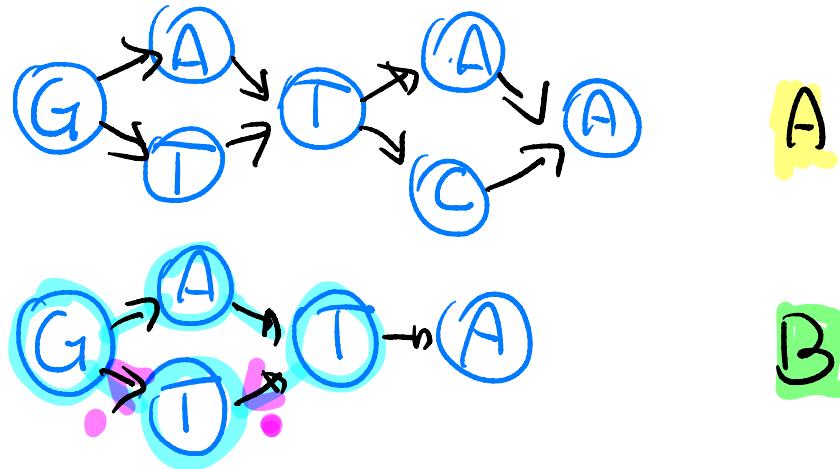
il numero di sequenze può essere esponenziale nel numero di nodi



nel nu

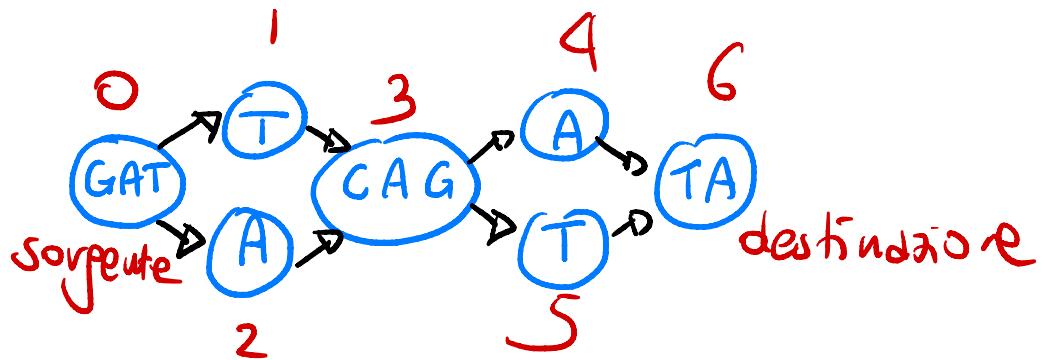
H VN:Z:1.0

S	11	G			
S	12	A			
S	13	T			
S	14	T			
S	15	A			
S	16	C			
S	17	A			
S	21	G			
S	22	A			
S	23	T			
S	24	T			
S	25	A			
L	11	+	12	+	*
L	11	+	13	+	*
L	12	+	14	+	*
L	13	+	14	+	*
L	14	+	15	+	*
L	14	+	16	+	*
L	15	+	17	+	*
L	16	+	17	+	*
L	21	+	22	+	*
L	21	+	23	+	*
L	22	+	24	+	*
L	23	+	24	-	*
L	24	+	25	+	*



> attraversa l'arco nella sua direzione  
< " " " nella direzione inversa

P	A	11+, 12+, 14+, 15+, 17+      *, *, *, *			
P	B	21+, 22+, 24+, 25+      *, *, *			
W	sample 1	A	0	5	>11>12>14>15>17
W	sample 2	A	0	5	>11>13>14>16>17
W	sample 1	B	0	5	>21>22>24<23<21
W	sample 2	B	0	4	>21>22>24>25



adj = array di vector  
label = array di stringhe

adj[ ] orientato

0	→ 1, 2
1	→ 3
2	→ 3
3	→ 4, 5
4	→ 6
5	→ 6
6	

per codificare  
+  
-  
usate interne negativi per -

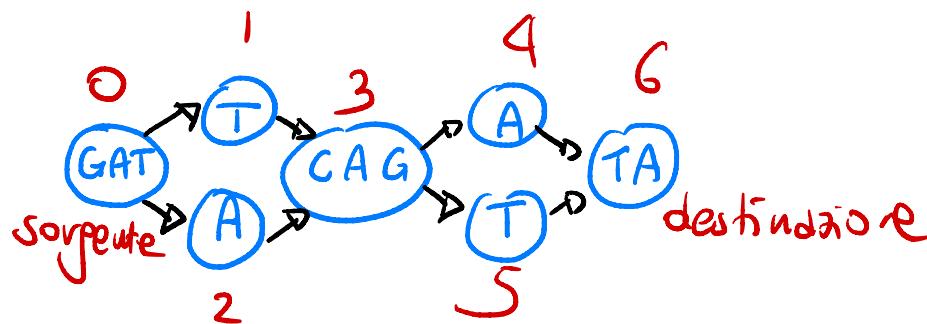
label[ ]

0	→ GAT
1	→ T
2	→ A
3	→ CAG
4	→ A
5	→ T
6	→ TA

Progetto :

- ① leggere file .gfa e creare grafo etichettato con label()  
"liste" di adiacenze  
 $\text{adjL}$ )

- ② Considerare tutti i possibili cammini  
sorgente-destinazione senza materializzarli  
oss. se ci sono più sorgenti e/o destinazioni  
scegliere una sola.



DFS sul grafo  
non scopre tutti i  
cammini?

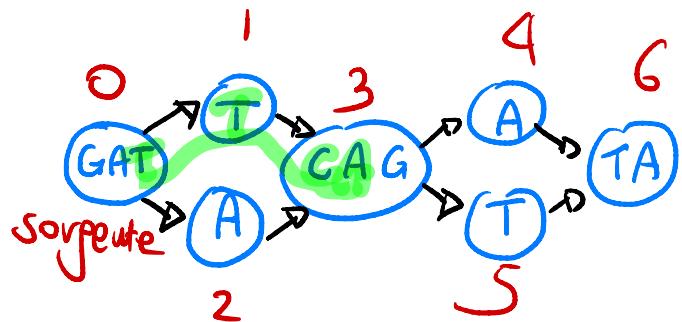
SUGGERIMENTO: Siccome  
il grafo è aciclico  
ipnotate il booleano "visitato"

Usando un array di appoggio  $S$ , modificare la  $DF(u)$  in modo che "appunga/tolga" la stringa del nodo  $u$  (appunga: inizio visita di  $u$ ; toglie: fine visita di  $u$ )

Fatto interessante: quando  $u$  è destinazione (cioè  $\text{pred}_u \text{uscite} = 0$ ), in  $S$  trovi la corrispondente sequenza.

- ③ Data una sequenza pattern  $P$ , verificare che sia contenuta in una delle sequenze generate nel punto ②.

es.  $P = \text{TTCA}$        $S = \text{GAT } \underline{\text{T}} \text{ CAG } \underline{\text{T}} \text{ TA}$



suggerimenti:

- a) sfruttate l'array di appoggio
- b) sfruttate l'hash visto a lezione (rolling hash)

se  $k = |P| = 4$ , calcolate l'hash delle porzioni lunghe  $k$  di  $S$  e confrontatelo con l'hash ( $P$ )

Note Se  $P$  occorre, è chiamato  $k$ -mero.

La sua frequenza è il numero di occurrente.

Per esempio  $P = ATA$  ha frequenza 2.

④ Dato  $K$  e un pangenome graph  $G$ , trovare  
i 10  $k$ -meri più frequenti in  $G$ .