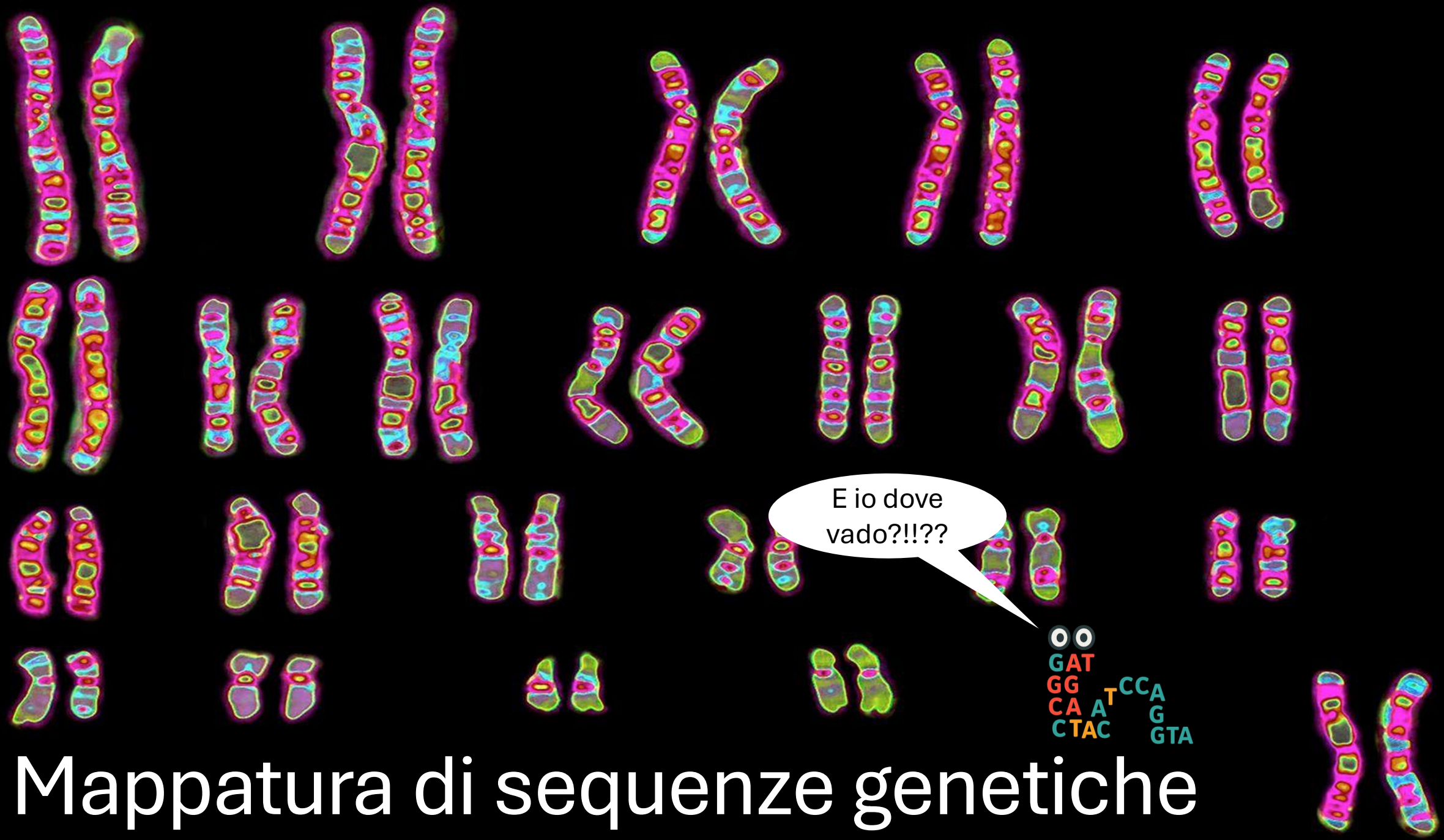
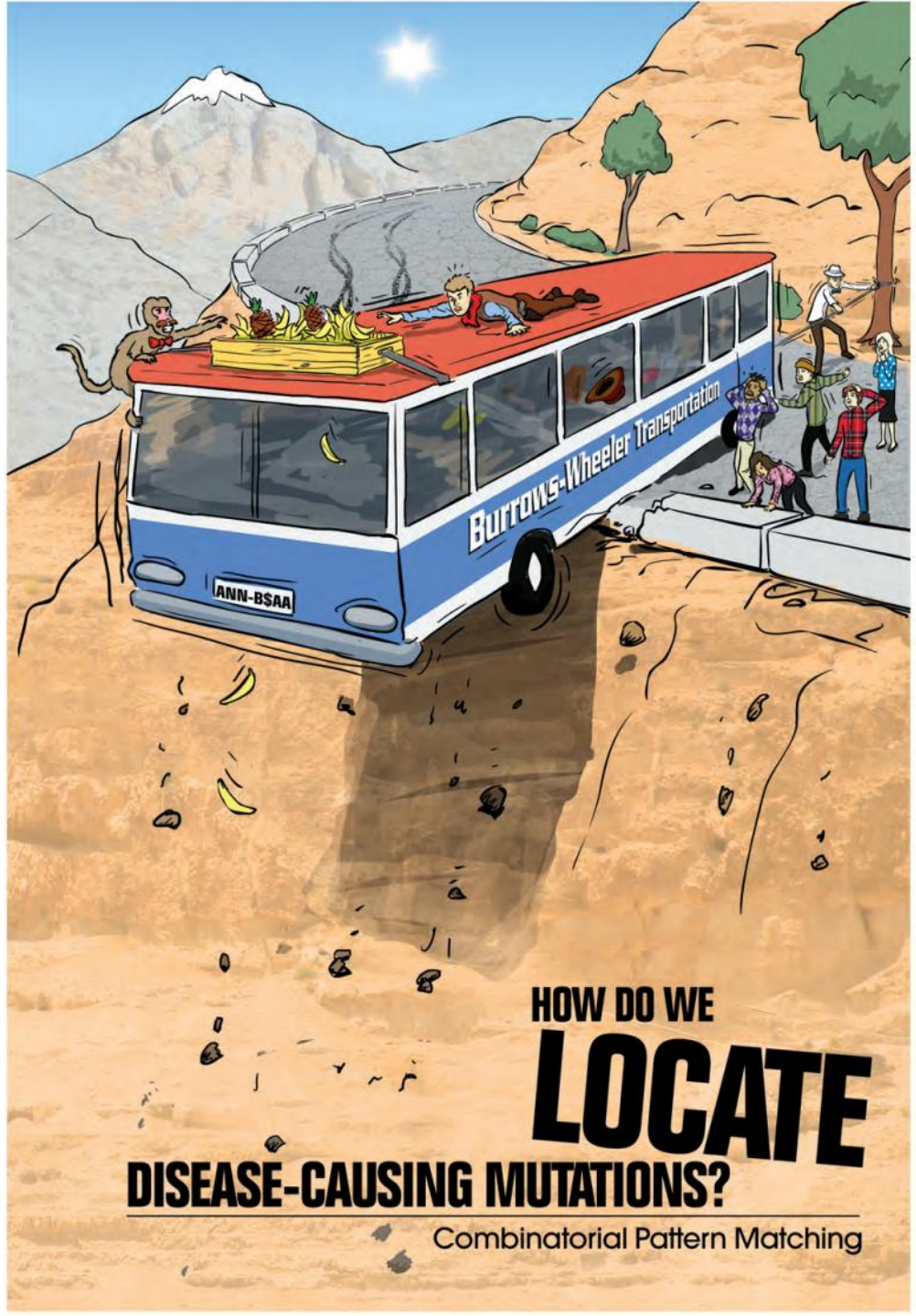


Mappatura di sequenze genetiche

Fabrizio Mafessoni, Genomica Computazionale, Laurea Specialistica in Genomica Funzionale, Universita' di Trieste



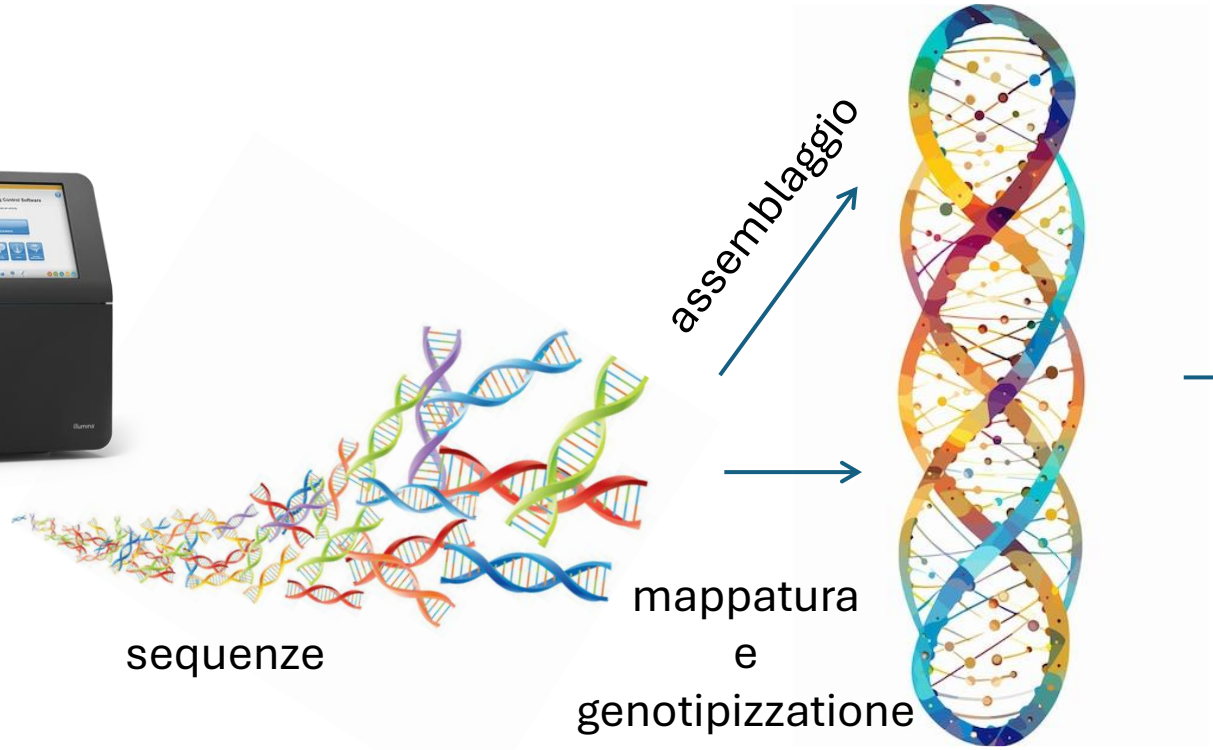
- Capitolo 9 del Libro Bioinformatics Algorithms.

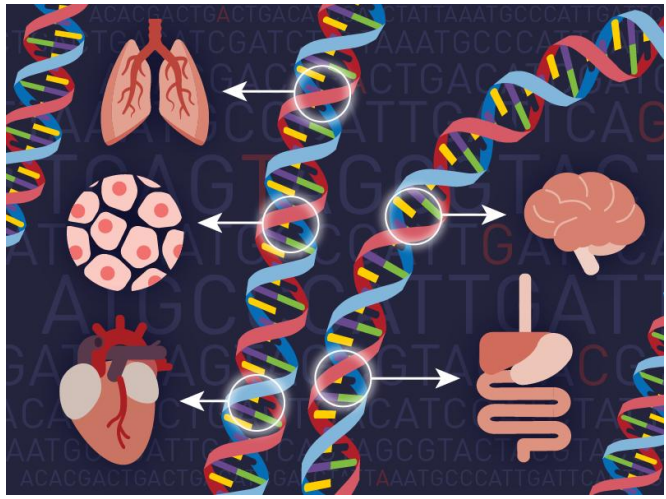
Scopo ultimo del corso: saper partire da sequenze di DNA fino ad assemblare interi genomi e caratterizzarli funzionalmente

Dal sequenziatore..

..al genoma..

..alla funzione





Scopo ultimo del corso: saper partire da sequenze di DNA fino ad assemblare interi genomi e caratterizzarli funzionalmente

Dal sequenziatore..

sequenze

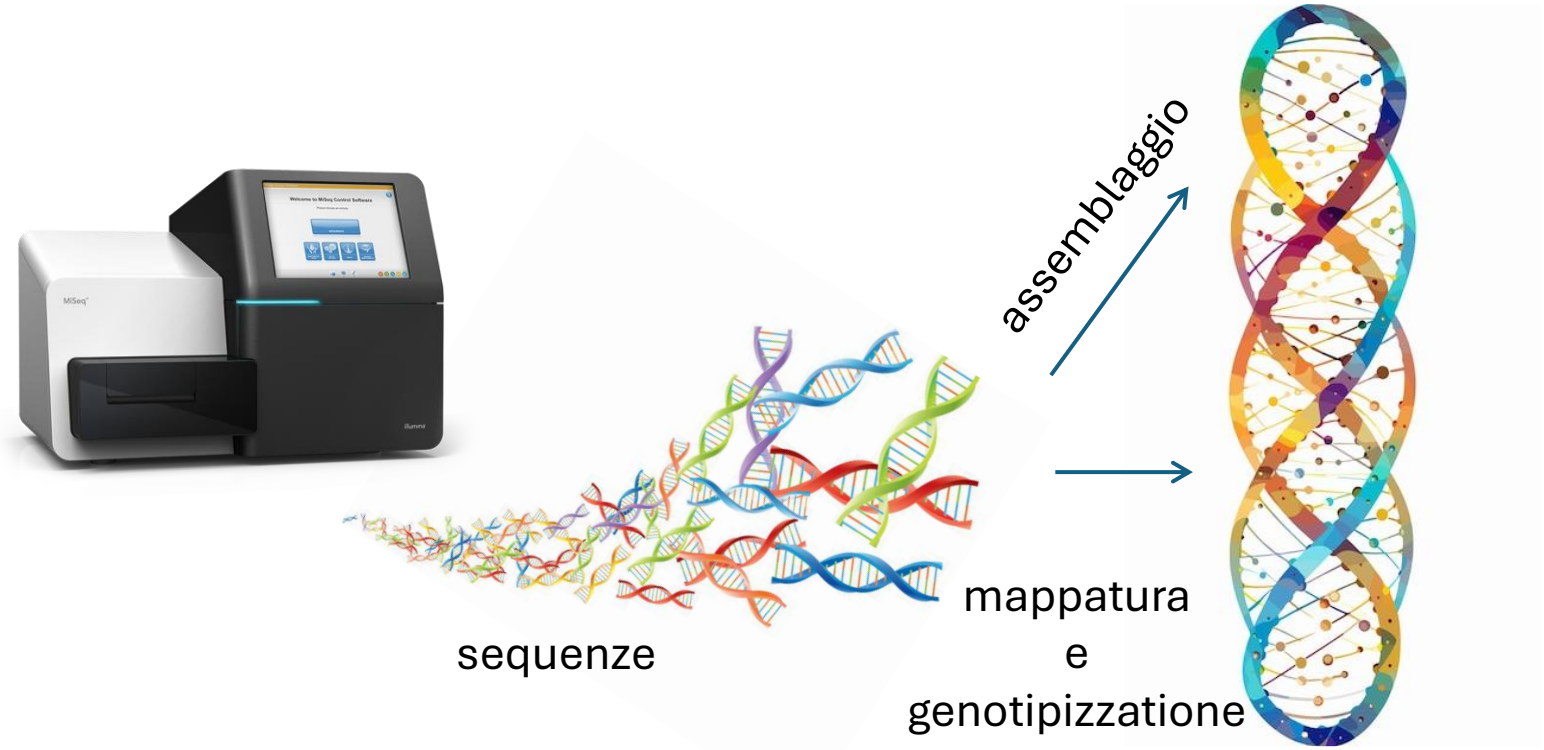
assemblaggio

..al genoma..

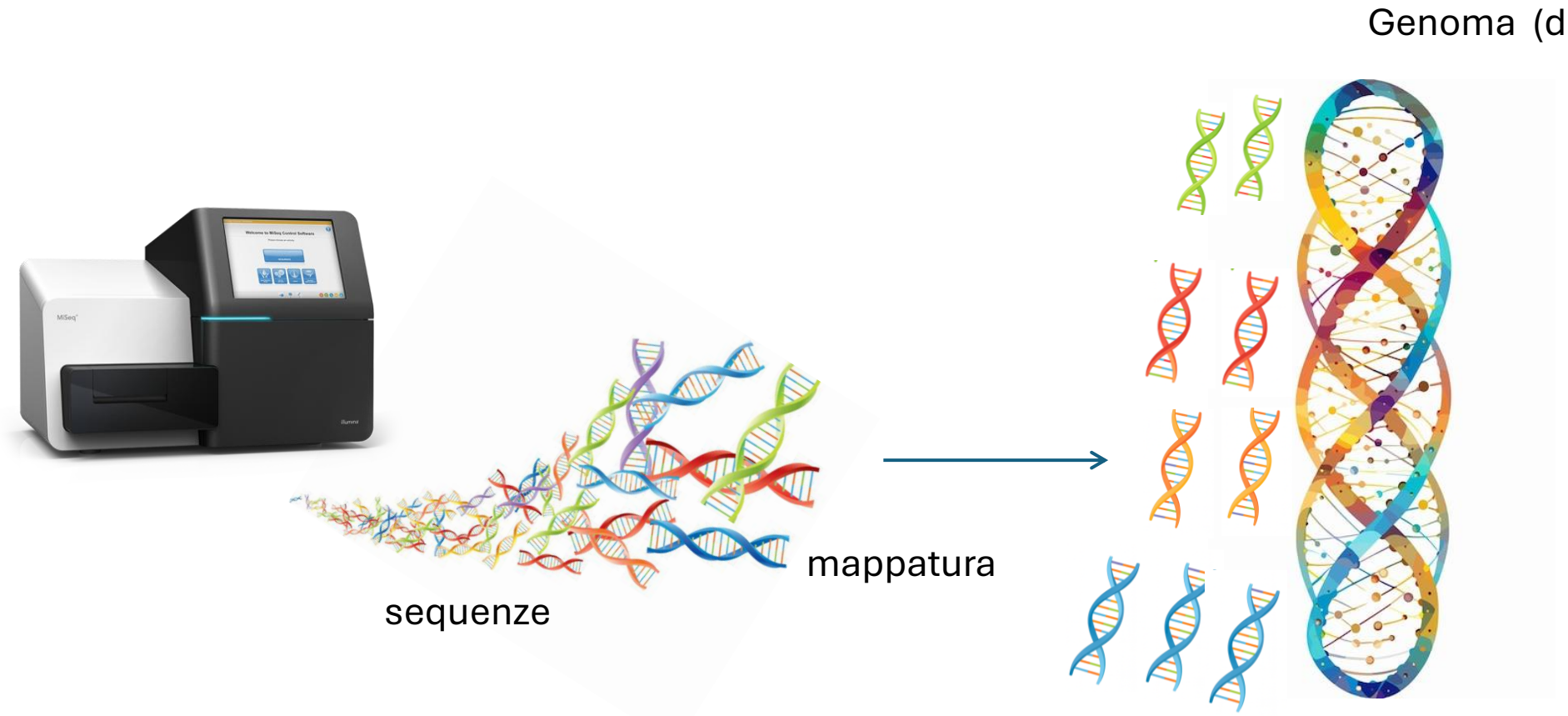
Scopo ultimo del corso: saper partire da sequenze di DNA fino ad assemblare interi genomi e caratterizzarli funzionalmente

Dal sequenziatore..

..al genoma



Mappare sequenze lungo il genoma vuol dire individuare la loro posizione originaria rispetto ad un genoma di referenza..

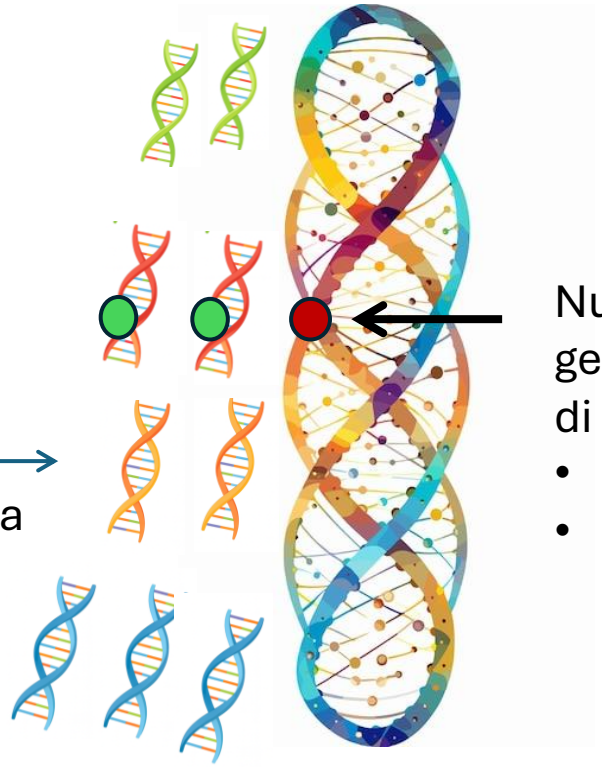


..anche tollerando potenziali differenze con il genoma di referenza

sequenze

mappatura

Genoma (di referenza)



Nucleotidi differenti tra genoma
di referenza e sequenze:

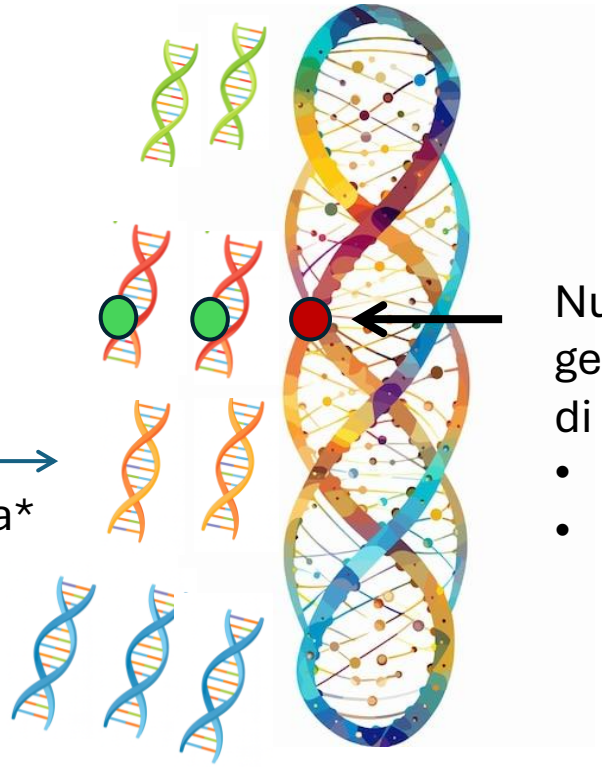
- errori
- differenze tra individui

..anche tollerando potenziali differenze con il genoma di referenza

sequenze

mappatura*

Genoma (di referenza)



Nucleotidi differenti tra genoma
di referenza e sequenze:

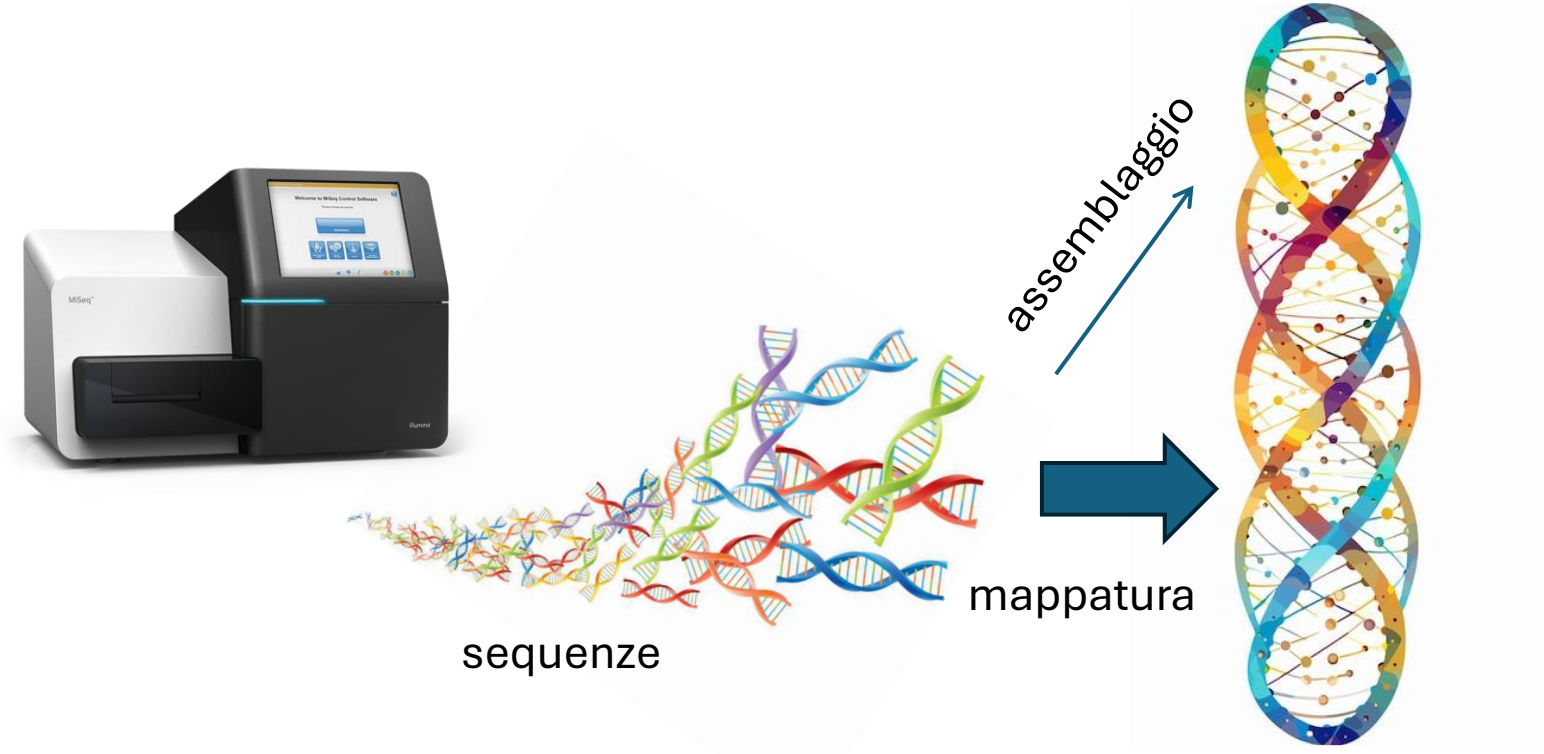
- errori
- differenze tra individui

*Il processo della caratterizzazione dei “genotipi” e delle differenze genetiche tra individui verra’ trattato nel prossimo argomento, la “**genotipizzazione**”.

La larga maggioranza delle analisi di genomica coinvolgono mappatura

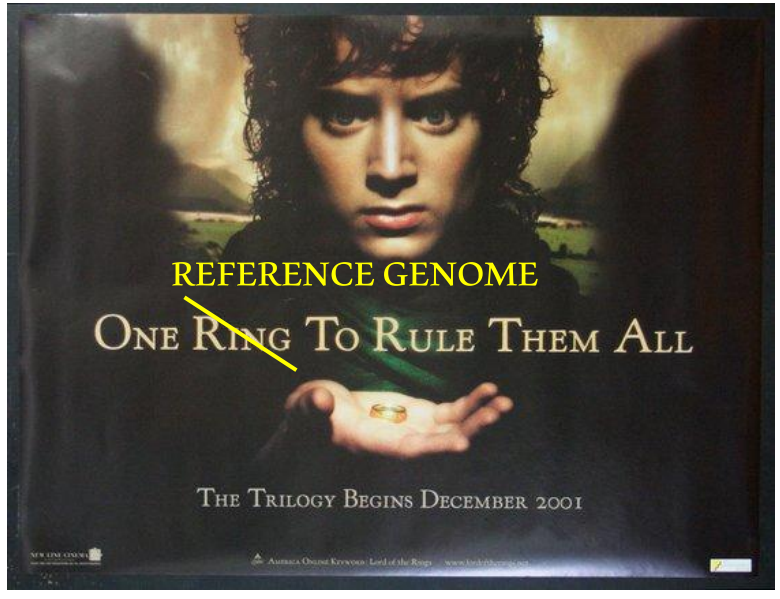
Dal sequenziatore..

..al genoma



La mappatura è molto piu' comune dell'assemblaggio nelle operazioni di genomica:

Ci basta un «genoma umano di referenza», tutti gli altri ricercatori usano quello per referenza!!

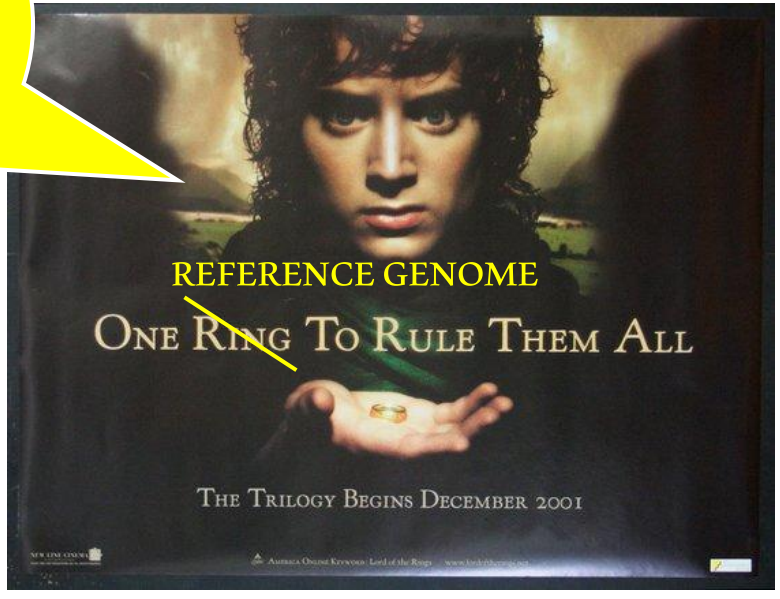


La larga maggioranza delle analisi di genomica coinvolgono mappatura

Gandalf, quali sono i rispettivi **vantaggi** e **svantaggi** di assemblaggio e mappatura?

La mappatura è molto piu' comune dell'assemblaggio nelle operazioni di genomica:

Ci basta un «genoma umano di referenza», tutti gli altri ricercatori usano quello per referenza!!

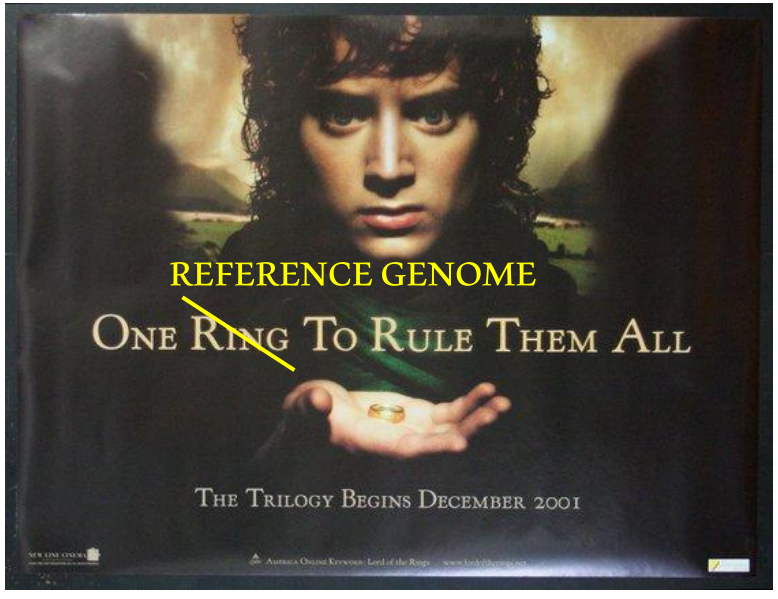


La larga maggioranza delle analisi di genomica coinvolgono mappatura

Assemblaggio	Mappatura
Necessita molte sequenze, lunghe e corte	Sono sufficienti molte meno sequenze; molto piu' economico
Comparazione delle differenze genetiche tra individui piu' complessa	Semplice e diretta comparazione delle differenze genetiche tra individui
Possibilmente de novo	Necessita di un genoma di referenza
In principio permette l'analisi di differenze tra zone complesse e ripetute del genoma, o di presenza/assenza di geni unici per individui	Limitato a zone del genoma «mappabili»* e rappresentate nel genoma di referenza (quindi non zone unicamente presenti nei nuovi individui sequenziati)

La mappatura è molto piu' comune dell'assemblaggio nelle operazioni di genomica:

Ci basta un «genoma umano di referenza», tutti gli altri ricercatori usano quello per referenza!!



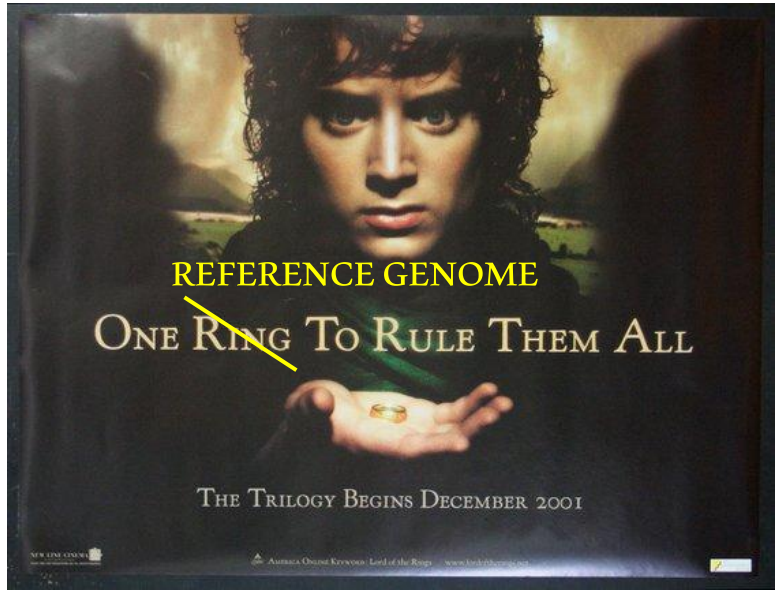
La larga maggioranza delle analisi di genomica coinvolgono mappatura

Assemblaggio	Mappatura
Necessita molte sequenze, lunghe e corte	Sono sufficienti molte meno sequenze; molto piu' economico
Comparazione delle differenze genetiche tra individui piu' complessa	Semplice e diretta comparazione delle differenze genetiche tra individui
Possibilmente de novo	Necessita di un genoma di referenza
In principio permette l'analisi di differenze tra zone complesse e ripetute del genoma, o di presenza/assenza di geni unici per individui	Limitato a zone del genoma «mappabili»* e rappresentate nel genoma di referenza (quindi non zone unicamente presenti nei nuovi individui sequenziati)

*Il piu' delle volte le zone non-ripetute e mappabili sono comunque le piu' informative in quanto codificanti per geni e piu' semplici da analizzare.

La mappatura è molto piu' comune dell'assemblaggio nelle operazioni di genomica:

Ci basta un «genoma umano di referenza», tutti gli altri ricercatori usano quello per referenza!!

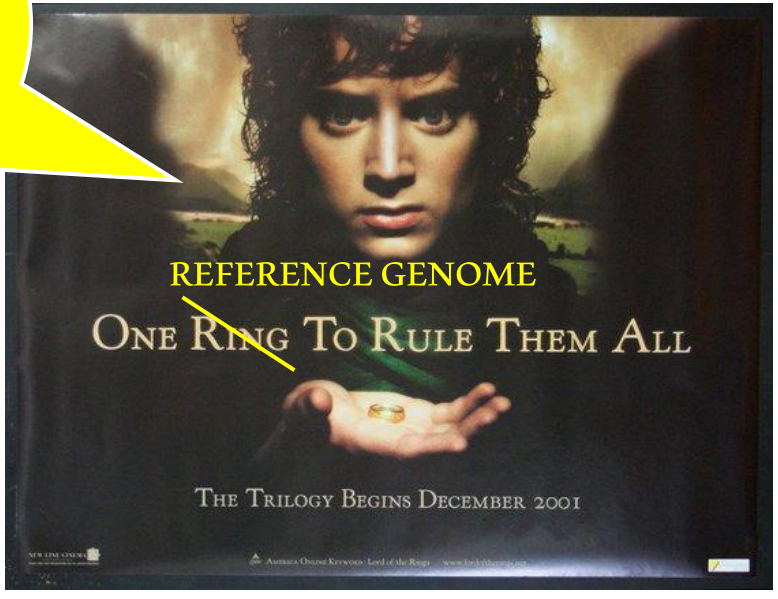


La larga maggioranza delle analisi di genomica coinvolgono mappatura

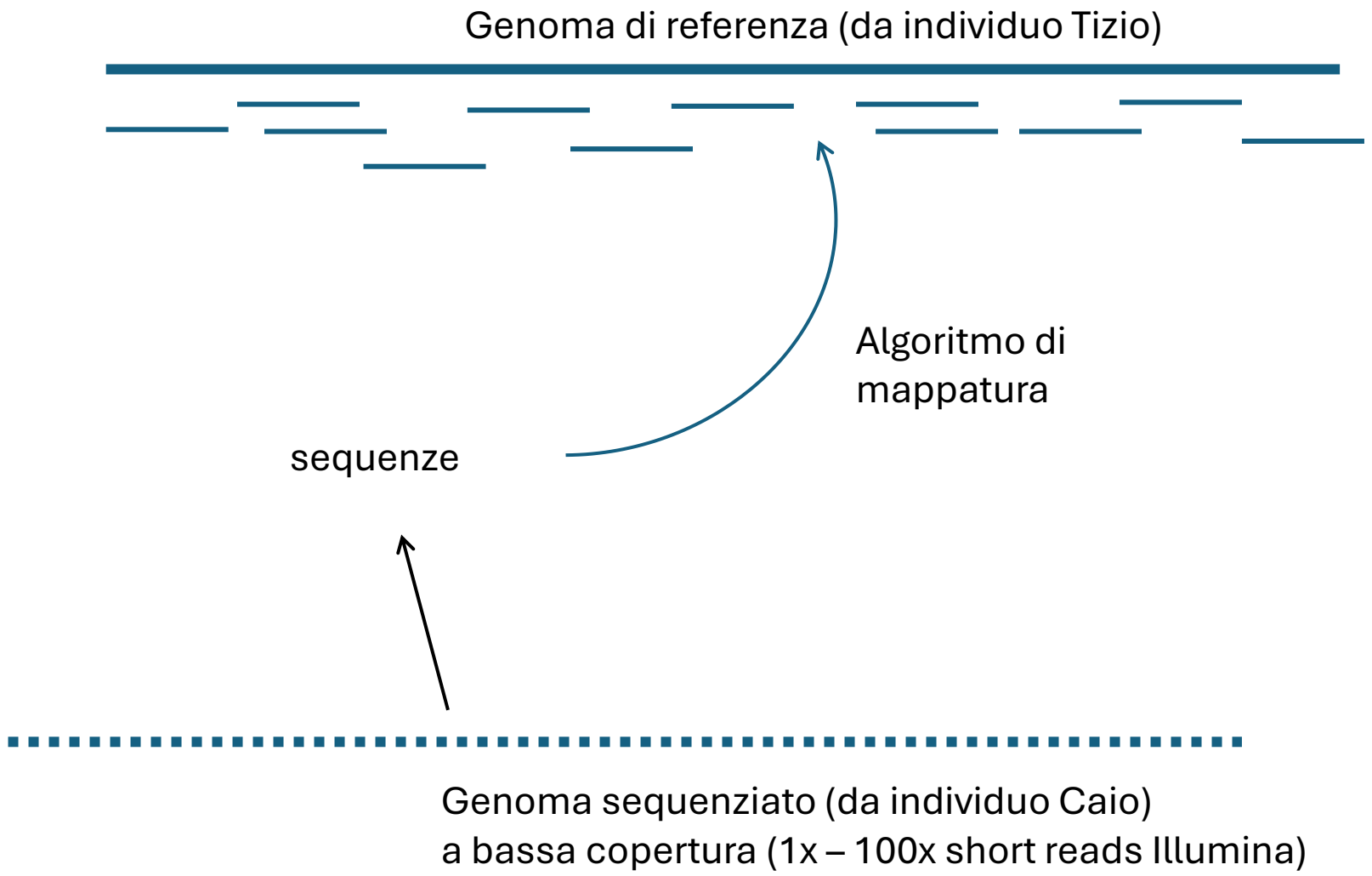
La mappatura è molto piu' comune dell'assemblaggio nelle operazioni di genomica:

Ci basta un «genoma umano di referenza», tutti gli altri ricercatori usano quello per referenza!!

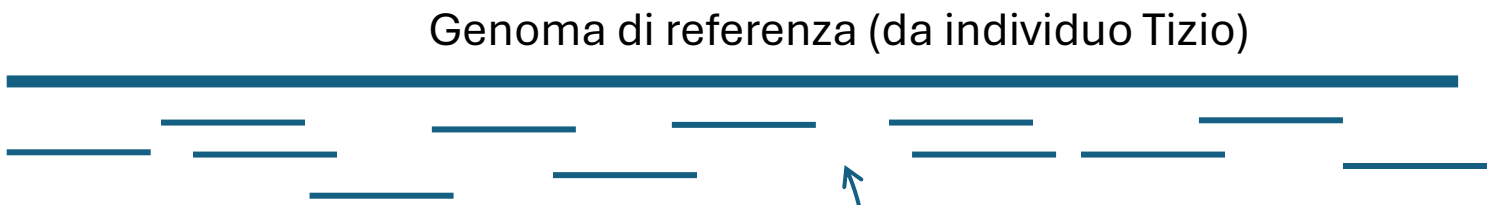
E allora Gandalf, quali sono gli utilizzi della mappatura?



Una mappatura di sequenze tipica

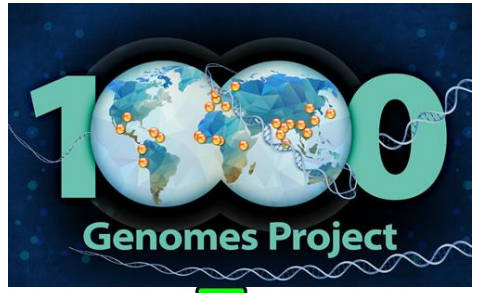


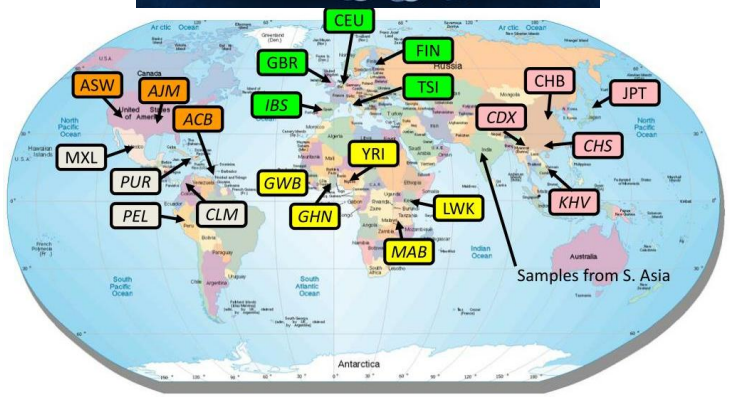
Una mappatura di sequenze tipica



Algoritmo di mappatura

sequenze





Genoma sequenziato (da individuo Caio)
con 1x – 100x short reads Illumina
(persino a bassa copertura < 10x)

Economico!

Per uno o migliaia di individui!

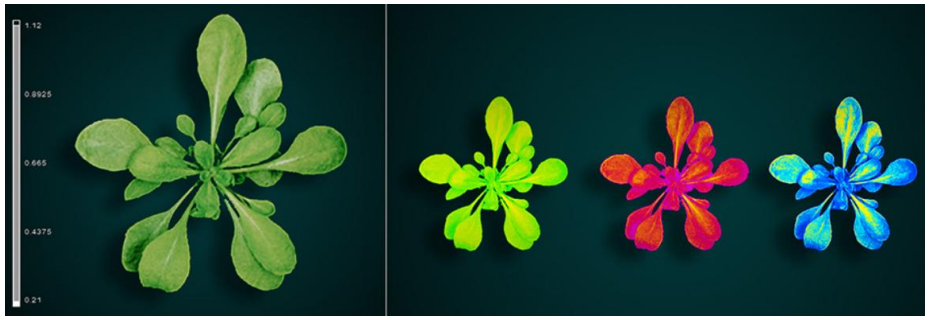
Referenza			
…	… …	… …	… …
C	C C	C C	C C
C	C C	C C	C C
A	A A	A A	A A
T	**C** T	**C** T	T T
G	G G	G G	G G
C	C C	C C	C C
A	A A	A A	A A
C	C C	C C	C C
G	G G	G G	**T** G
A	A A	A A	A A
T	T T	T T	T T
A	A A	A A	A A
C	C C	C C	C C
T	T T	T T	T T
A	**T** A	**T** A	A A
T	T T	T T	T T
T	T T	T T	T T
T	T T	T T	T T
C	C C	C C	C C
C	C C	C C	C C
…	… …	… …	… …

Mappare rapidamente milioni di sequenze ad un genoma di referenza ha permesso la ricostruzione (economica) di genomi individuali e di ampi datasets di migliaia di genomi per i) studi clinici e individuare varianti patogeniche, ii) studi di popolazione ed evolutivi, iii) screening medico

Referenza						
						
…	…	…	…	…	…	…
C	C	C	C	C	C	C
C	C	C	C	C	C	C
A	A	A	A	A	A	A
T	**C**	T	**C**	T	T	T
G	G	G	G	G	G	G
C	C	C	C	C	C	C
A	A	A	A	A	A	A
C	C	C	C	C	C	C
G	G	G	G	G	**T**	G
A	A	A	A	A	A	A
T	T	T	T	T	T	T
A	A	A	A	A	A	A
C	C	C	C	C	C	C
T	T	T	T	T	T	T
A	**T**	A	**T**	A	A	A
T	T	T	T	T	T	T
T	T	T	T	T	T	T
T	T	T	T	T	T	T
C	C	C	C	C	C	C
C	C	C	C	C	C	C
…	…	…	…	…	…	…

Mappare rapidamente milioni di sequenze ad un genoma di referenza ha permesso la ricostruzione (economica) di genomi individuali e di ampi datasets di migliaia di genomi per i) studi clinici e individuare varianti patogeniche, ii) studi di popolazione ed evolutivi, iii) screening medico

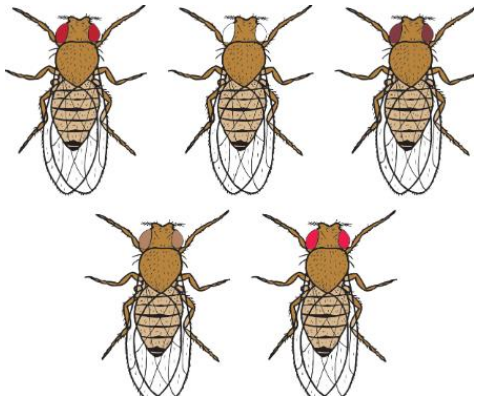
1,135 Genomes Reveal the Global Pattern of Polymorphism in *Arabidopsis thaliana*



gnomAD

Genome Aggregation Database

Drosophila Genetic Reference Panel



biobank uk

Improving the health of future generations

500.000 individui con tanto di dati fenotipici e medici

Article

Sequence diversity lost in early pregnancy

deCODE genetics

https://doi.org/10.1038/s41586-025-09031-w

Received: 16 July 2024

Accepted: 16 April 2025

Published online: 21 May 2025

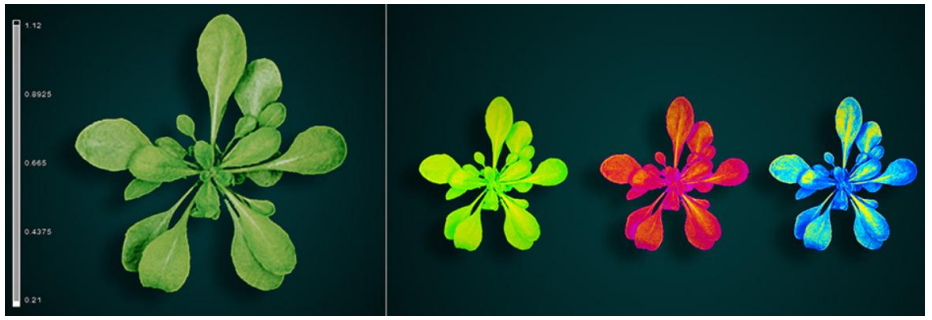
Gudny A. Arnadottir[1,19], Hakon Jonsson[1,19] ✉, Tanja Schlaikjær Hartwig[2,19], Jennifer R. Gruhn[3,19], Peter Loof Møller[4], Arnaldur Gylfason[1], David Westergaard[2], Andrew Chi-Ho Chan[3], Asmundur Oddsson[1], Lilja Stefansdottir[1], Louise le Roux[1], Valgerdur Steinthorsdottir[1], Kristjan H. Swerford Moore[1], Sigurgeir Olafsson[1], Pall I. Olason[1], Hannes P. Eggertsson[1], Gísli H. Halldórsson[1], G. Bragi Walters[1], Hreinn Stefansson[1], Sigurjon A. Gudjonsson[1], Gunnar Palsson[1], Brynjar O. Jensson[1], Run Fridriksdottir[1], Jesper Friis Petersen[3,5,6],

Referenza						
…	…	…	…	…	…	…
C	C	C	C	C	C	C
C	C	C	C	C	C	C
A	A	A	A	A	A	A
T	**C**	T	**C**	T	T	T
G	G	G	G	G	G	G
C	C	C	C	C	C	C
A	A	A	A	A	A	A
C	C	C	C	C	C	C
G	G	G	G	G	**T**	G
A	A	A	A	A	A	A
T	T	T	T	T	T	T
A	A	A	A	A	A	A
C	C	C	C	C	C	C
T	T	T	T	T	T	T
A	**T**	A	**T**	A	A	A
T	T	T	T	T	T	T
T	T	T	T	T	T	T
T	T	T	T	T	T	T
C	C	C	C	C	C	C
C	C	C	C	C	C	C
…	…	…	…	…	…	…

Mappare rapidamente milioni di sequenze ad un genoma di referenza ha permesso la ricostruzione (economica) di genomi individuali e di ampi datasets di migliaia di genomi per i) studi clinici e individuare varianti patogeniche, ii) studi di popolazione ed evolutivi, iii) screening medico

E questi sono una minuscola parte!!!

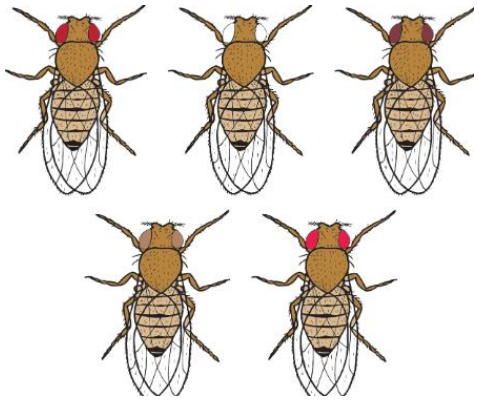
1,135 Genomes Reveal the Global Pattern of Polymorphism in *Arabidopsis thaliana*



gnomAD

Genome Aggregation Database

Drosophila Genetic Reference Panel



biobank uk

Improving the health of future generations

500.000 individui con tanto di dati fenotipici e medici

Article

Sequence diversity lost in early pregnancy

deCODE genetics

https://doi.org/10.1038/s41586-025-09031-w
Received: 16 July 2024
Accepted: 16 April 2025
Published online: 21 May 2025

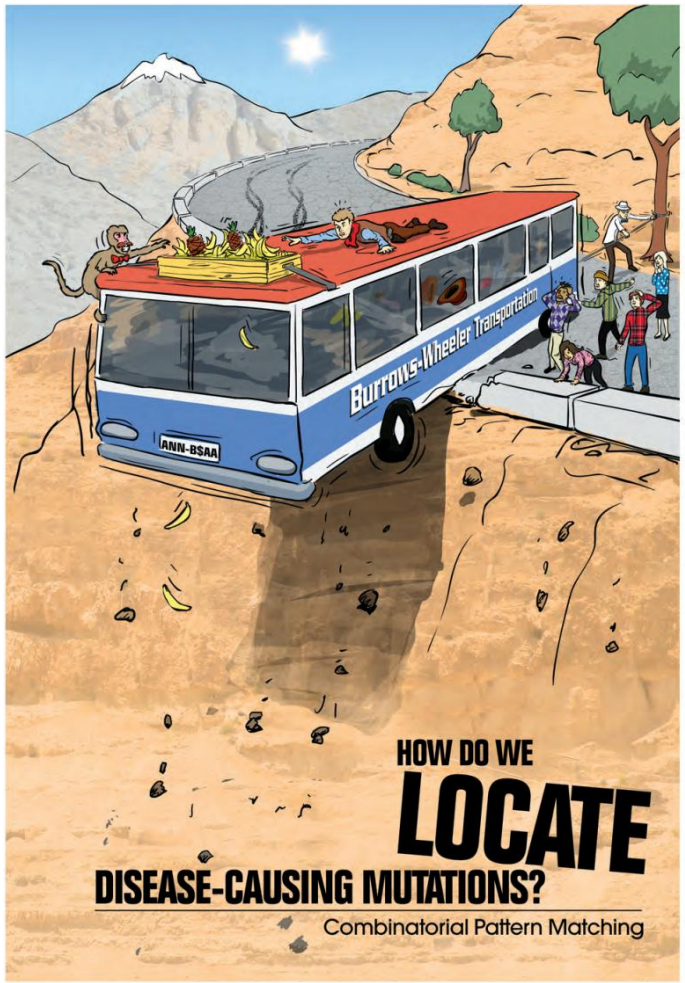
Gudny A. Arnadottir[1,19], Hakon Jonsson[1,19] ✉, Tanja Schlaikjær Hartwig[2,19], Jennifer R. Gruhn[3,19], Peter Loof Møller[4], Arnaldur Gylfason[1], David Westergaard[2], Andrew Chi-Ho Chan[3], Asmundur Oddsson[1], Lilja Stefansdottir[1], Louise le Roux[1], Valgerdur Steinthorsdottir[1], Kristjan H. Swerford Moore[1], Sigurgeir Olafsson[1], Pall I. Olason[1], Hannes P. Eggertsson[1], Gísli H. Halldórsson[1], G. Bragi Walters[1], Hreinn Stefansson[1], Sigurjon A. Gudjonsson[1], Gunnar Palsson[1], Brynjar O. Jensson[1], Run Fridriksdottir[1], Jesper Friis Petersen[3,5,6],

Referenza			
…	… …	… …	… …
C	C C	C C	C C
C	C C	C C	C C
A	A A	A A	A A
T	**C** T	**C** T	T T
G	G G	G G	G G
C	C C	C C	C C
A	A A	A A	A A
C	C C	C C	C C
G	G G	G G	**T** G
A	A A	A A	A A
T	T T	T T	T T
A	A A	A A	A A
C	C C	C C	C C
T	T T	T T	T T
A	**T** A	**T** A	A A
T	T T	T T	T T
T	T T	T T	T T
T	T T	T T	T T
C	C C	C C	C C
C	C C	C C	C C
…	… …	… …	… …

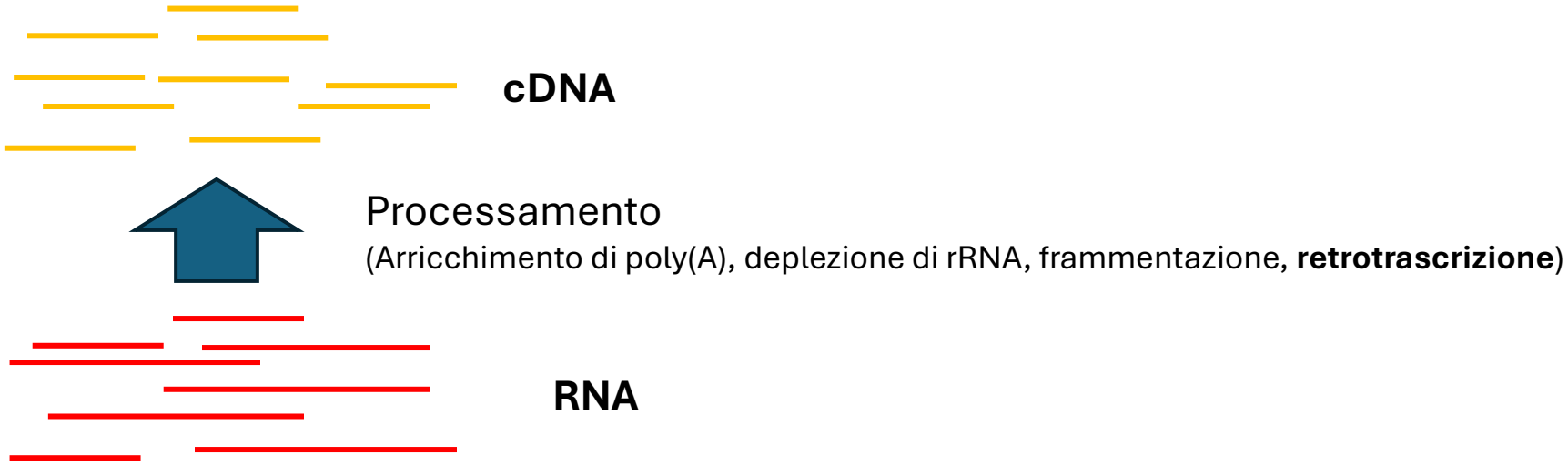
Prime due proprietà fondamentali di un buon algoritmo per mappare:

- Veloce e capace di gestire milioni di sequenze con poche risorse computazionali
- Deve poter tollerare piccole differenze tra individui

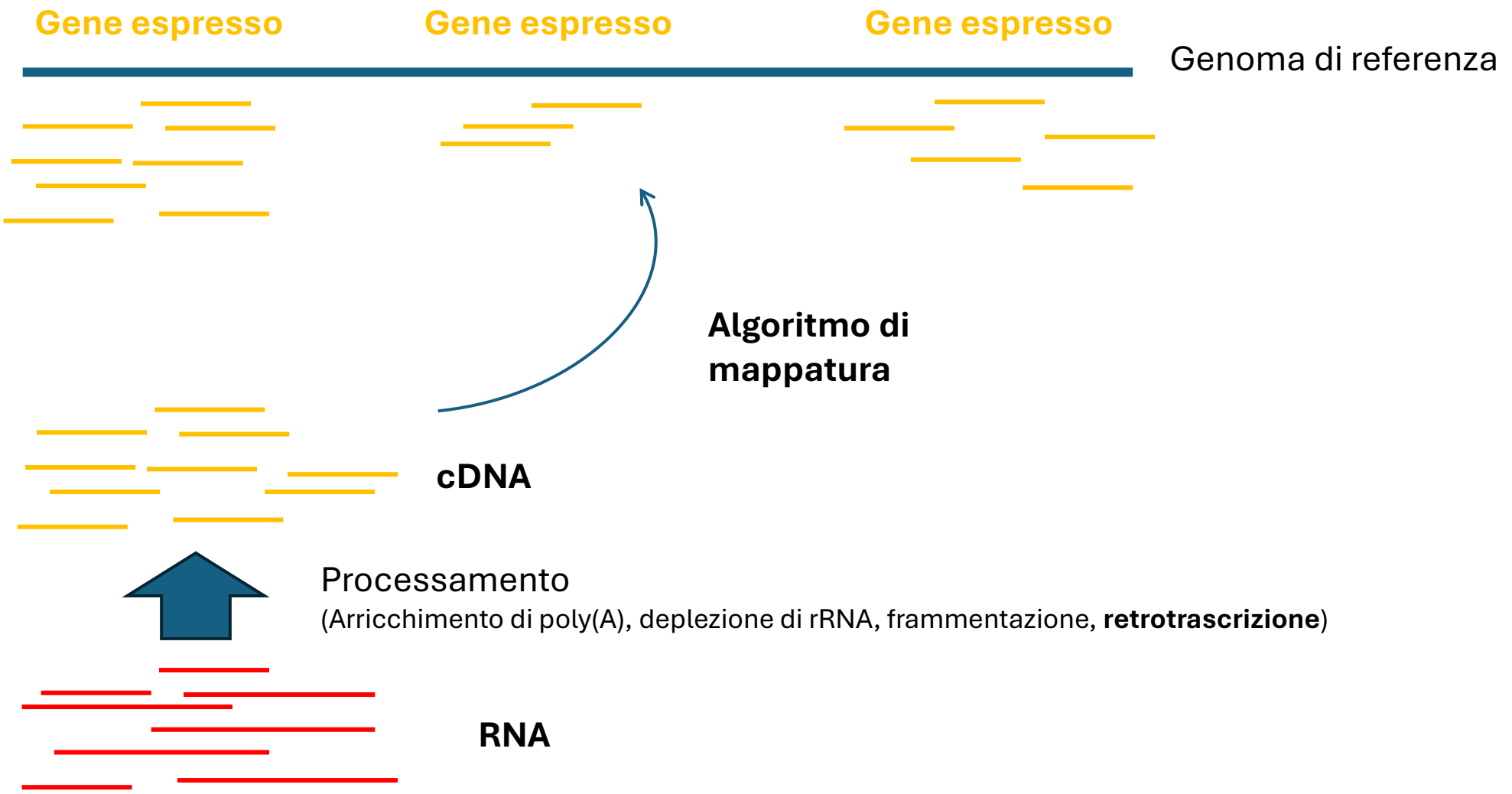
Per le patologie genetiche sono proprio le differenze che ci interessano!



Studio dell’espressione genica (RNAseq)

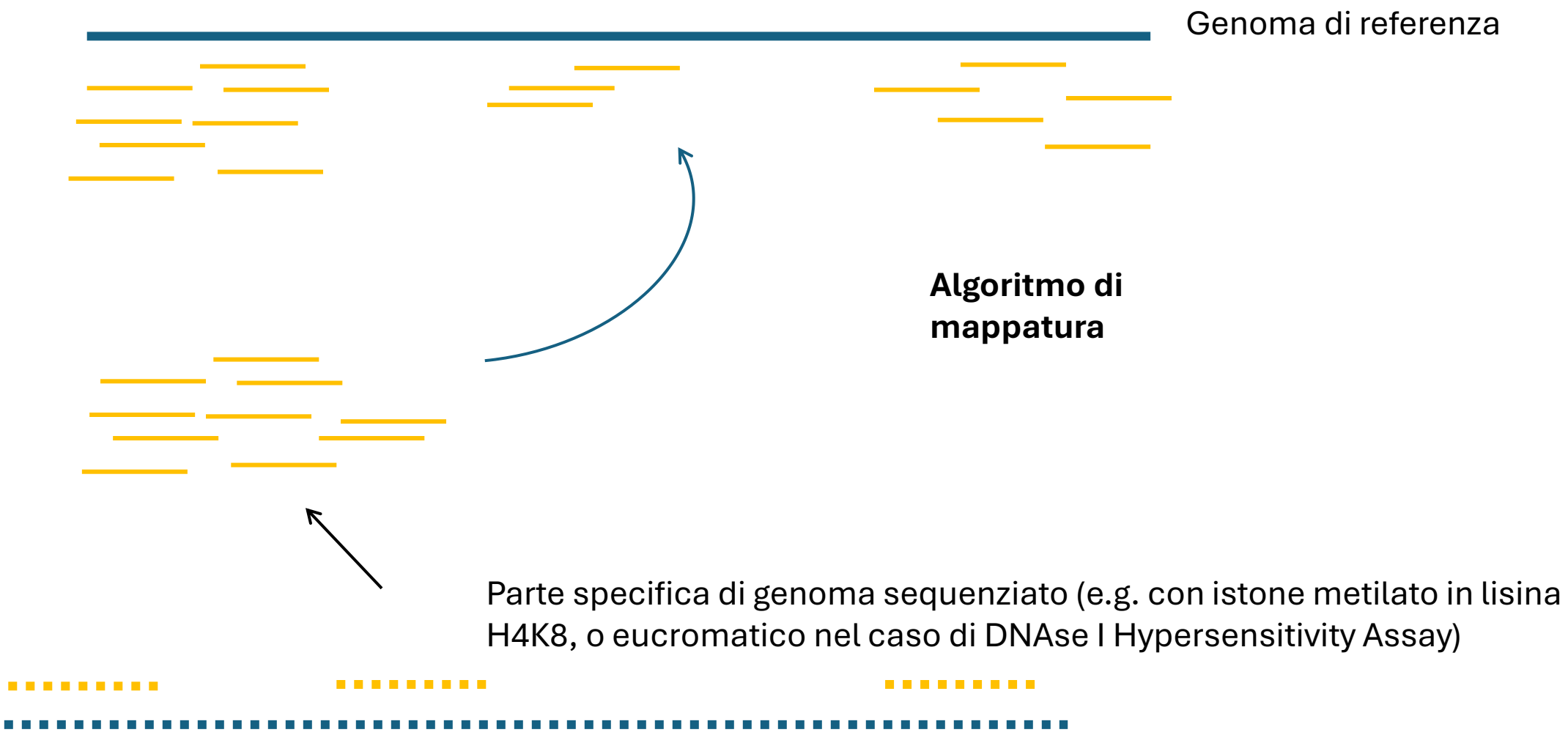


Studio dell'espressione genica (RNAseq)



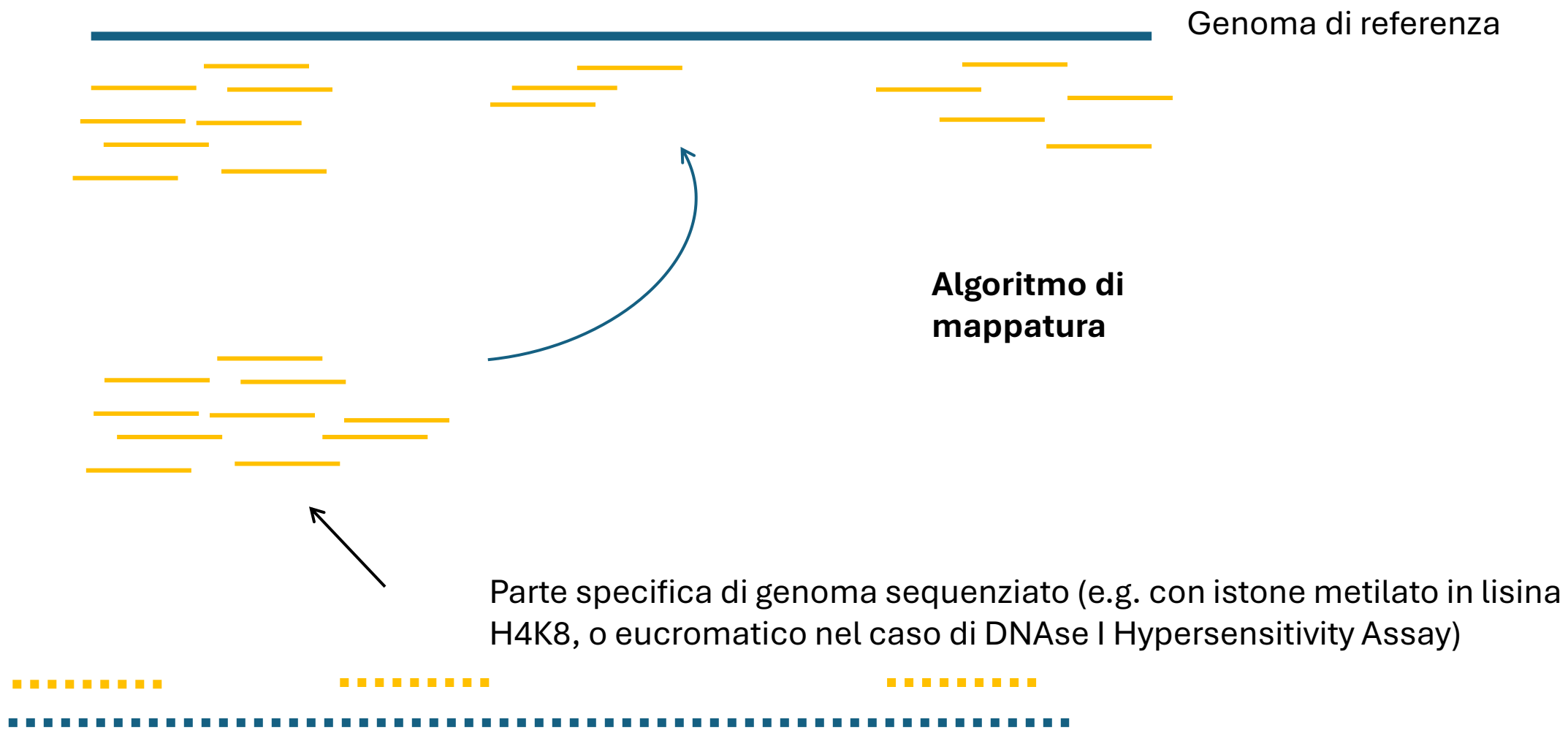
Chip-seq (Chromatin ImmunoPrecipitation followed by sequencing)

Identificazione di zone del genoma con modificazioni specifiche della cromatina o legate da un fattore di trascrizione



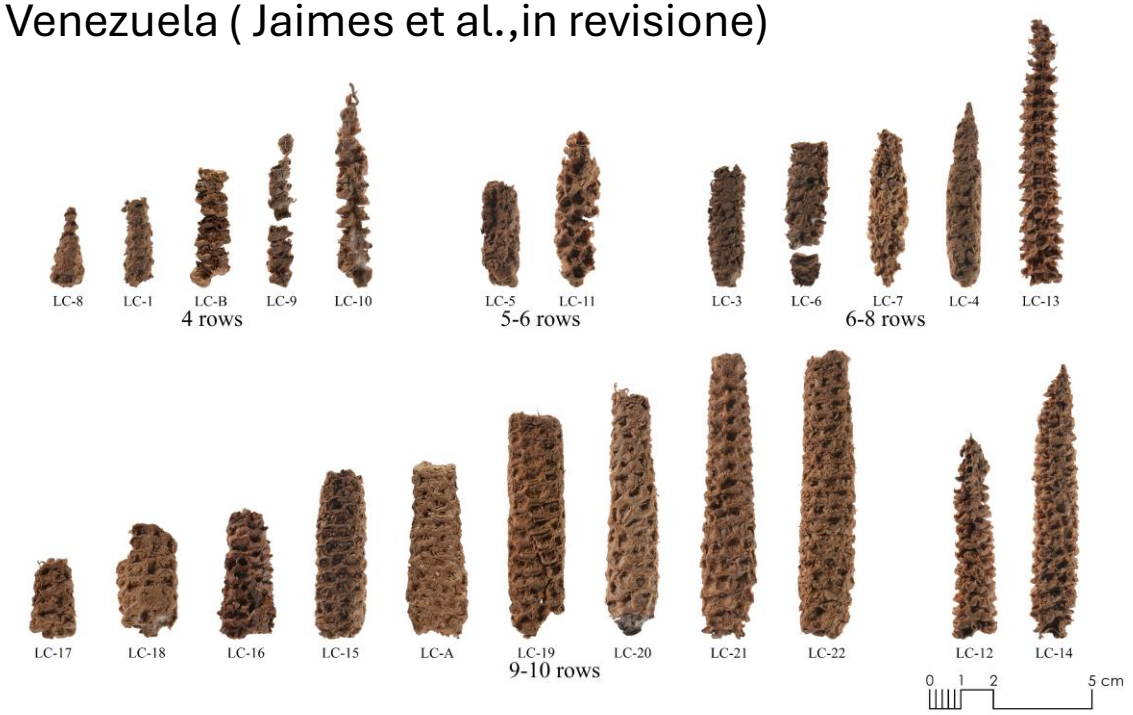
Chip-seq (Chromatin ImmunoPrecipitation followed by sequencing)

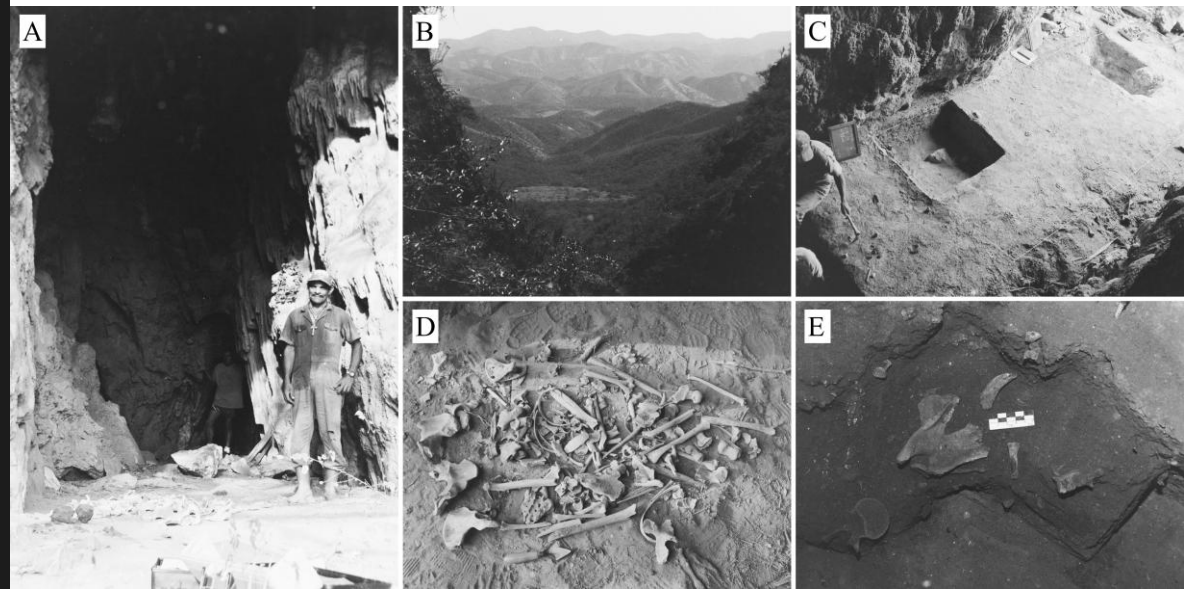
Identificazione di zone del genoma con modificazioni specifiche della cromatina o legate da un fattore di trascrizione



Sequenze da dei rachidi probabilmente di mais di 2000 anni fa dal Venezuela (Jaimes et al.,in revisione)

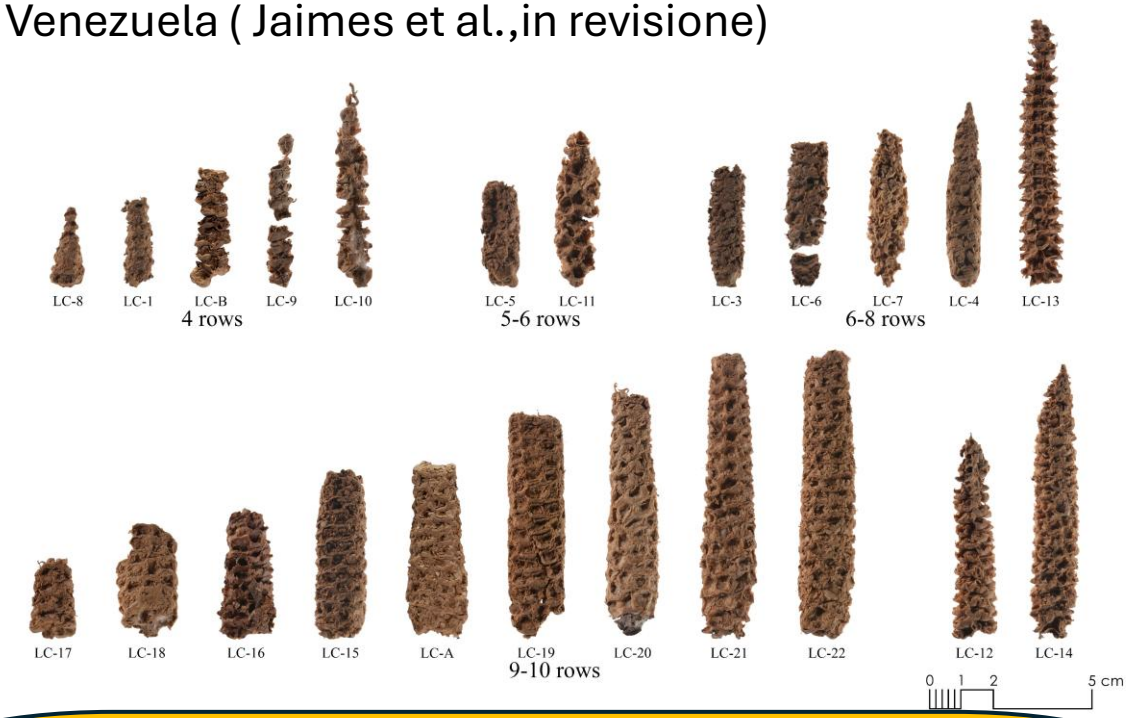
```
~$ ./create_report_sequences.sh report.txt
~$ cat report.txt
================
sequences1.fastq
----------------
number of reads
250
number of high quality (F) bases
22881
number of adaptors
9
================
sequences2.fastq
----------------
number of reads
250
number of high quality (F) bases
23064
number of adaptors
3
================
```





Sequenze da dei rachidi probabilmente di mais di 2000 anni fa dal Venezuela (Jaimes et al.,in revisione)

```
~$ ./create_report_sequences.sh report.txt
~$ cat report.txt
================
sequences1.fastq
----------------
number of reads
250
number of high quality (F) bases
22881
number of adaptors
9
================
sequences2.fastq
----------------
number of reads
250
number of high quality (F) bases
23064
number of adaptors
3
================
```



1) Vengo davvero da mais? Oppure da qualche altra pianta simile (competitive mapping)? E c'è anche DNA umano?

Competitive mapping

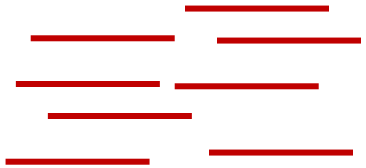
Genomi di referenza

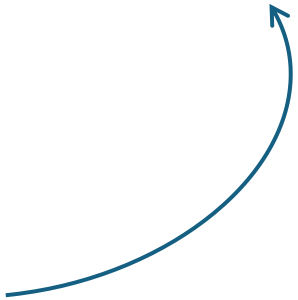
specie A

specie B

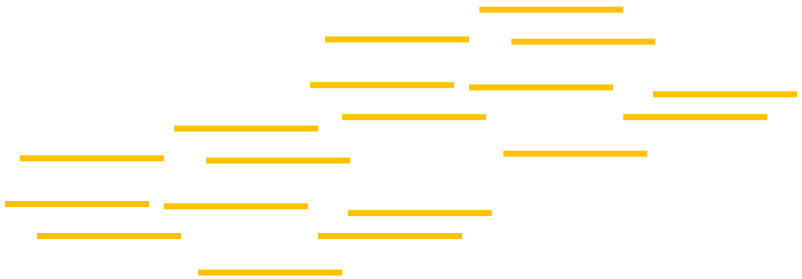
specie C

....





Algoritmo di mappatura



Campione ambientale da suolo, ambiente, organismi non identificati

Competitive mapping

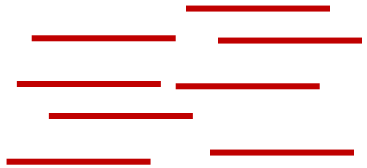
Genomi di referenza

specie A

specie B

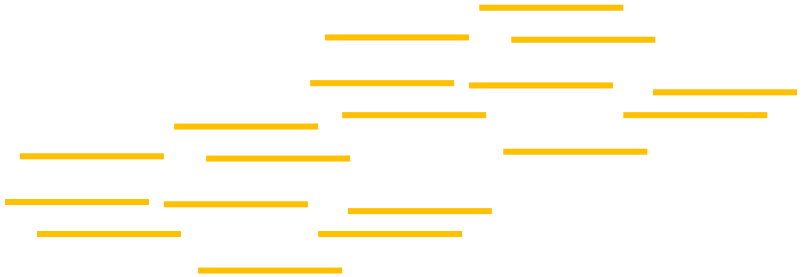
specie C

....



Algoritmo di mappatura

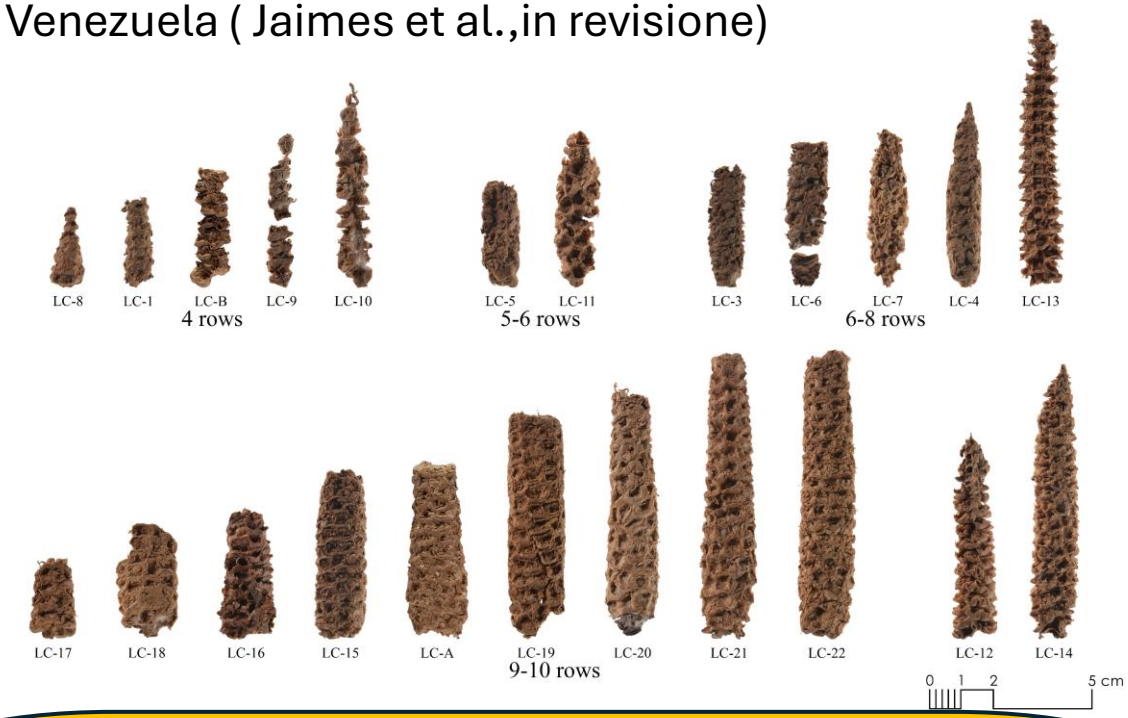
In questo caso il campione sembra provenire da specie A, con qualche traccia di B



Campione ambientale da suolo, ambiente, organismi non identificati

Sequenze da dei rachidi probabilmente di mais di 2000 anni fa dal Venezuela (Jaimes et al.,in revisione)

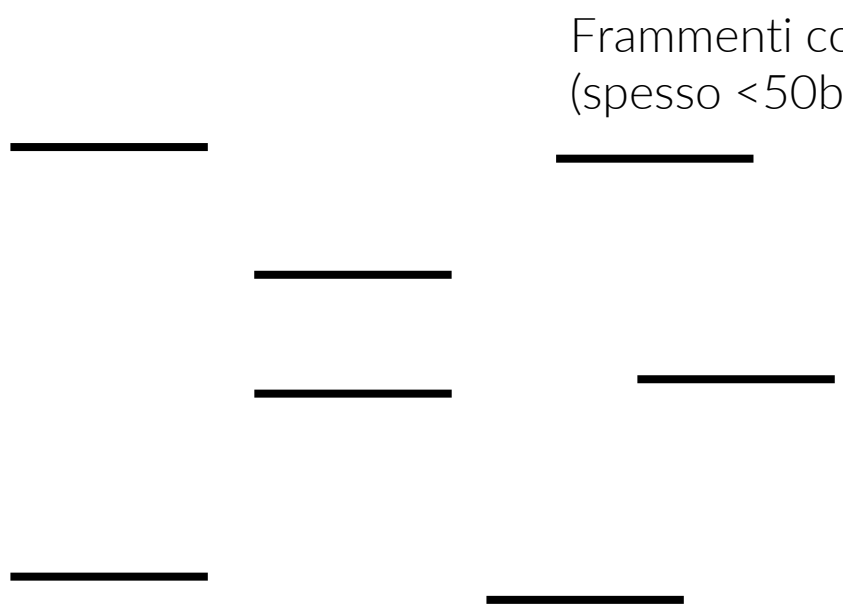
```
~$ ./create_report_sequences.sh report.txt
~$ cat report.txt
================
sequences1.fastq
----------------
number of reads
250
number of high quality (F) bases
22881
number of adaptors
9
================
sequences2.fastq
----------------
number of reads
250
number of high quality (F) bases
23064
number of adaptors
3
================
```



1) Vengo davvero da mais? Oppure da qualche altra pianta simile (competitive mapping)? E c'è anche DNA umano?
2) Sono antico, o vengo da una delle tortillas di Marcelo?

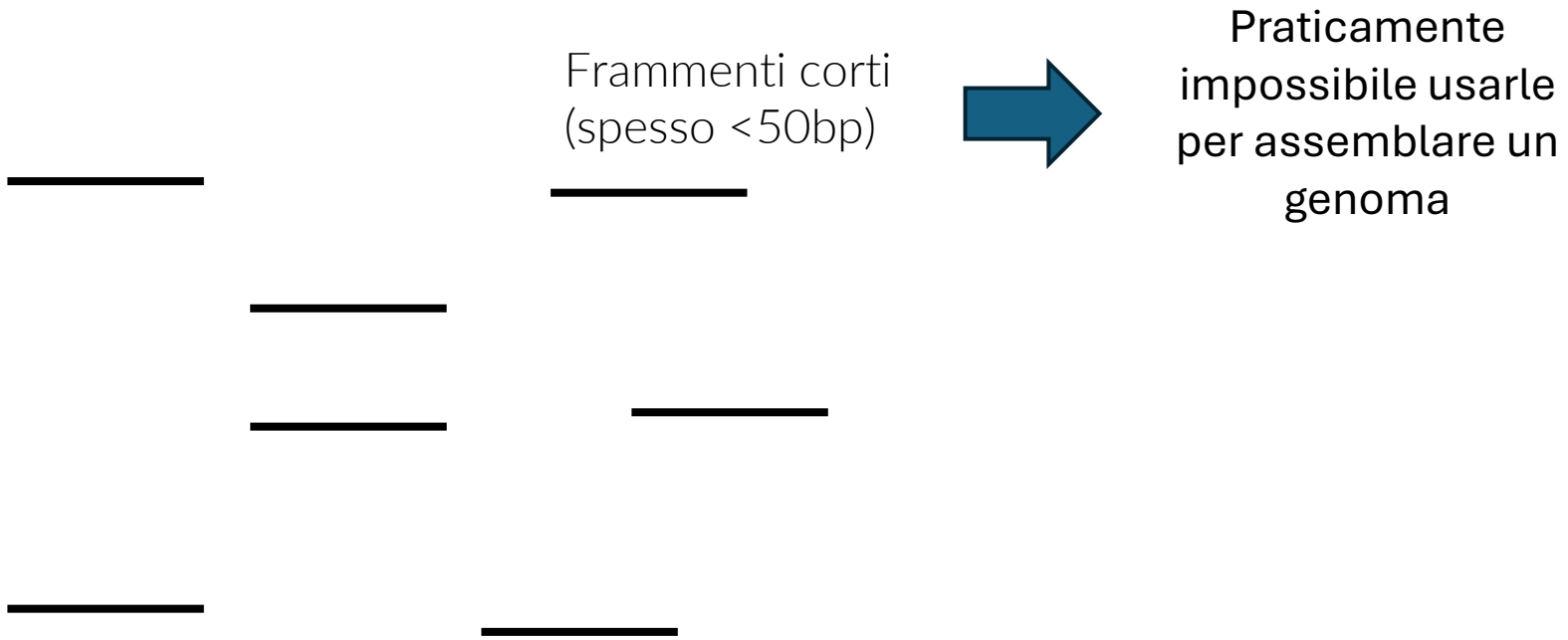
Caratteristiche del DNA antico

DNA endogeno (mais antico)



Caratteristiche del DNA antico

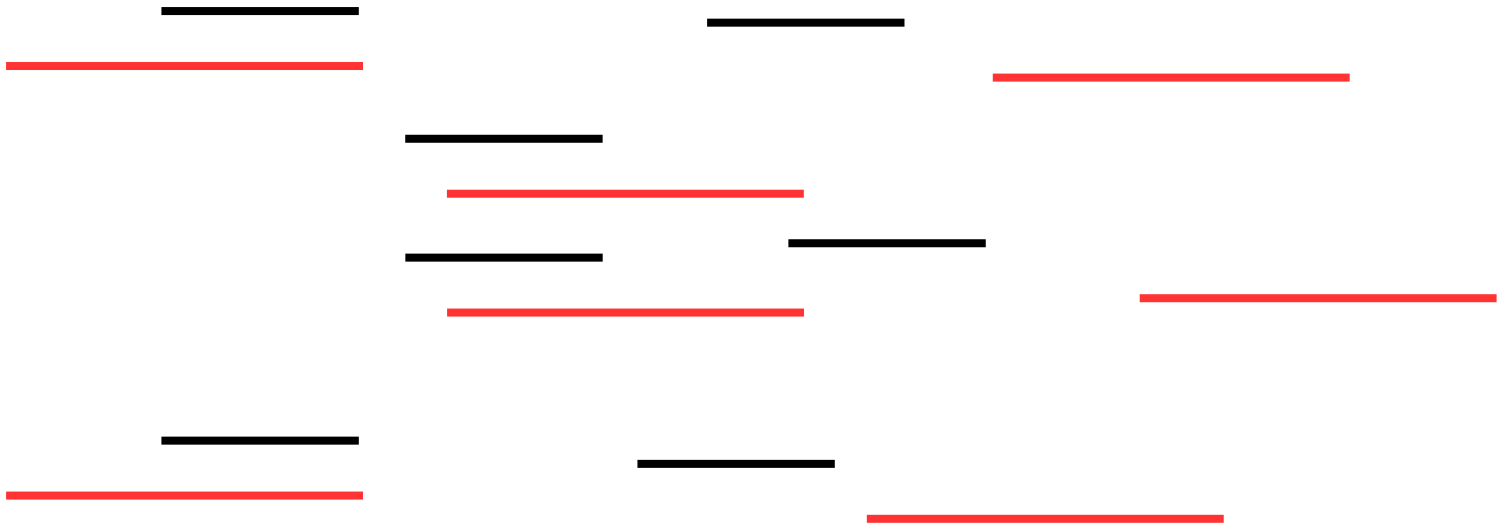
DNA endogeno (mais antico)



Caratteristiche del DNA antico

DNA endogeno (mais antico)

DNA contaminante (tortilla di Marcelo)

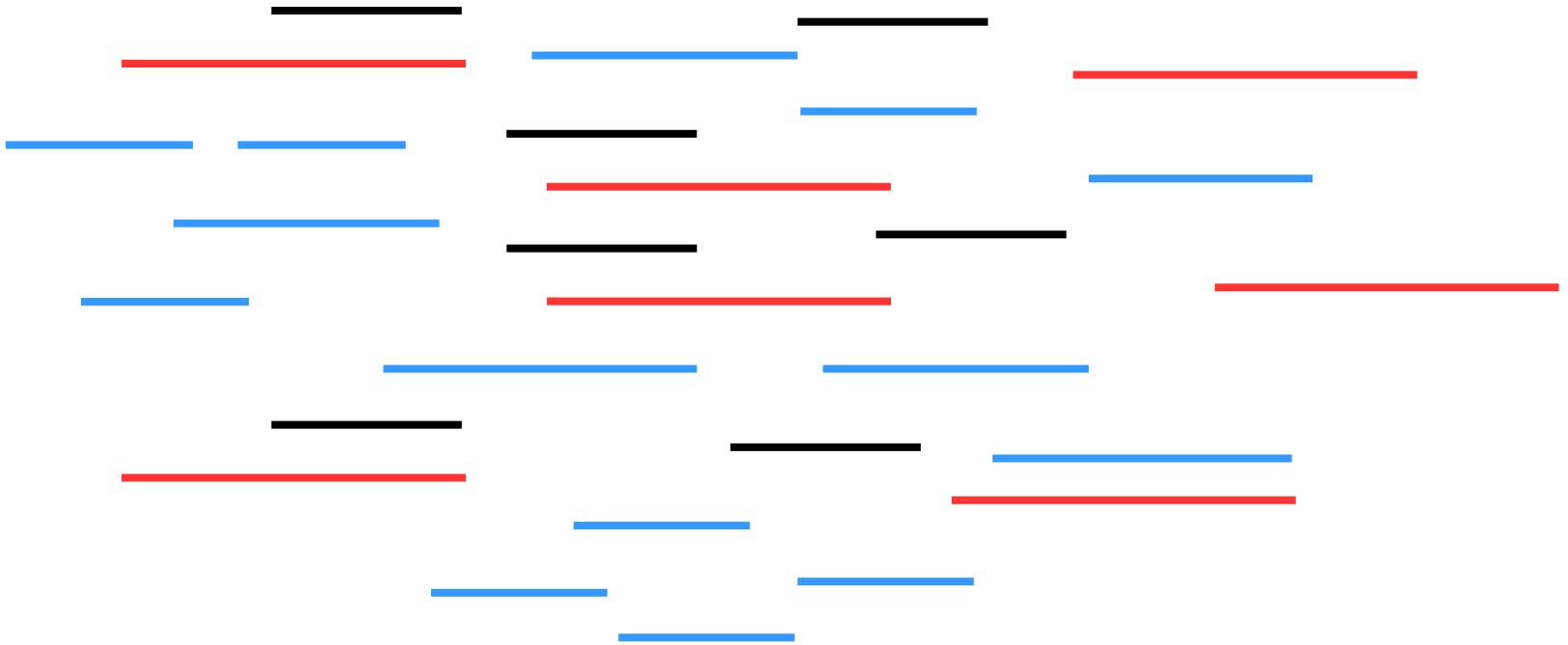


Caratteristiche del DNA antico

DNA endogeno (mais antico)

DNA contaminante (tortilla di Marcelo)

DNA batterico (ed altro)

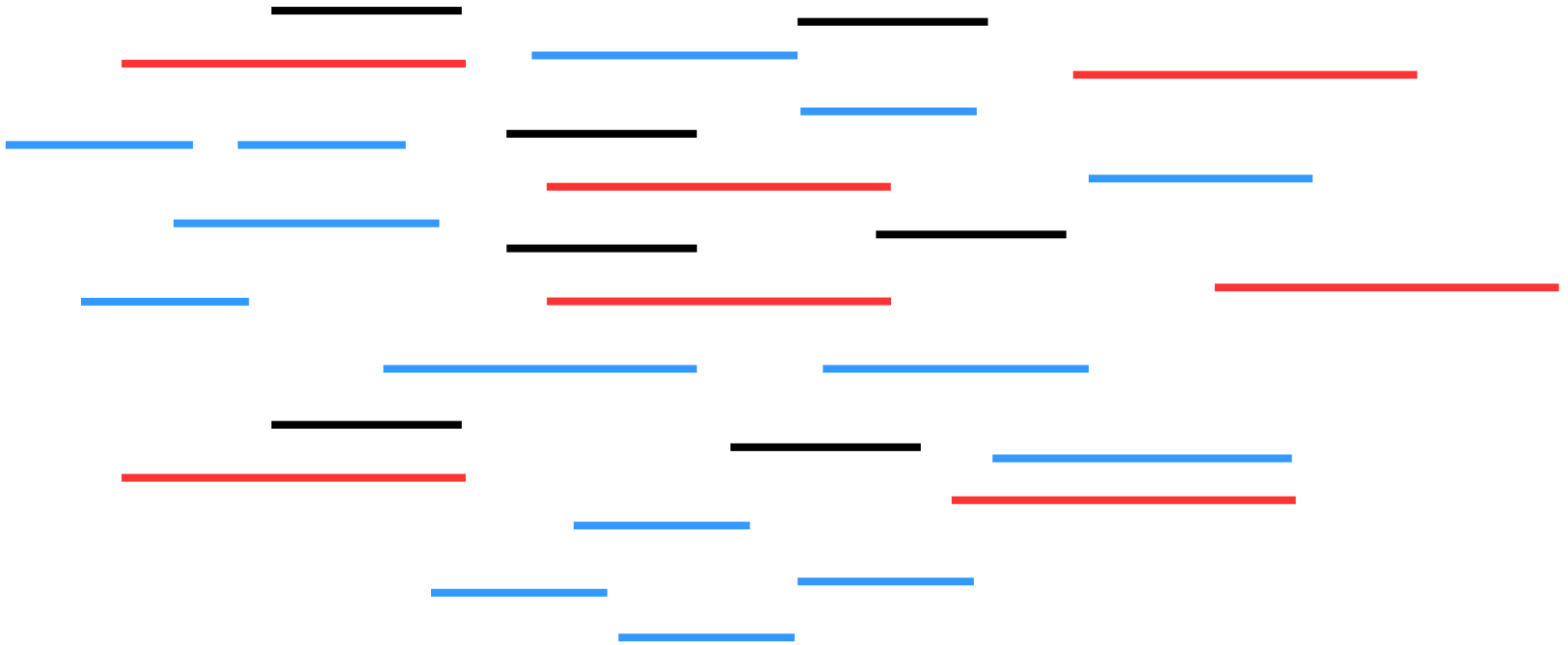


Caratteristiche del DNA antico

DNA endogeno (Neandertal)

DNA contaminante (*H.sapiens*)

DNA batterico (ed altro)

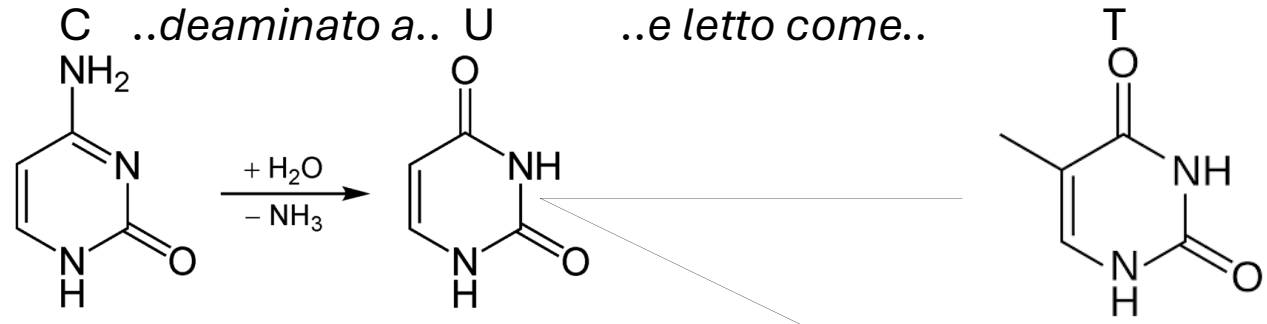


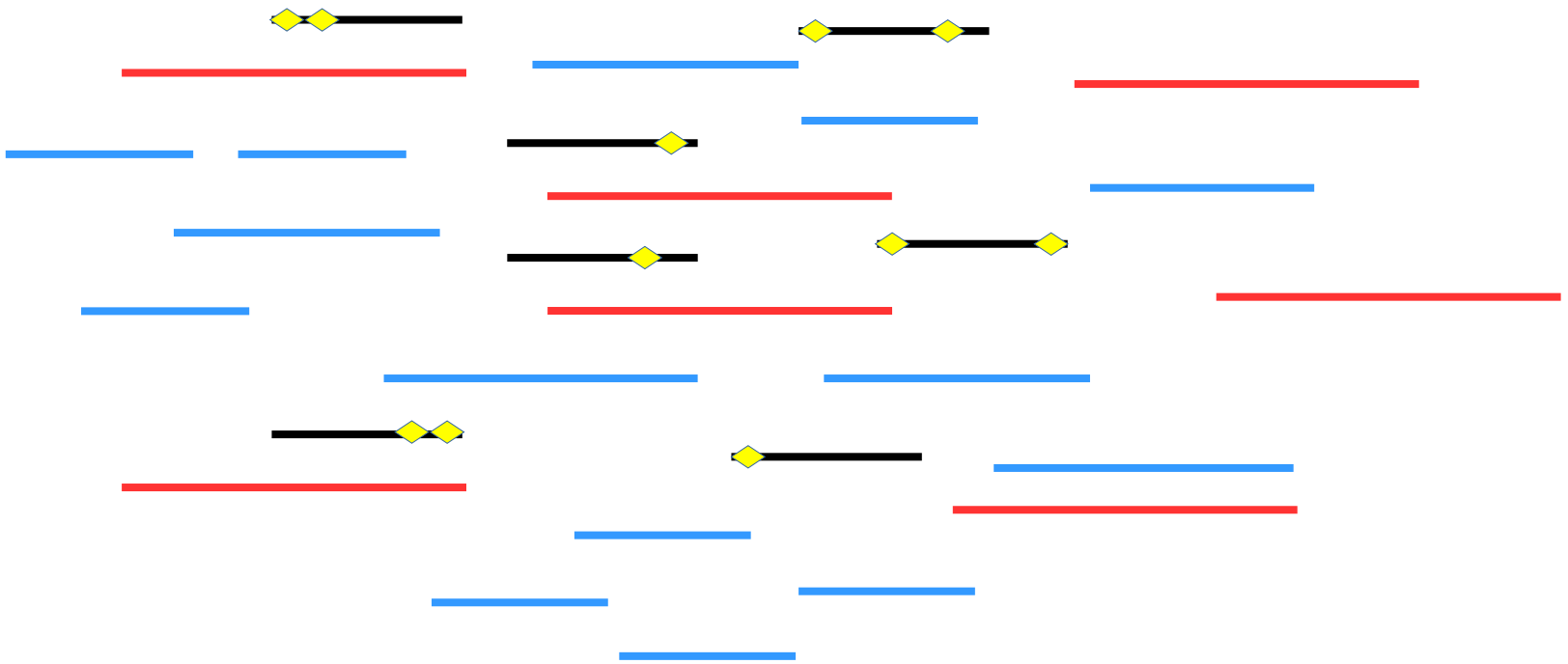
Caratteristiche del DNA antico

DNA endogeno (Neandertal)

DNA contaminante (*H.sapiens*)

DNA batterico (ed altro)





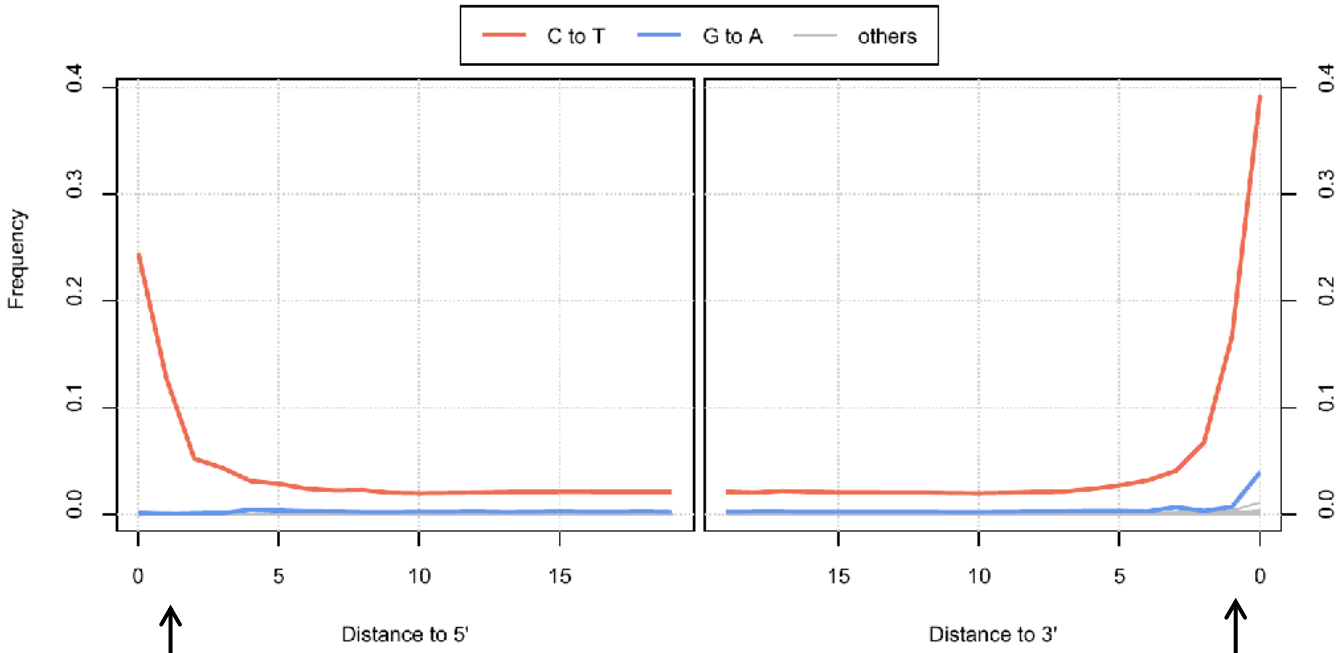
Referenza		
…	…	…
C	**T**	C
C	**T**	C
A	A	A
T	T	T
C	**T**	C
C	C	C
A	A	A
C	C	**G**
G	G	G
A	A	A
T	T	T
A	**G**	A
C	C	C
T	T	T
A	A	A
T	T	T
T	T	T
T	T	T
C	C	C
C	C	C
…	…	…

5’ (inizio sequenza)

Legenda:
Deaminazioni C->T in DNA antico
Altre sostituzioni

Deaminazione del DNA antico

Proporzione di posizioni per le quali la referenza ha una C e il campione una T

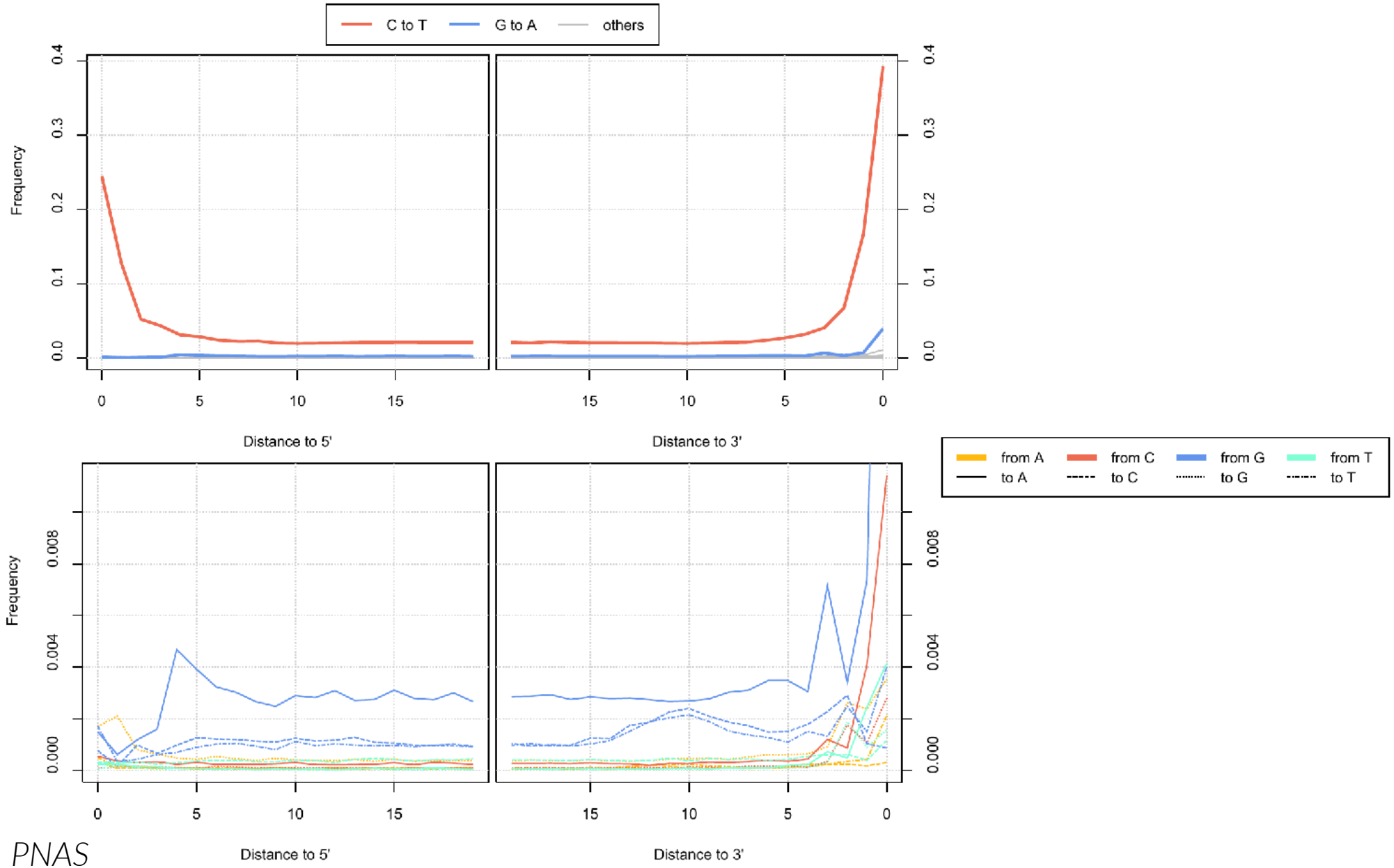


Inizio della sequenza

Fine della sequenza

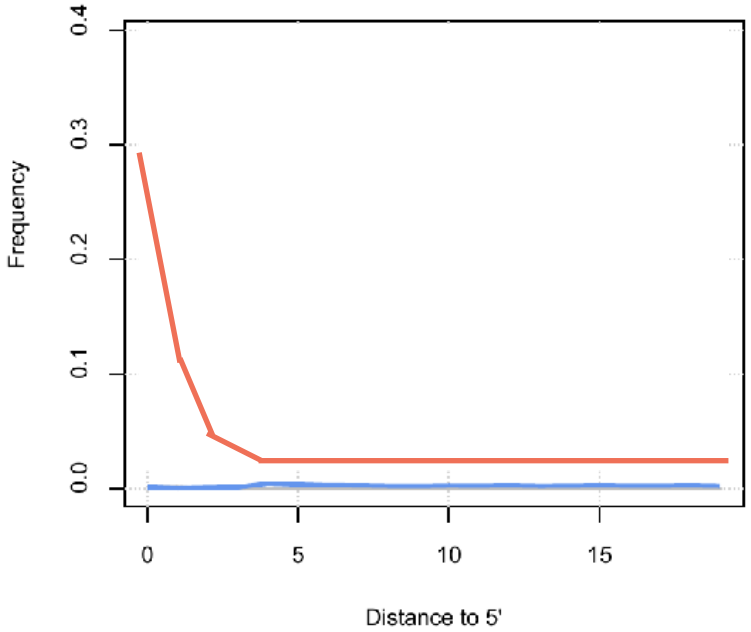
Mafessoni et al., 2020, *PNAS*

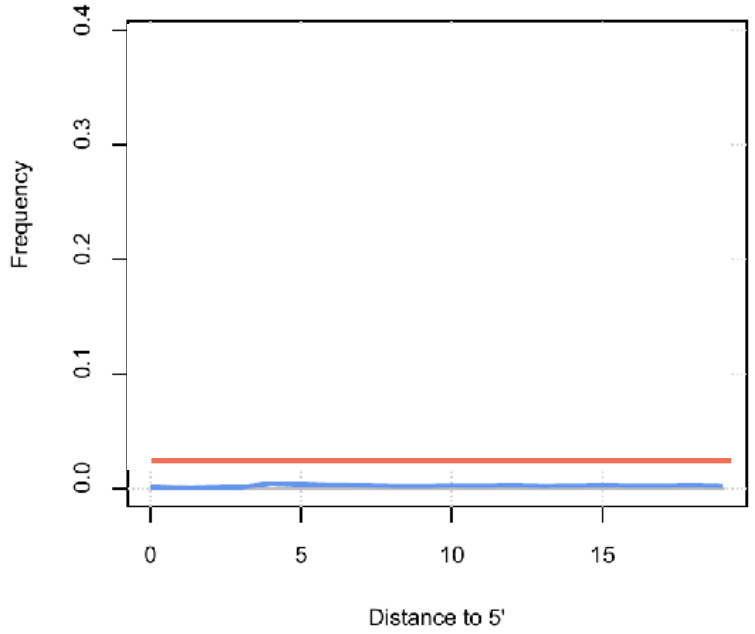
Deaminazione del DNA antico



Mafessoni et al., 2020, *PNAS*

La deaminazione permette di riconoscere il DNA antico



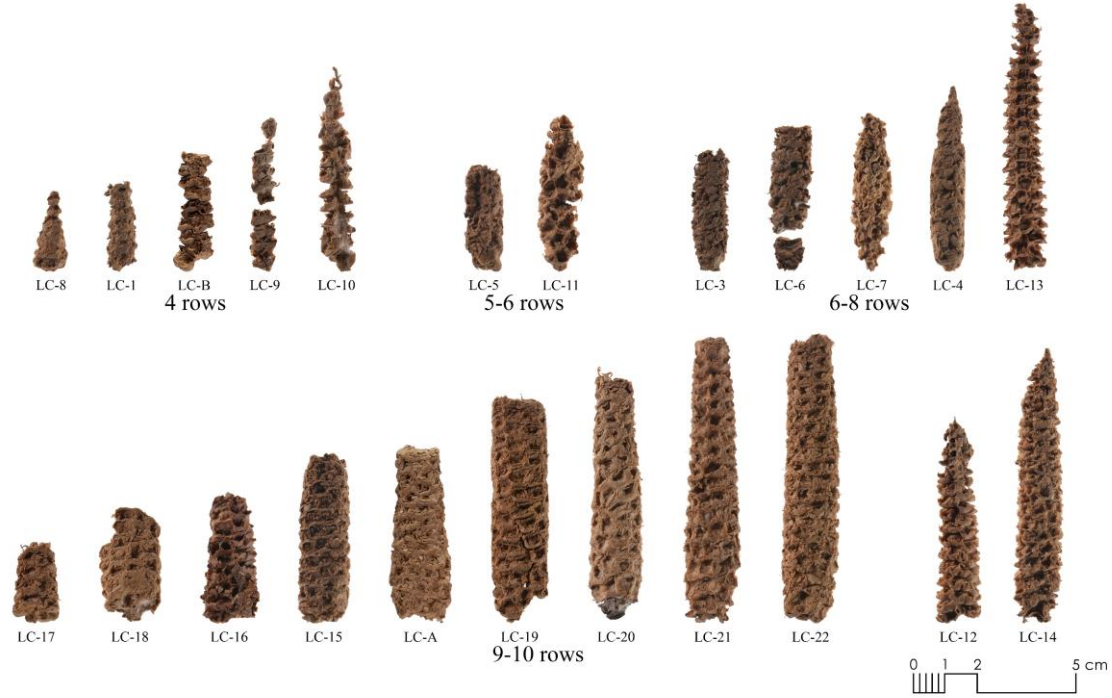


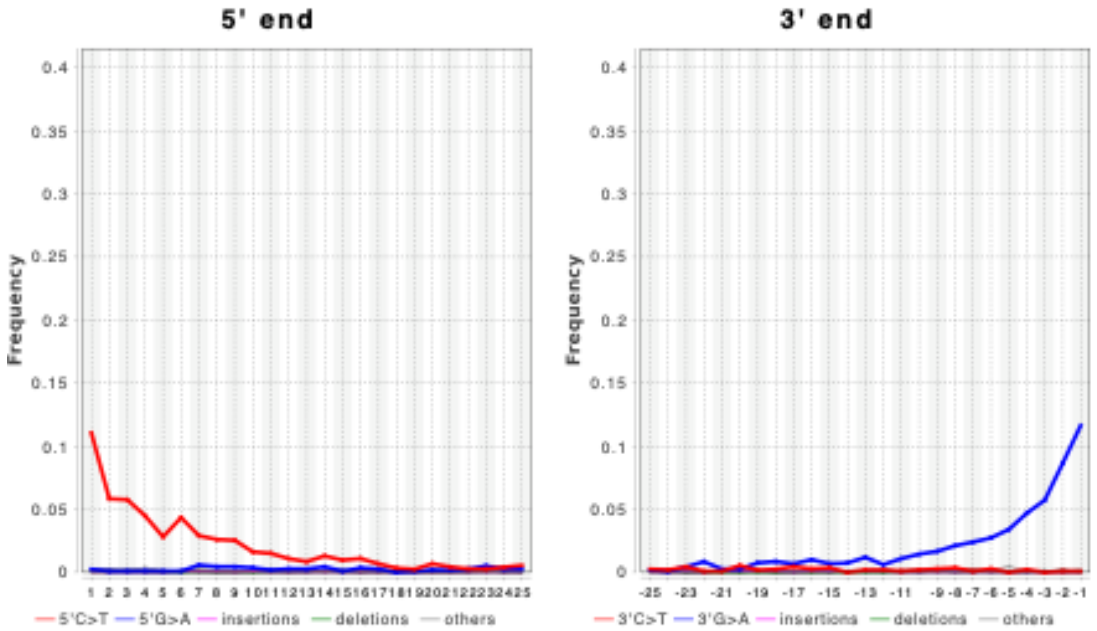
DNA antico

endogeno

DNA moderno

contaminante

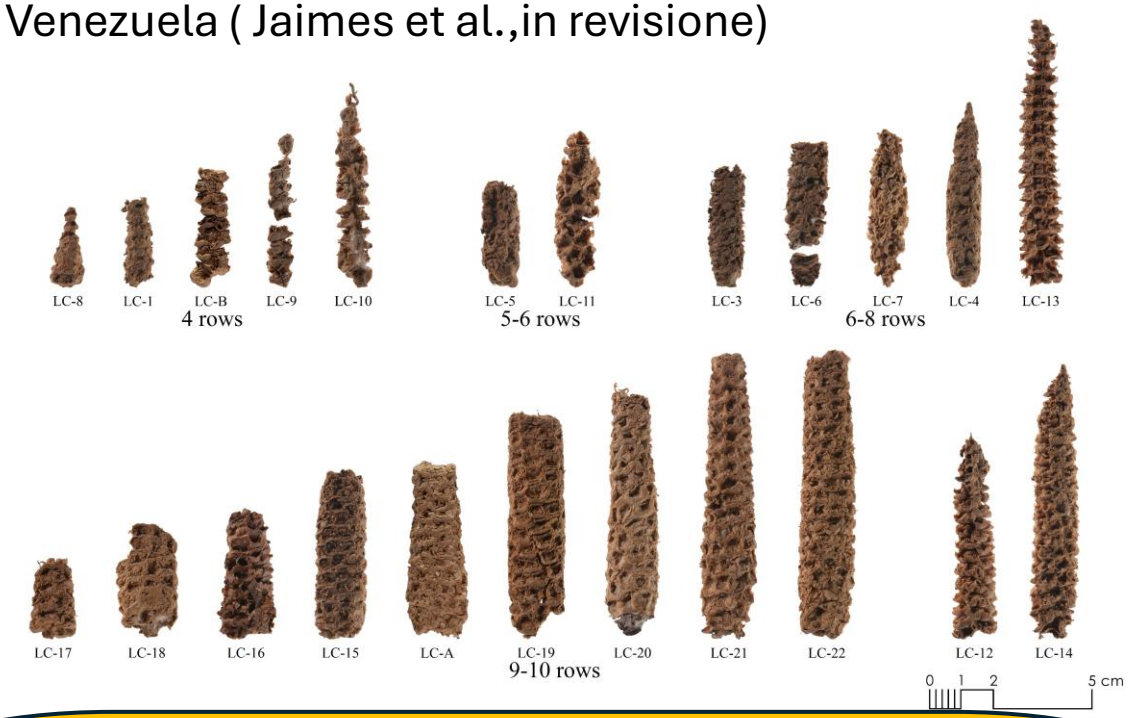




Effettivamente le sequenze non sembrano tortillas (almeno per questo campione)

Sequenze da dei rachidi probabilmente di mais di 2000 anni fa dal Venezuela (Jaimes et al.,in revisione)

```
~$ ./create_report_sequences.sh report.txt
~$ cat report.txt
================
sequences1.fastq
----------------
number of reads
250
number of high quality (F) bases
22881
number of adaptors
9
================
sequences2.fastq
----------------
number of reads
250
number of high quality (F) bases
23064
number of adaptors
3
================
```



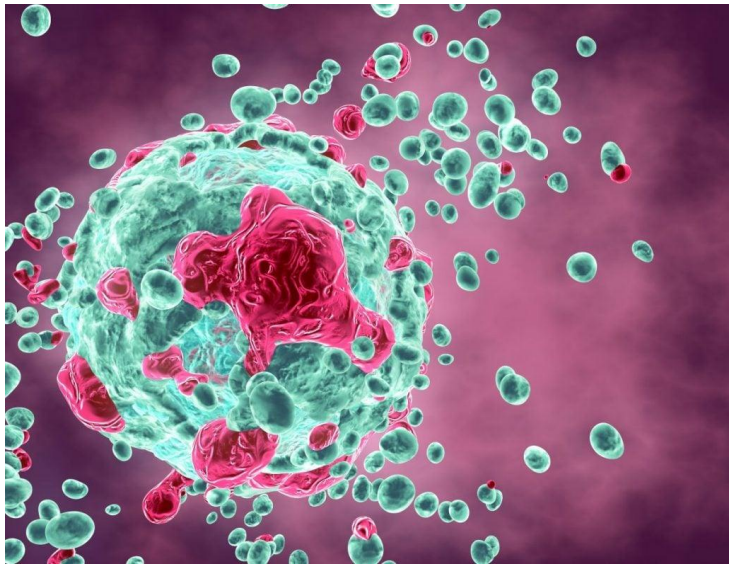
1) Vengo davvero da mais? Oppure da qualche altra pianta simile (competitive mapping)? E c'è anche DNA umano?
2) Sono antico, o vengo da una delle tortillas di Marcelo?

- Non erano le tortillas di Marcelo.
- E si, si possono mappare poche sequenze degradate (con le quali sarebbe impossibile assemblare un genoma) su una referenza moderna per applicazioni di **paleogenomica** e **genetica forense**.

Quasi ogni analisi di genomica necessita mappatura

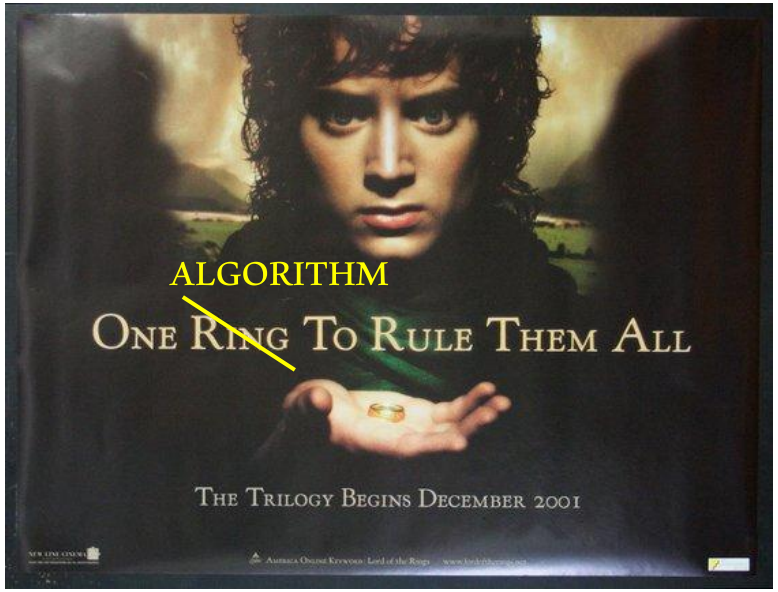
Espressione genica

Quali geni sono espressi in questa linea tumorale?



Assemblaggio di genomi

«Polishing» con short reads

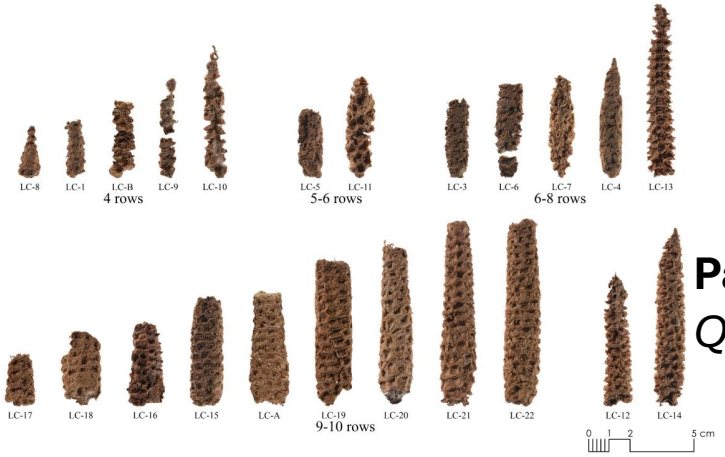


GWAS e genetica di popolazione

Quali varianti causano il diabete?

biobank uk

Improving the health of future generations

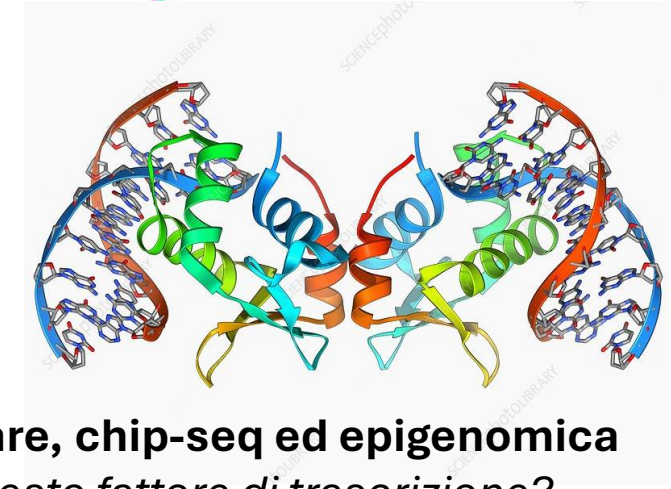


Paleogenomica e genetica forense

Questo osso è autentico ed antico?

Metagenomica

Quali specie sono presenti in questo campione?



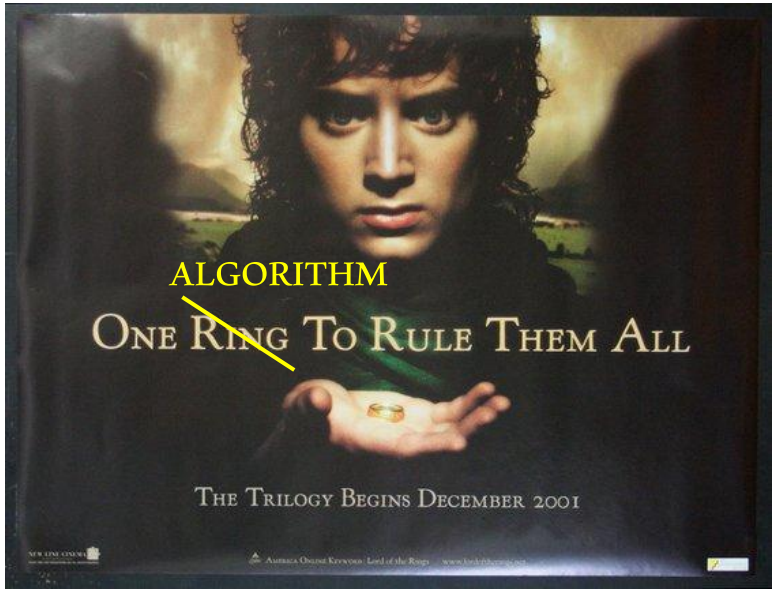
Biologia molecolare, chip-seq ed epigenomica

Che geni regola questo fattore di trascrizione?

Quali regioni hanno questa modifica della cromatina?

Quali sono le caratteristiche ideali del nostro algoritmo di mappatura?

- Veloce e capace di gestire milioni di sequenze con poche risorse computazionali
- Deve poter tollerare piccole differenze tra individui (varianti genetiche, errori di sequenziamento, mutazioni post-mortem)..
- ..MA allo stesso tempo deve essere accurato ed evitare di mappare una sequenza in zone aspecifiche!
- Deve poter funzionare su sequenze corte (Illumina)



Voi (un pochino) sapete già mappare. Come fareste voi con i comandi appresi durante il corso?

Voi (un pochino) sapete già mappare. Come fareste voi con i comandi appresi durante il corso?

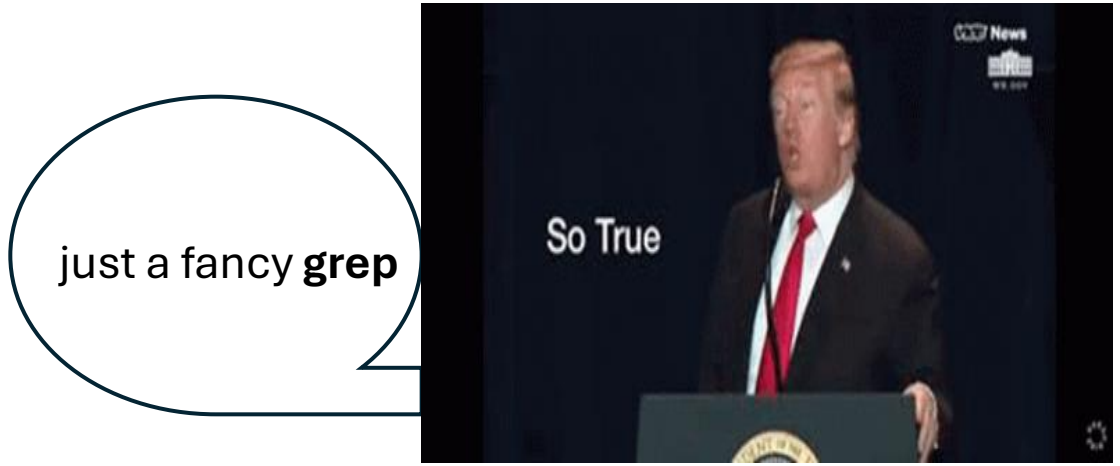
```
$ echo "ATGCGAGAGAGATCACGATTTTTTTTTTTTTAGACAGATAAATGACAG" | grep "TTTTAGACAG"
ATGCGAGAGAGATCACGATTTTTTTTTTTTAGACAGATAAATGACAG
```

just a fancy **grep**

Voi (un pochino) sapete già mappare. Come fareste voi con i comandi appresi durante il corso?

```
$ echo "ATGCGAGAGAGATCACGATTTTTTTTTTTTTAGACAGATAAATGACAG" | grep "TTTTAGACAG"
ATGCGAGAGAGATCACGATTTTTTTTTTTTTAGACAGATAAATGACAG
```

Quali sono i problemi di grep come mappatore?



Voi (un pochino) sapete già mappare. Come fareste voi con i comandi appresi durante il corso?

```
$ echo "ATGCGAGAGAGATCACGATTTTTTTTTTTTAGACAGATAAATGACAG" | grep "TTTTAGACAG"
ATGCGAGAGAGATCACGATTTTTTTTTTTTAGACAGATAAATGACAG
```

grep* potrebbe in principio gestire anche qualche “mutazione”..

```
$ echo "ATGCGAGAGAGATCACGATTTTTTTTTTTTAGACAGATAAATGACAG" | grep "TTT.AGACAG"
ATGCGAGAGAGATCACGATTTTTTTTTTTTAGACAGATAAATGACAG
```

*I caratteri speciali . e * permettono di considerare:

- . : un singolo carattere qualsiasi
- * : molteplici caratteri qualsiasi

Voi (un pochino) sapete già mappare. Come fareste voi con i comandi appresi durante il corso?

```
$ echo "ATGCGAGAGAGATCACGATTTTTTTTTTTTAGACAGATAAATGACAG" | grep "TTTTAGACAG"
ATGCGAGAGAGATCACGATTTTTTTTTTTTAGACAGATAAATGACAG
```

grep* potrebbe in principio gestire anche qualche “mutazione”..

```
$ echo "ATGCGAGAGAGATCACGATTTTTTTTTTTTAGACAGATAAATGACAG" | grep "TTT.AGACAG"
ATGCGAGAGAGATCACGATTTTTTTTTTTTAGACAGATAAATGACAG
```

*I caratteri speciali . e * permettono di considerare:

- . : un singolo carattere qualsiasi
- * : molteplici caratteri qualsiasi

Voi (un pochino) sapete già mappare. Come fareste voi con i comandi appresi durante il corso?

```
$ echo "ATGCGAGAGAGATCACGATTTTTTTTTTTTTAGACAGATAAATGACAG" | grep "TTTTAGACAG"
ATGCGAGAGAGATCACGATTTTTTTTTTTTAGACAGATAAATGACAG
```

grep* potrebbe in principio gestire anche qualche “mutazione”..

```
$ echo "ATGCGAGAGAGATCACGATTTTTTTTTTTTAGACAGATAAATGACAG" | grep "TTT.AGACAG"
ATGCGAGAGAGATCACGATTTTTTTTTTTTAGACAGATAAATGACAG
```

..e persino degli indels (delezioni e inserzioni)

```
$ echo "ATGCGAGAGAGATCACGATTTTTTTTTTTTAGACAGATAAATGACAG" | grep "TTT*AGACAG"
ATGCGAGAGAGATCACGATTTTTTTTTTTTAGACAGATAAATGACAG
```

*I caratteri speciali . e * permettono di considerare:

- . : un singolo carattere qualsiasi
- * : molteplici caratteri qualsiasi

Voi (un pochino) sapete già mappare. Come fareste voi con i comandi appresi durante il corso?

```
$ echo "ATGCGAGAGAGATCACGATTTTTTTTTTTTTAGACAGATAAATGACAG" | grep "TTTTAGACAG"
ATGCGAGAGAGATCACGATTTTTTTTTTTTAGACAGATAAATGACAG
```

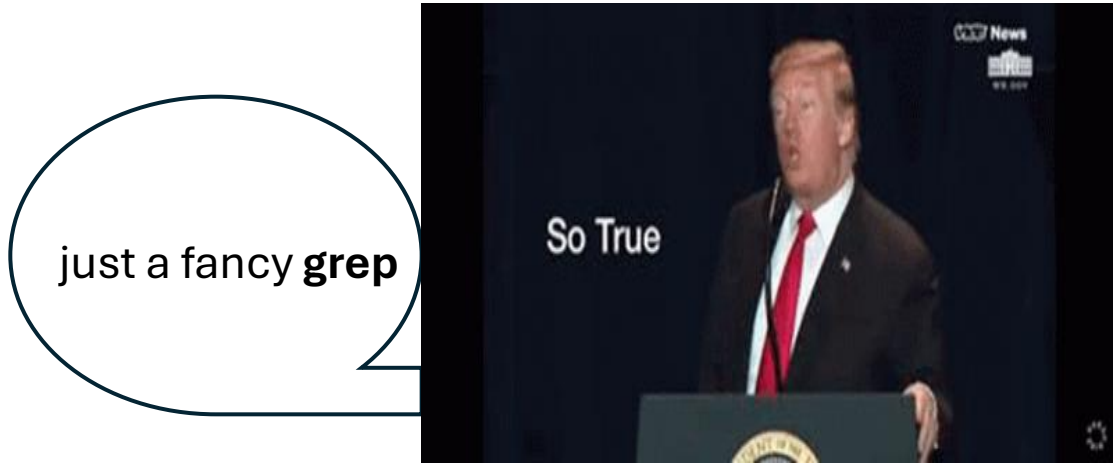
Se poi volete divertirvi e ammirare il potere infinito di grep, possiamo scrivere un piccolo **mapper giocattolo** con grep:

```
$ seq="ATGCGAGAGAGATCACGATTTTTTTTTTTTAGACAGATAAATGACAG"
read="TTTTAGACAG"
readlen=${#read}
for pos in $(seq 0 $((readlen - 1))); do
    pattern="${read:0:$pos}.${read:$((pos+1))}"
    for i in $(seq 0 $(( ${#seq} - readlen ))); do
        sub=${seq:$i:$readlen}
        if echo "$sub" | grep -qE "$pattern"; then echo "Match in posizione $i: $sub con sequenza $pattern";fi
done;done
Match in posizione 26: TTTTAGACAG con sequenza .TTTAGACAG
Match in posizione 26: TTTTAGACAG con sequenza T.TTAGACAG
Match in posizione 26: TTTTAGACAG con sequenza TT.TAGACAG
Match in posizione 26: TTTTAGACAG con sequenza TTT.AGACAG
Match in posizione 26: TTTTAGACAG con sequenza TTTT.GACAG
Match in posizione 26: TTTTAGACAG con sequenza TTTTA.ACAG
Match in posizione 26: TTTTAGACAG con sequenza TTTTAG.CAG
Match in posizione 26: TTTTAGACAG con sequenza TTTTAGA.AG
Match in posizione 26: TTTTAGACAG con sequenza TTTTAGAC.G
Match in posizione 26: TTTTAGACAG con sequenza TTTTAGACA.
```

Voi (un pochino) sapete già mappare. Come fareste voi con i comandi appresi durante il corso?

```
$ echo "ATGCGAGAGAGATCACGATTTTTTTTTTTTTAGACAGATAAATGACAG" | grep "TTTTAGACAG"
ATGCGAGAGAGATCACGATTTTTTTTTTTTTAGACAGATAAATGACAG
```

Quali sono i problemi di grep come mappatore?



Voi (un pochino) sapete già mappare. Come fareste voi con i comandi appresi durante il corso?

```
$ echo "ATGCGAGAGAGATCACGATTTTTTTTTTTTTAGACAGATAAATGACAG" | grep "TTTTAGACAG"
ATGCGAGAGAGATCACGATTTTTTTTTTTTAGACAGATAAATGACAG
```

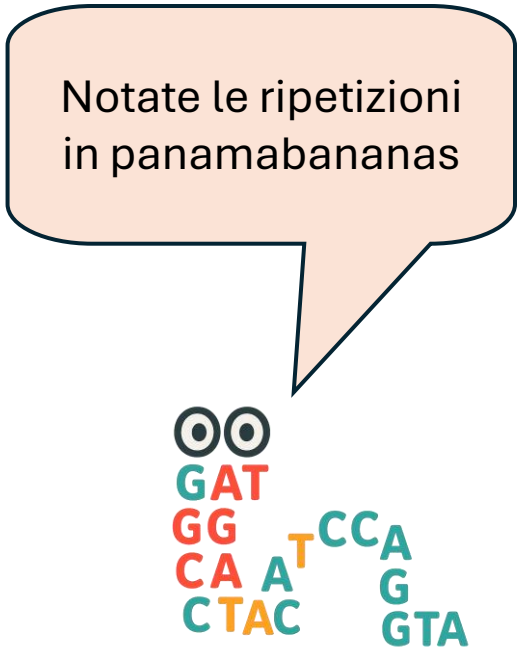
Quali sono i problemi di grep come mappatore?

grep deve ripercorrere tutta la sequenza per ogni mutazione e per ogni sequenza!!
Per dati reali sarebbe impraticabile!

Problema:

trovare in una stringa (il genoma) la posizone di sequenze molto piu' «corte»

Esempio di genoma: «*panamabananas*»



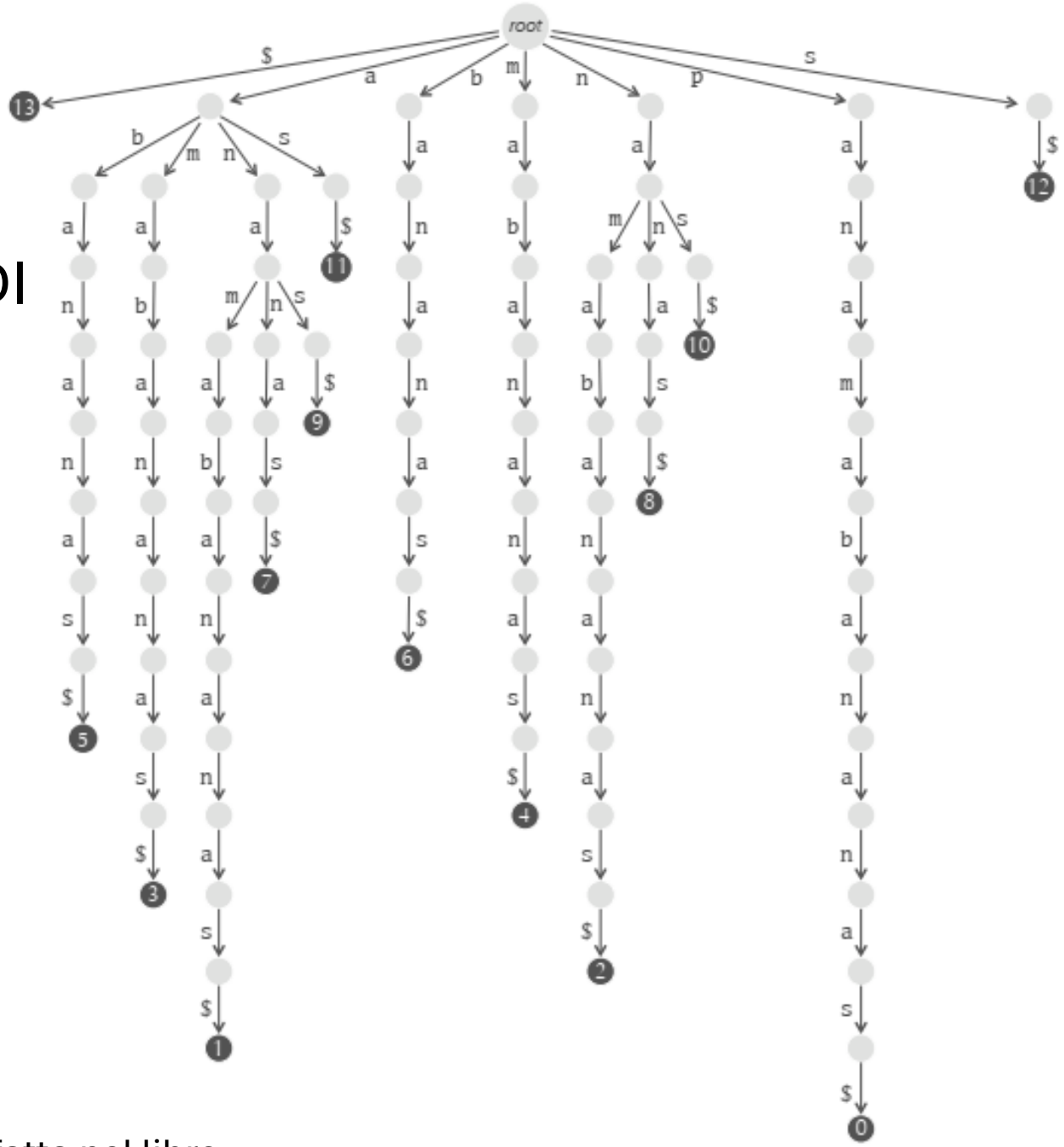
Come potremmo rappresentare il nostro genoma per evitare di doverlo ripercorrere tutto ogni volta?

Esempio di genoma: «*panamabananas*»

E se lo rappresentassimo in un albero?

ISTRUZIONI PER LA COSTRUZIONE DI UN SUFFIXTREE (albero dei suffissi):

1. Prima aggiungiamo un $ alla fine di panamabananas per marcarne la fine: ***panamabananas$***

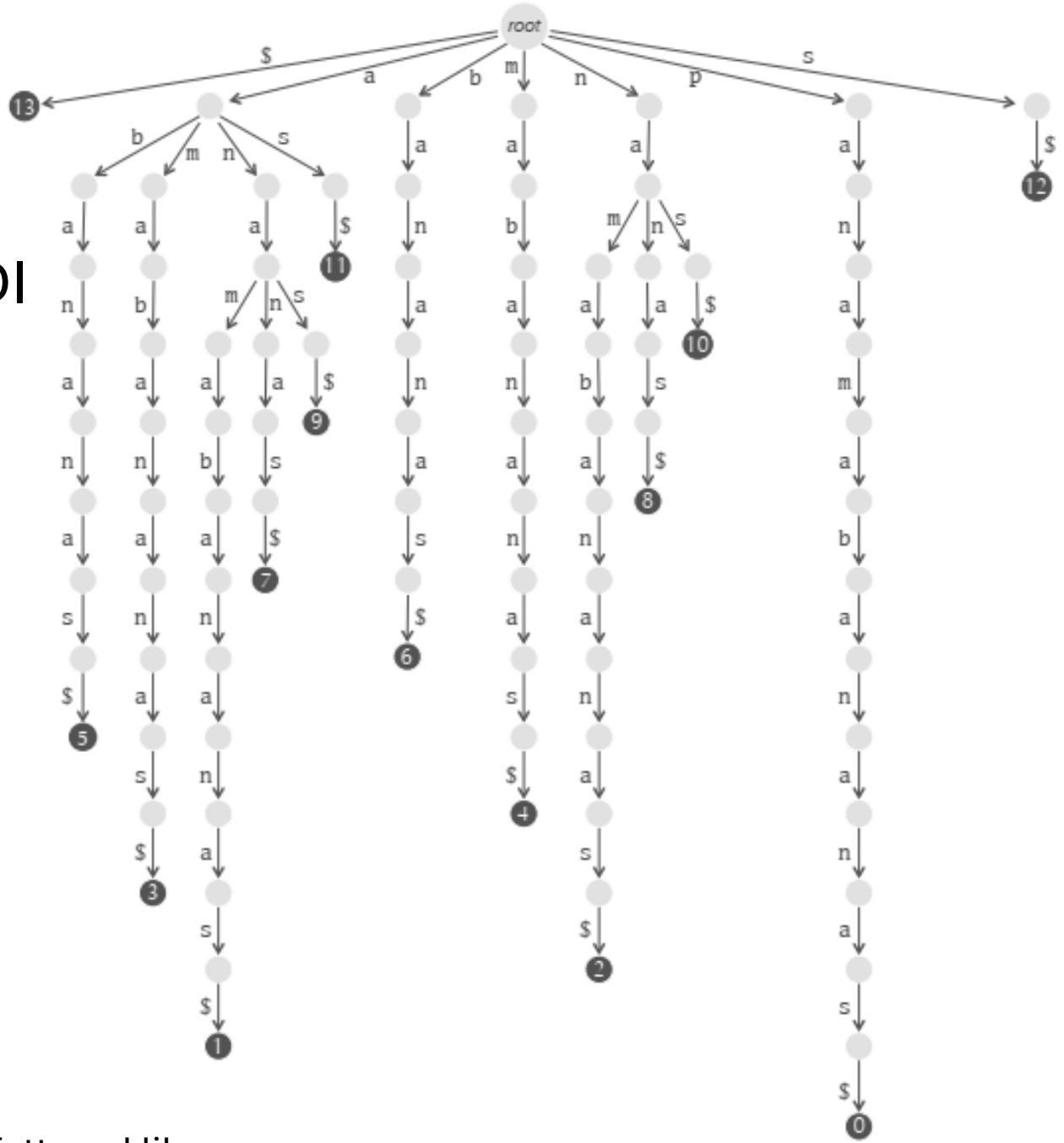


*nel corso non faremo distinzione tra ***trie*** e **tree**, invece fatta nel libro

E se lo rappresentassimo in un albero?

ISTRUZIONI PER LA COSTRUZIONE DI UN SUFFIXTREE (albero dei suffissi):

1. Prima aggiungiamo un $ alla fine di panamabananas per marcarne la fine: ***panamabananas$***
2. Prendiamo tutti i possibili ***suffissi*** di *panamabananas$* ($,s$,as$,nas$,anas$,etc.)

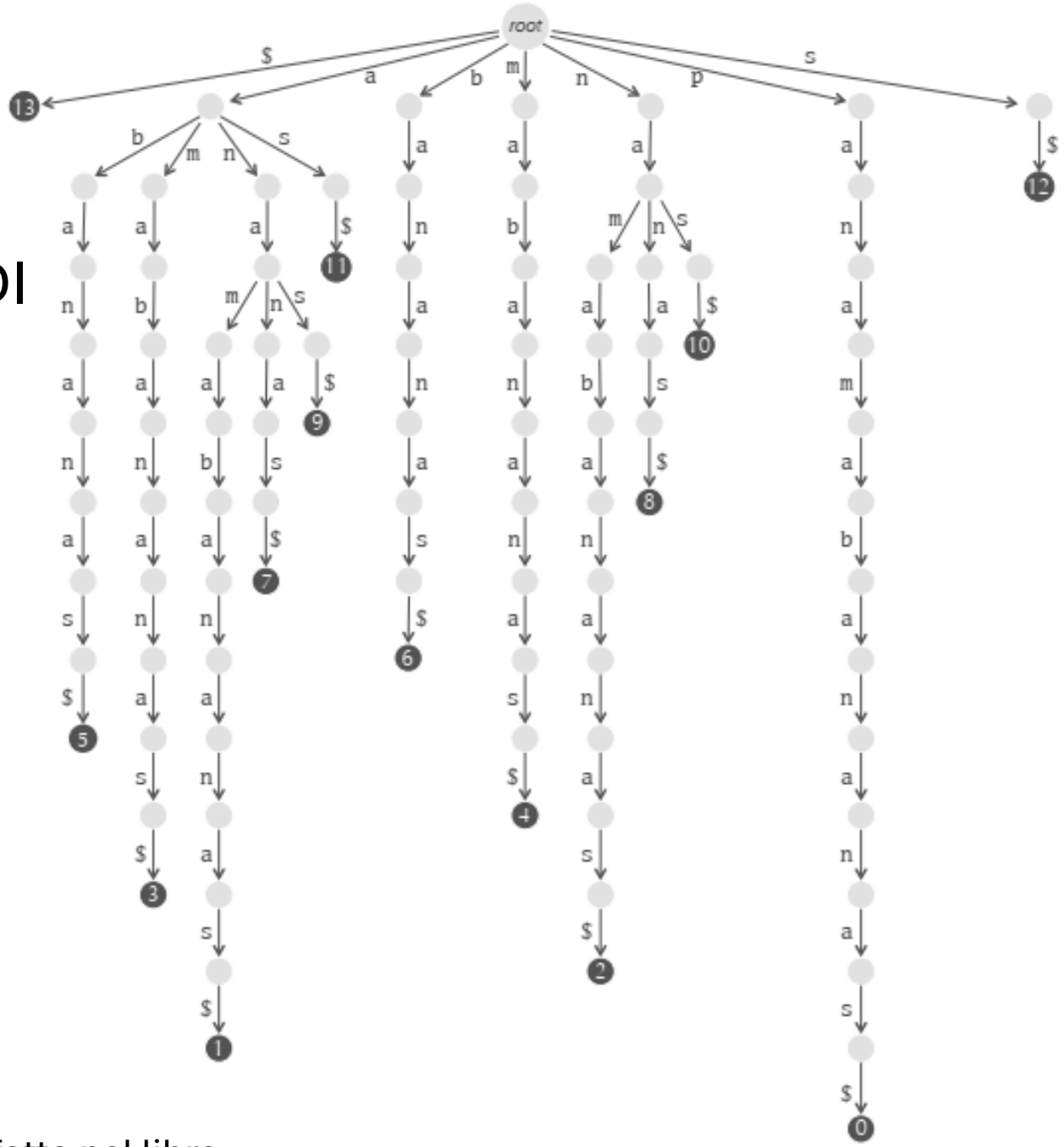


*nel corso non faremo distinzione tra ***trie*** e **tree**, invece fatta nel libro

E se lo rappresentassimo in un albero?

ISTRUZIONI PER LA COSTRUZIONE DI UN SUFFIXTREE (albero dei suffissi):

1. Prima aggiungiamo un $ alla fine di panamabananas per marcarne la fine: ***panamabananas$***
2. Prendiamo tutti i possibili ***suffissi*** di *panamabananas$* ($,s$,as$,nas$,anas$,etc.)

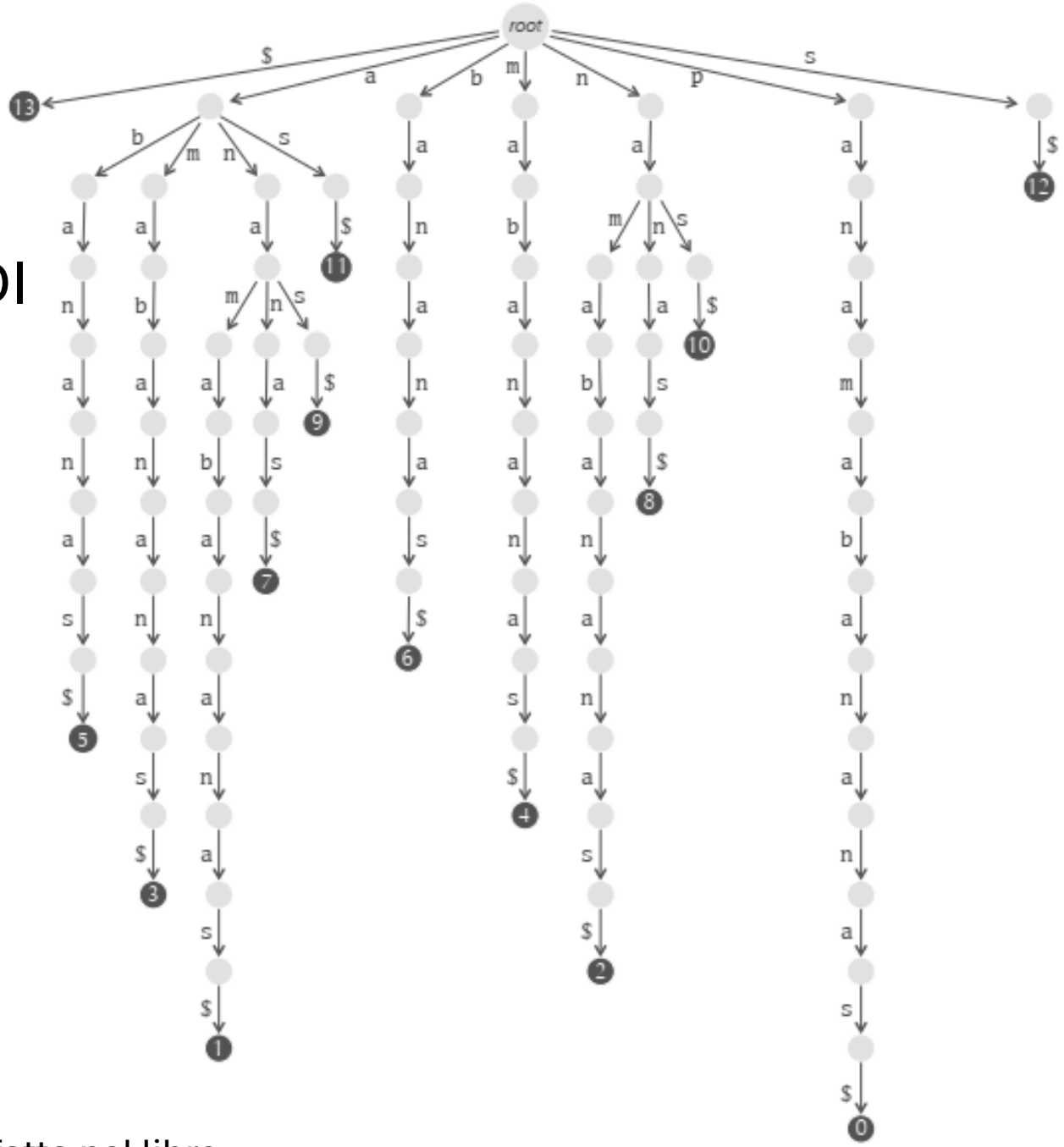


*nel corso non faremo distinzione tra ***trie*** e **tree**, invece fatta nel libro

E se lo rappresentassimo in un albero?

ISTRUZIONI PER LA COSTRUZIONE DI UN SUFFIXTREE (albero dei suffissi):

1. Prima aggiungiamo un $ alla fine di panamabananas per marcarne la fine: ***panamabananas$***
2. Prendiamo tutti i possibili ***suffissi*** di *panamabananas$* ($,s$,as$,nas$,anas$,etc.)

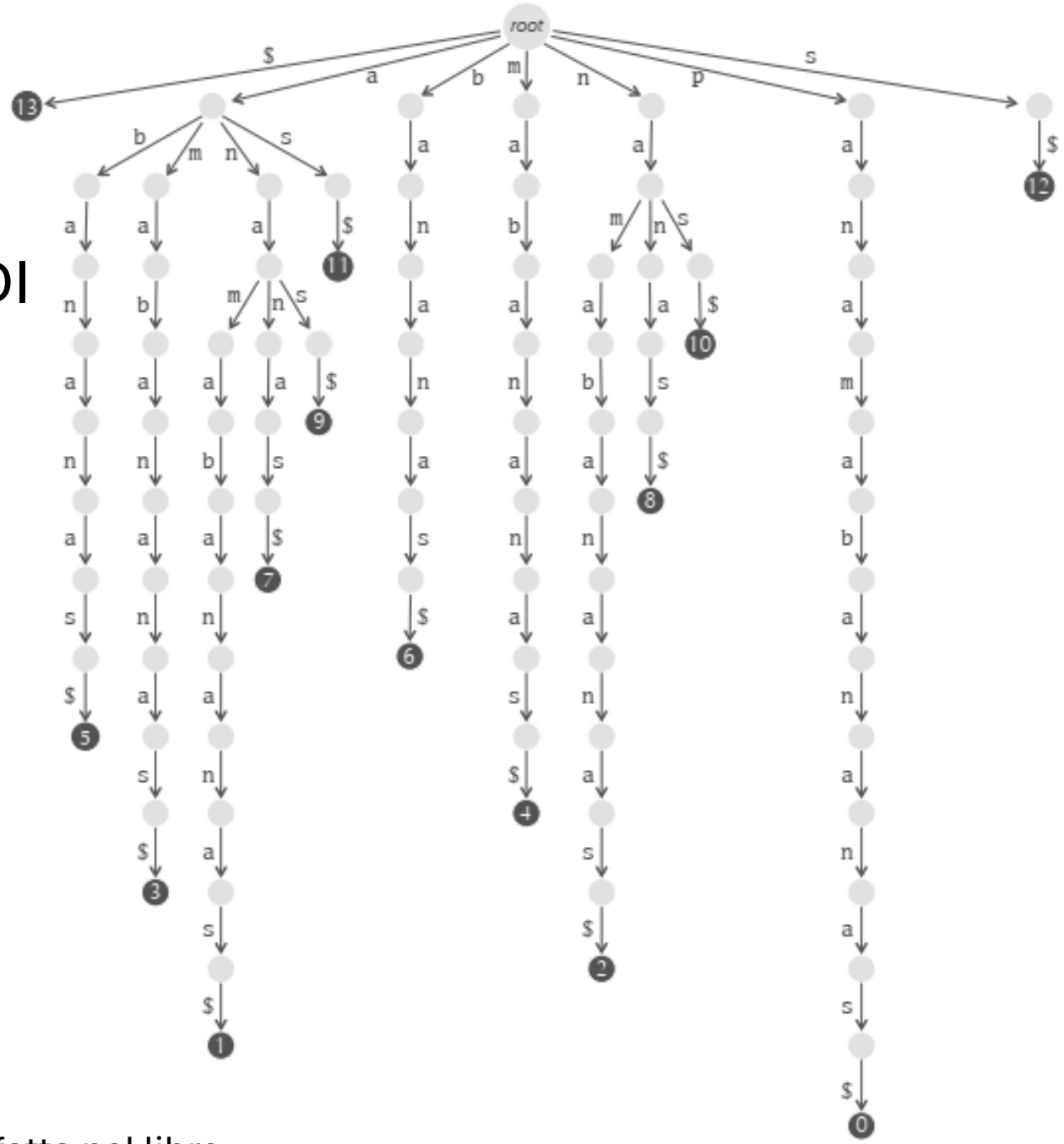


*nel corso non faremo distinzione tra ***trie*** e **tree**, invece fatta nel libro

E se lo rappresentassimo in un albero?

ISTRUZIONI PER LA COSTRUZIONE DI UN SUFFIXTREE (albero dei suffissi):

1. Prima aggiungiamo un $ alla fine di panamabananas per marcarne la fine: ***panamabananas$***
2. Prendiamo tutti i possibili ***suffissi*** di *panamabananas$* ($,s$,as$,nas$,anas$,etc.)

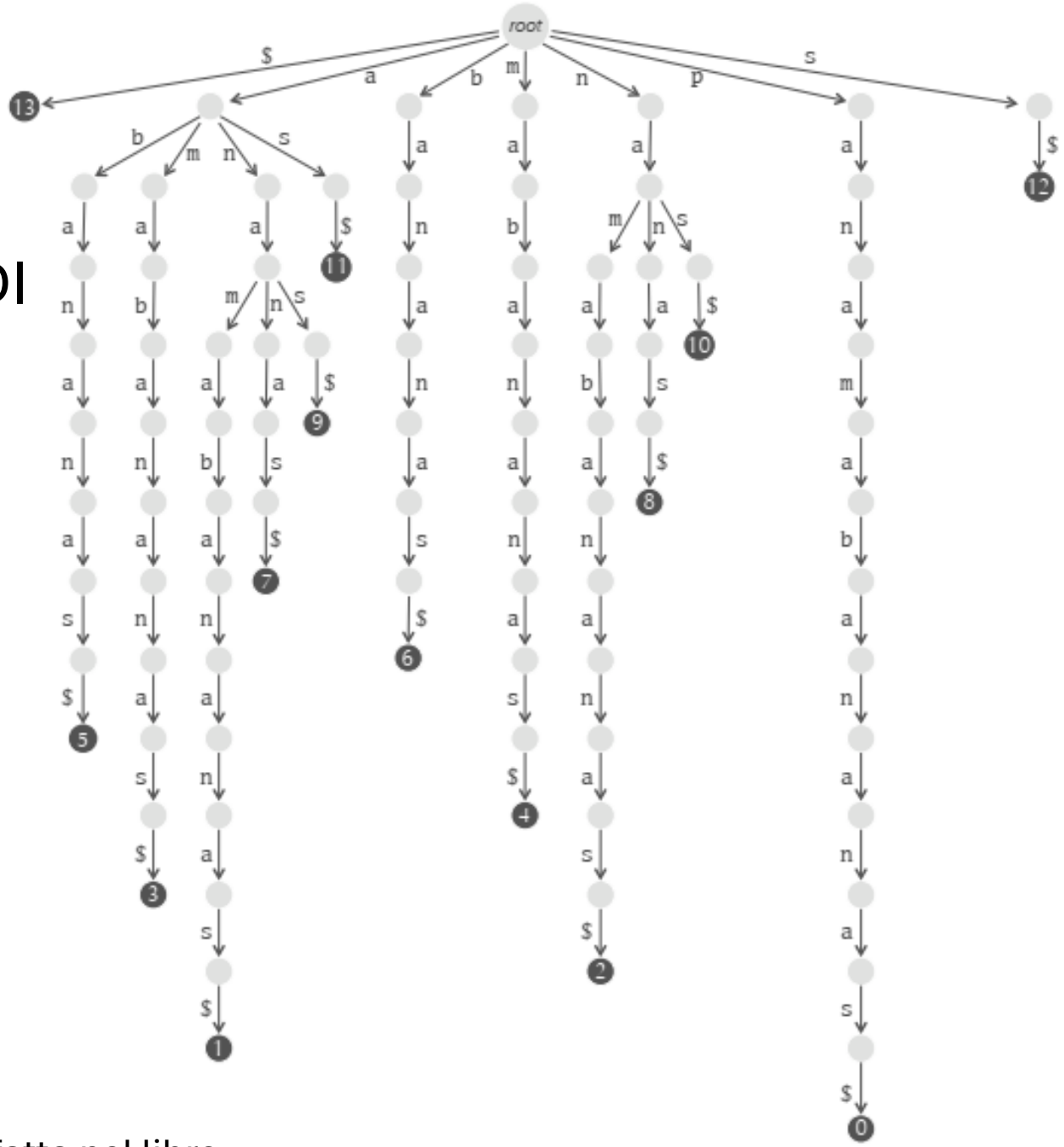


*nel corso non faremo distinzione tra ***trie*** e **tree**, invece fatta nel libro

E se lo rappresentassimo in un albero?

ISTRUZIONI PER LA COSTRUZIONE DI UN SUFFIXTREE (albero dei suffissi):

1. Prima aggiungiamo un $ alla fine di panamabananas per marcarne la fine: ***panamabananas$***
2. Prendiamo tutti i possibili ***suffissi*** di *panamabananas$* ($,s$,as$,nas$,anas$,etc.)

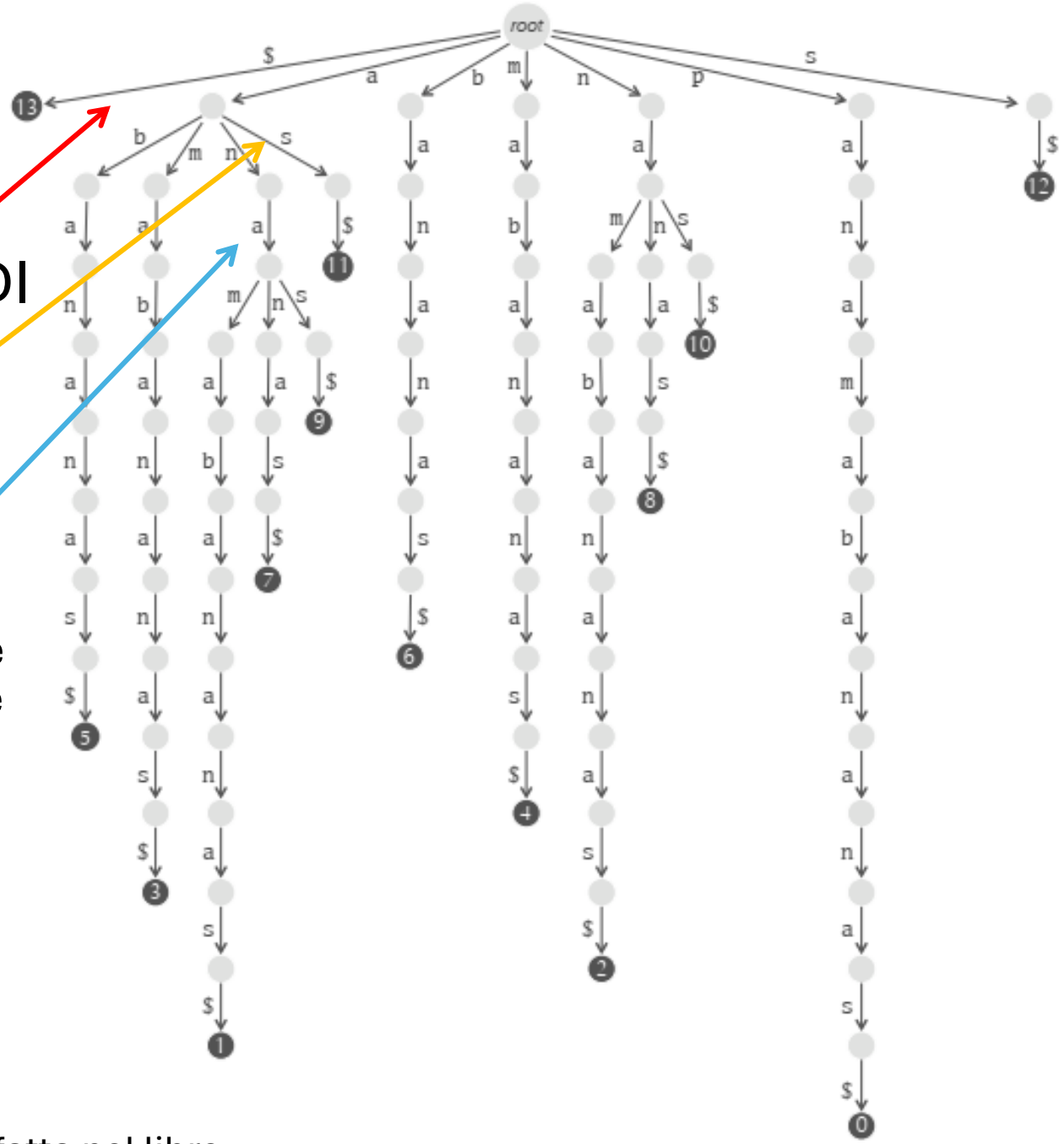


*nel corso non faremo distinzione tra ***trie*** e **tree**, invece fatta nel libro

E se lo rappresentassimo in un albero?

ISTRUZIONI PER LA COSTRUZIONE DI UN SUFFIXTREE (albero dei suffissi):

1. Prima aggiungiamo un $ alla fine di panamabananas per marcarne la fine: ***panamabananas$***
2. Prendiamo tutti i possibili ***suffissi*** di *panamabananas$* ($,s$,as$,nas$,anas$,etc.) e organizziamoli in un albero, facendo attenzione a:
   - raggruppare quelli che cominciano allo stesso modo in ramificazioni

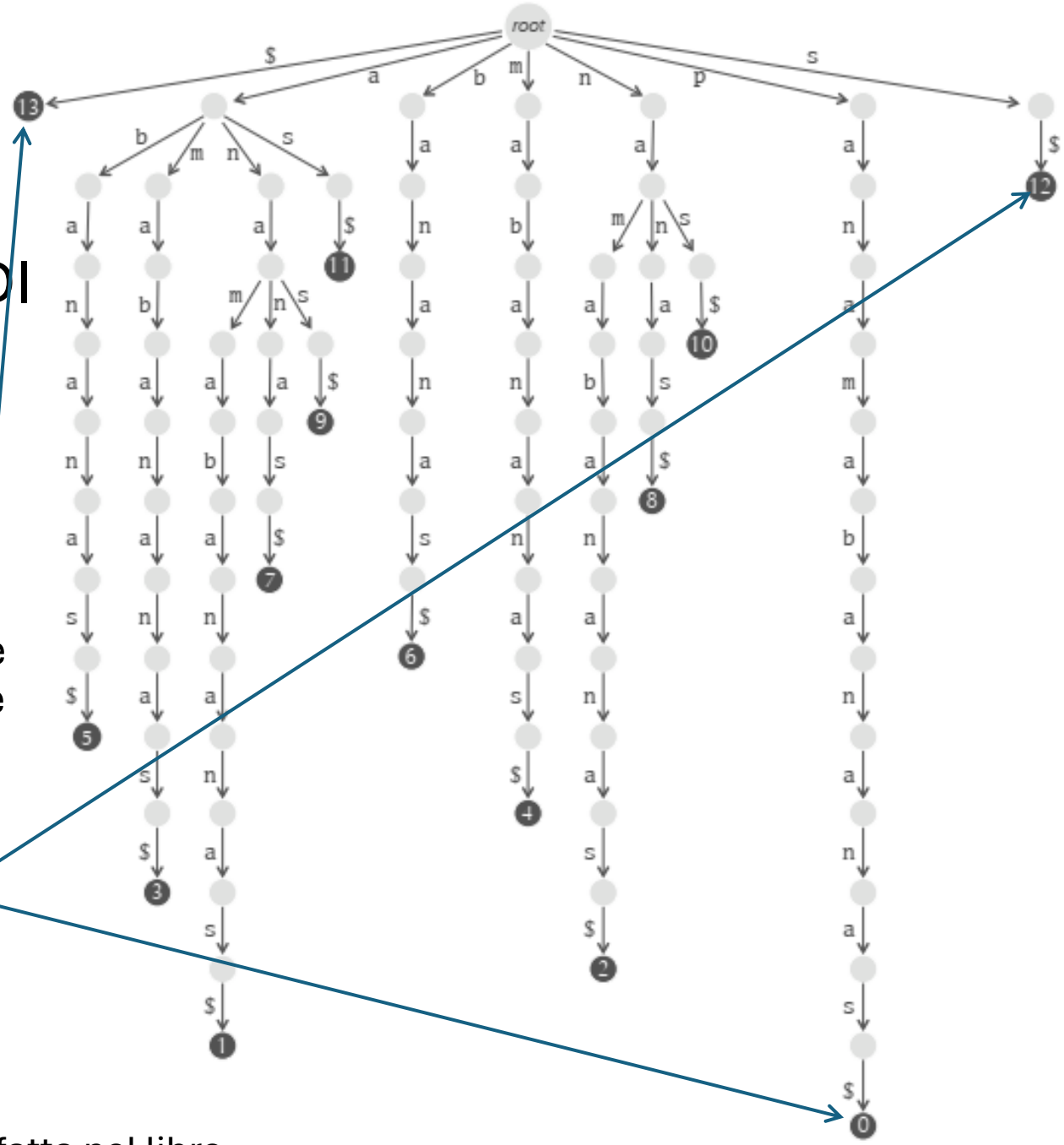


*nel corso non faremo distinzione tra ***trie*** e **tree**, invece fatta nel libro

E se lo rappresentassimo in un albero?

ISTRUZIONI PER LA COSTRUZIONE DI UN SUFFIXTREE (albero dei suffissi):

1. Prima aggiungiamo un $ alla fine di panamabananas per marcarne la fine: *panamabananas$*
2. Prendiamo tutti i possibili ***suffissi*** di *panamabananas$* ($,s$,as$,nas$,anas$,etc.) e organizziamoli in un albero, facendo attenzione a:
   - raggruppare quelli che cominciano allo stesso modo in ramificazioni
   - segnare la **posizione** del suffisso alla sua fine (pallino nero)

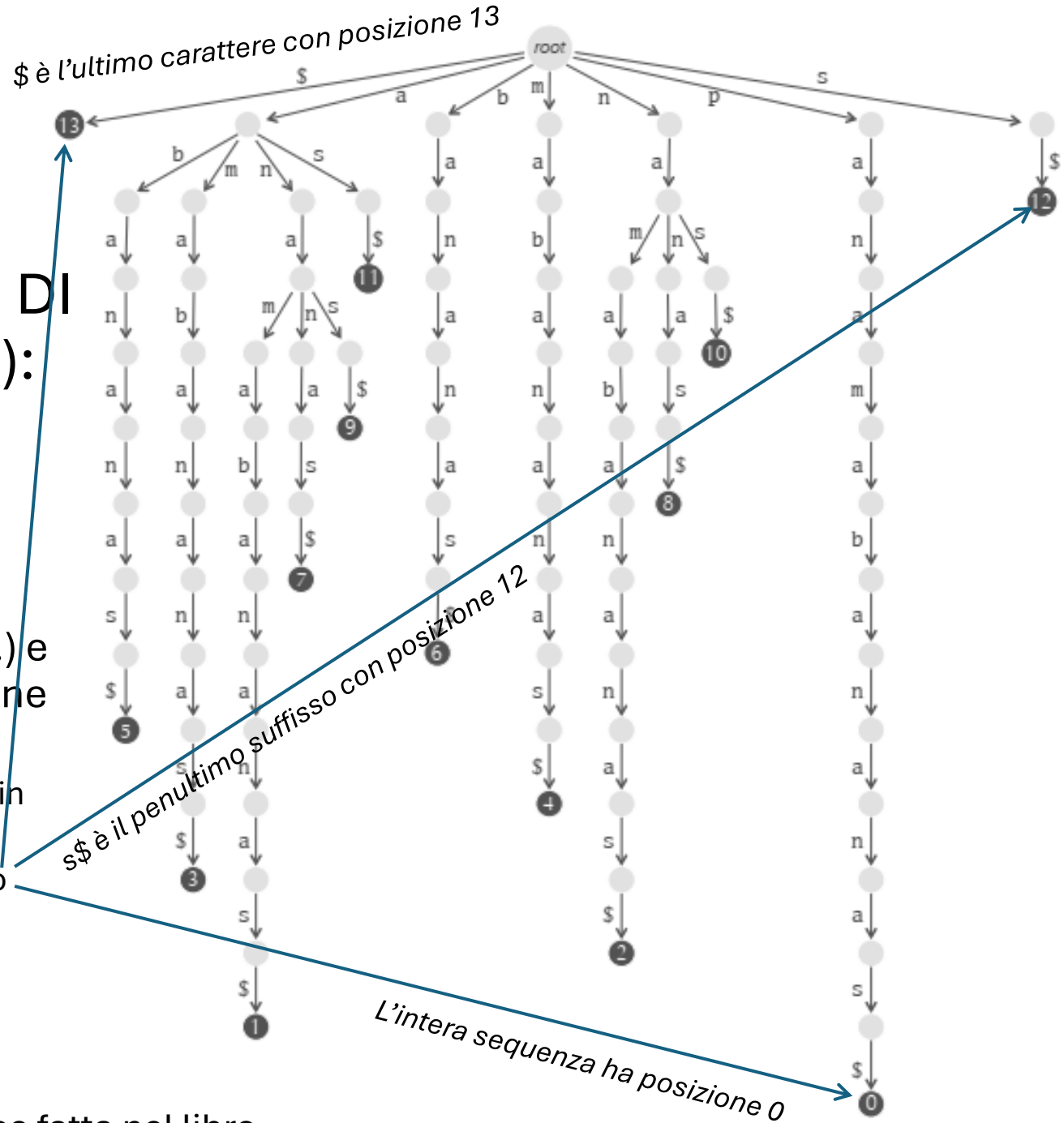


*nel corso non faremo distinzione tra ***trie*** e **tree**, invece fatta nel libro

E se lo rappresentassimo in un albero?

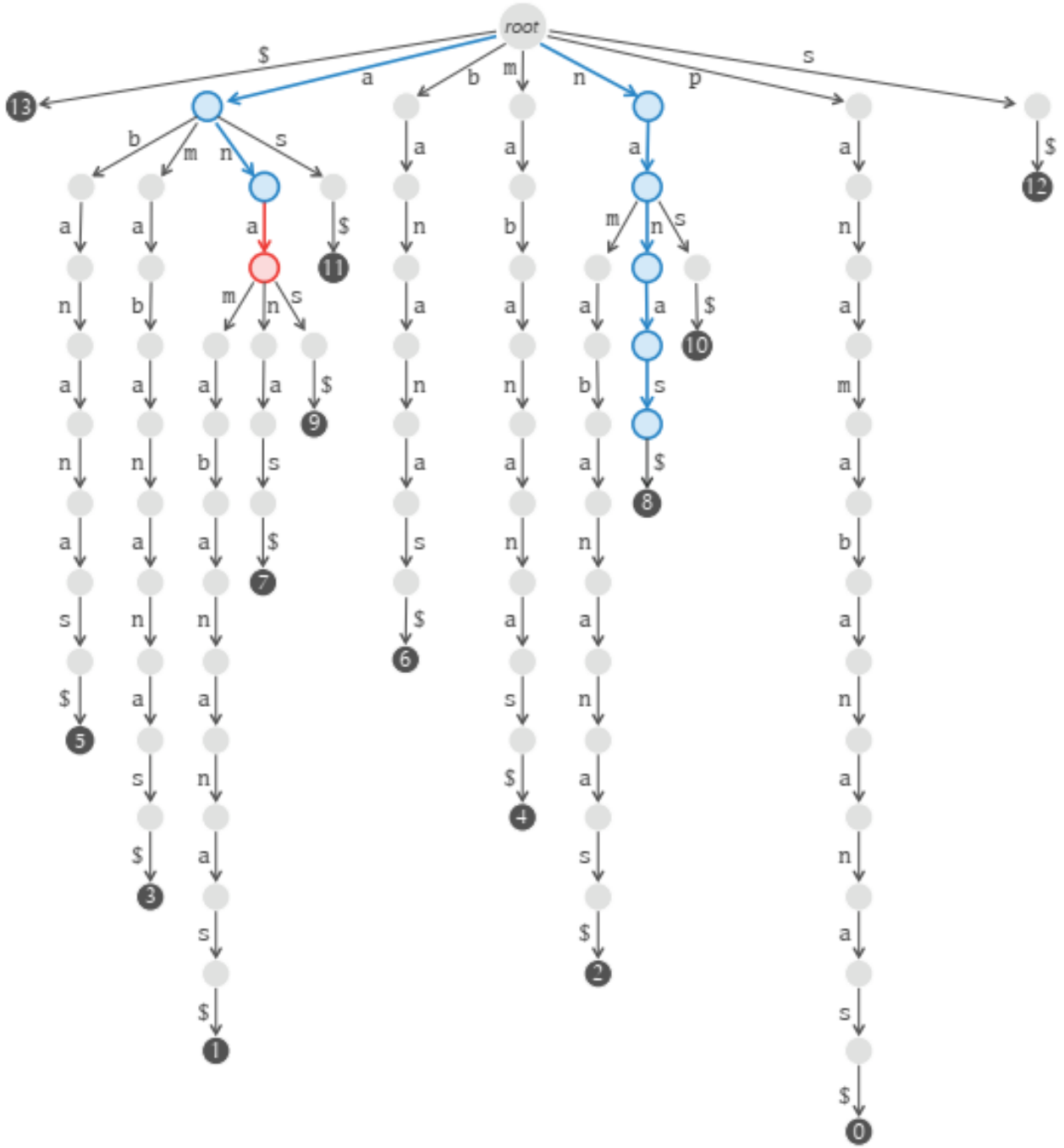
ISTRUZIONI PER LA COSTRUZIONE DI UN SUFFIXTREE (albero dei suffissi):

1. Prima aggiungiamo un $ alla fine di panamabananas per marcarne la fine: *panamabananas$*
2. Prendiamo tutti i possibili ***suffissi*** di *panamabananas$* ($,s$,as$,nas$,anas$,etc.) e organizziamoli in un albero, facendo attenzione a:
   - raggruppare quelli che cominciano allo stesso modo in ramificazioni
   - segnare la **posizione** del suffisso alla sua fine (pallino nero)



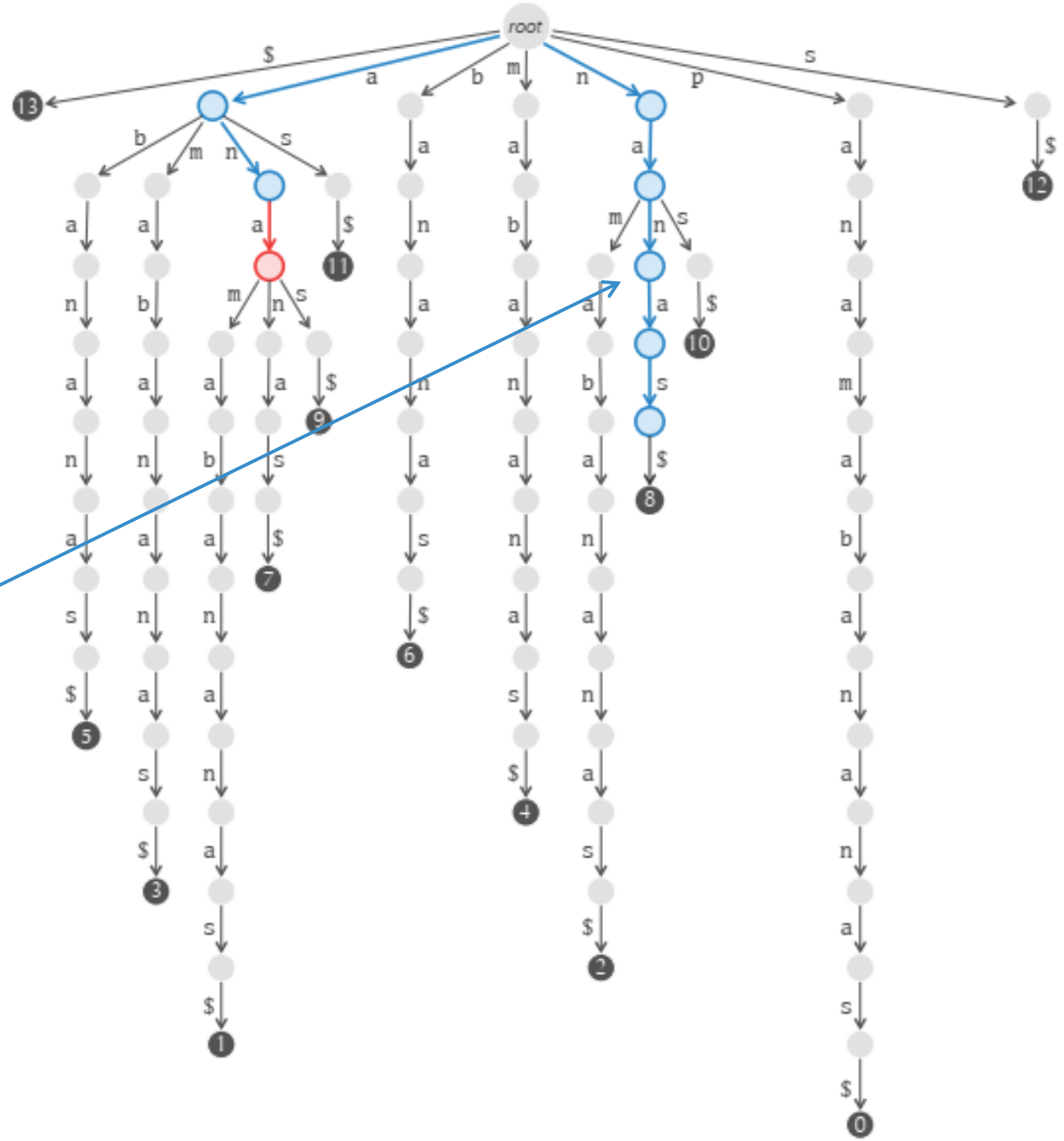
*nel corso non faremo distinzione tra ***trie*** e **tree**, invece fatta nel libro

Come cercare ora le nostre sequenze?
Basta sequire i rami dell'albero*!



* leggere la sequenza a partire dal suo prefisso e vedere se corrisponde al suffisso sul ramo controllando carattere per carattere

Come cercare ora le nostre sequenze? Basta sequire i rami dell'albero*!

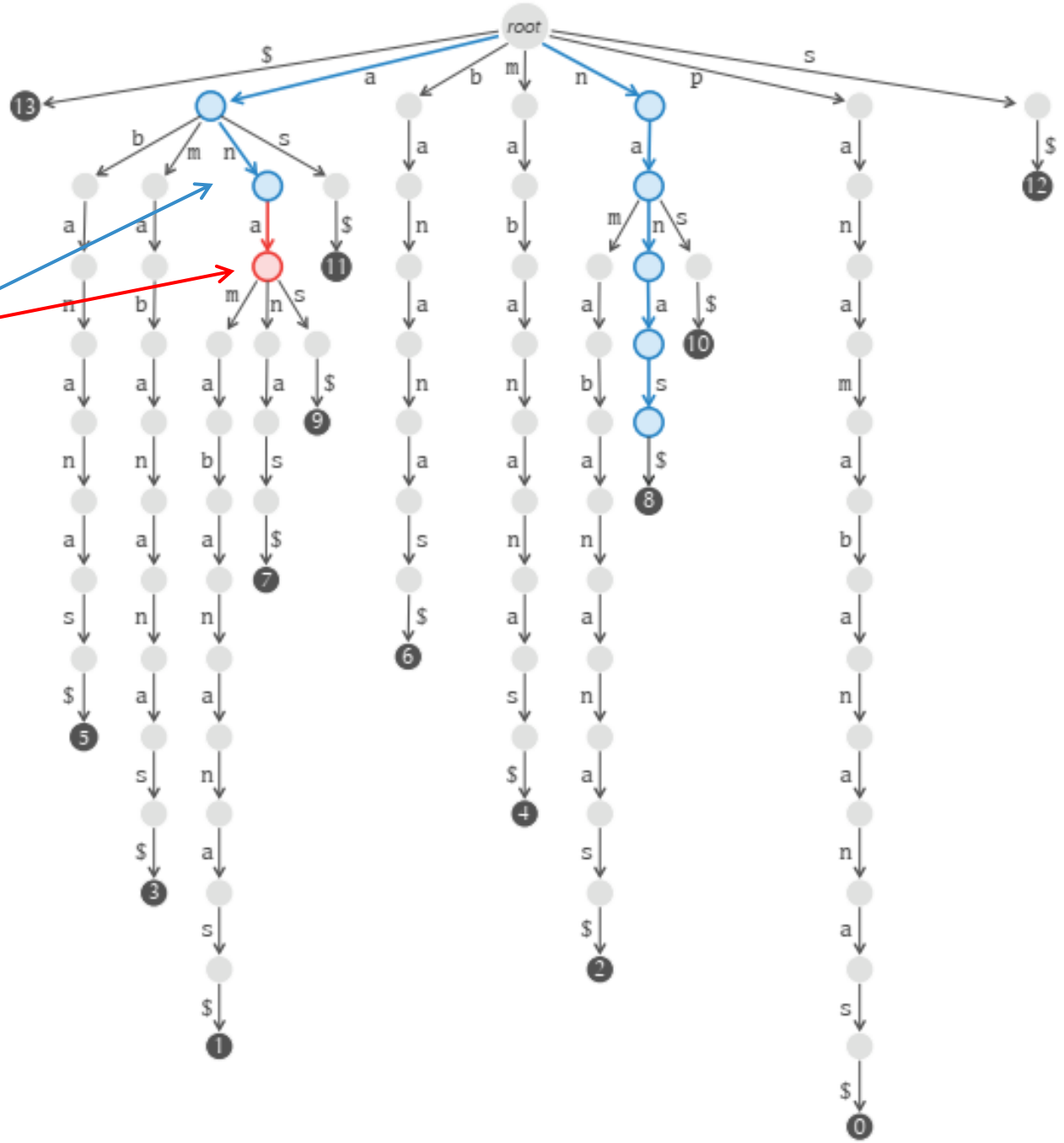


Esempio: La parola ***nanas$*** è contenuta in panamabananas$ con posizione 8!

* leggere la sequenza a partire dal suo prefisso e vedere se corrisponde al suffisso sul ramo controllando carattere per carattere

Come cercare ora le nostre sequenze? Basta sequire i rami dell'albero*!

Esempio: La parola ***antenna*** non è contenuta in panamabananas, in quanto ***an*** corrisponde (**in blu**) ma la ***t*** non è presente dopo ***an*** in panamabananas

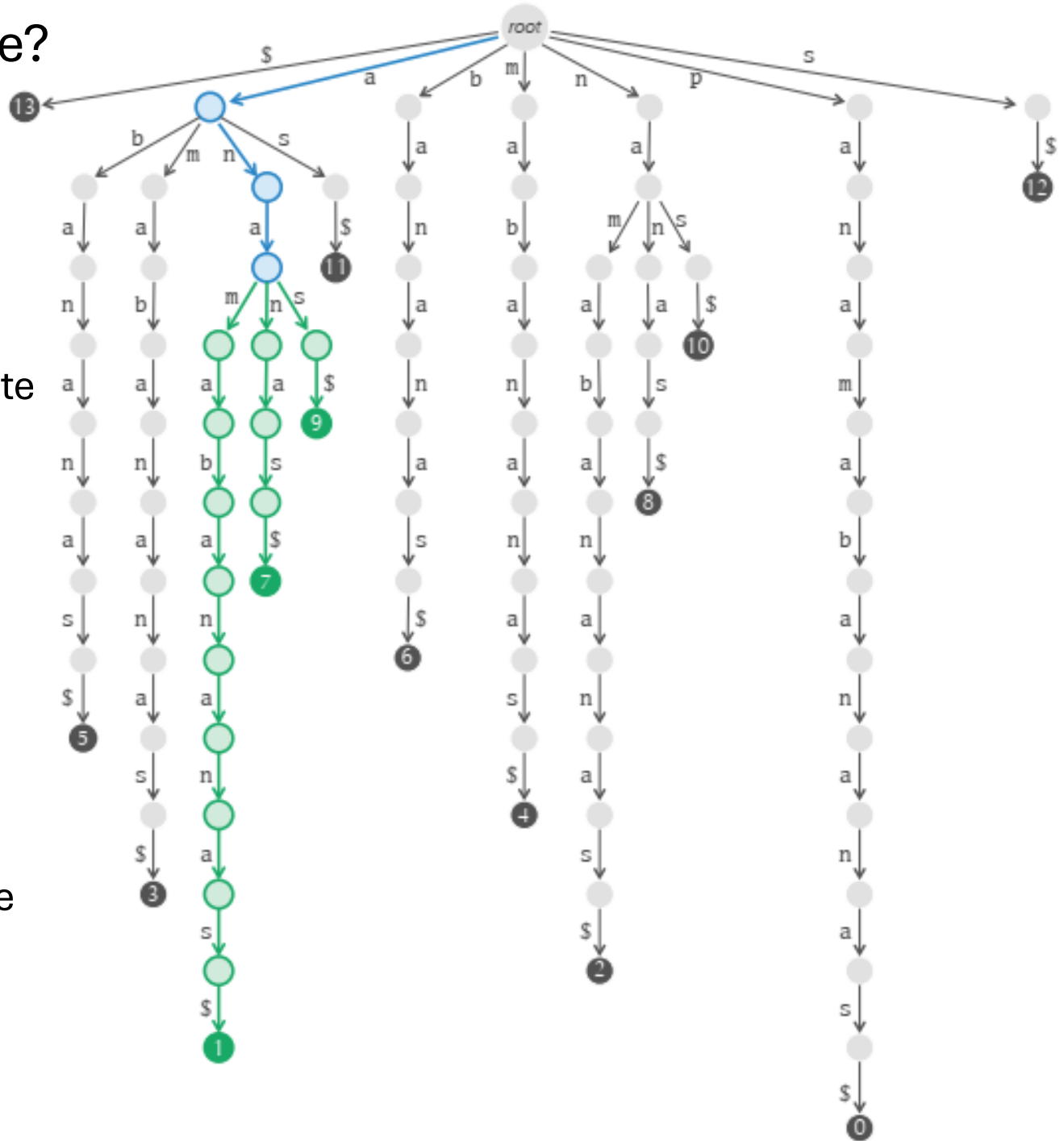


* leggere la sequenza a partire dal suo prefisso e vedere se corrisponde al suffisso sul ramo controllando carattere per carattere

Come cercare ora le nostre sequenze? Basta sequire i rami dell'albero*!

Esempio: Tutte le occorrenze della parola ***ana*** (posizioni **1,7 e 9**) sono mostrate e rappresentate in **varie ramificazioni**

* leggere la sequenza a partire dal suo prefisso e vedere se corrisponde al suffisso sul ramo controllando carattere per carattere



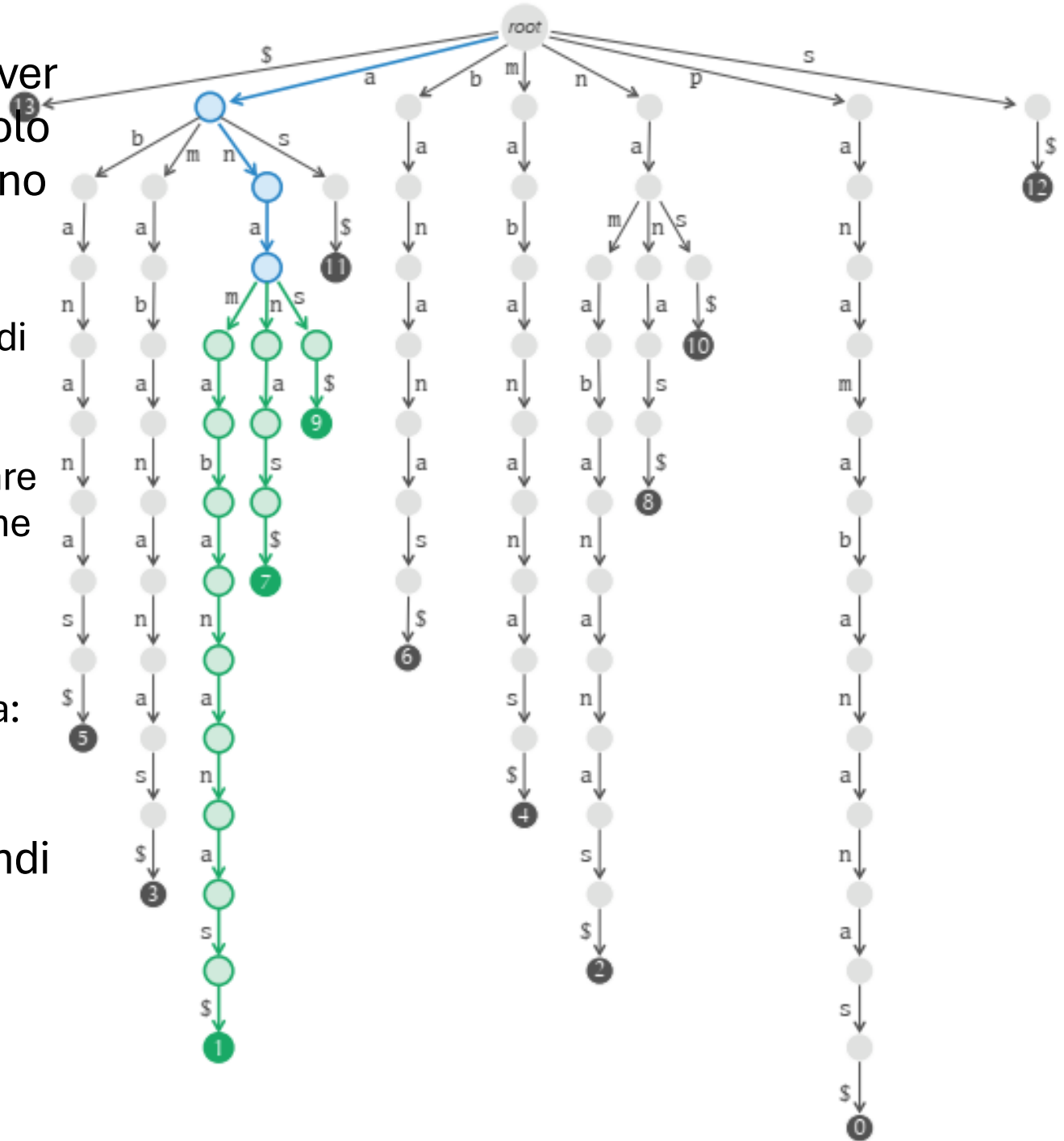
Gli alberi di suffissi permettono di non dover scorrere l'intero genoma ogni volta, ma solo percorrere quei percorsi che corrispondono alla nostra sequenza.

Per una sequenza di lunghezza ***m*** e un genoma di lunghezza ***n***:

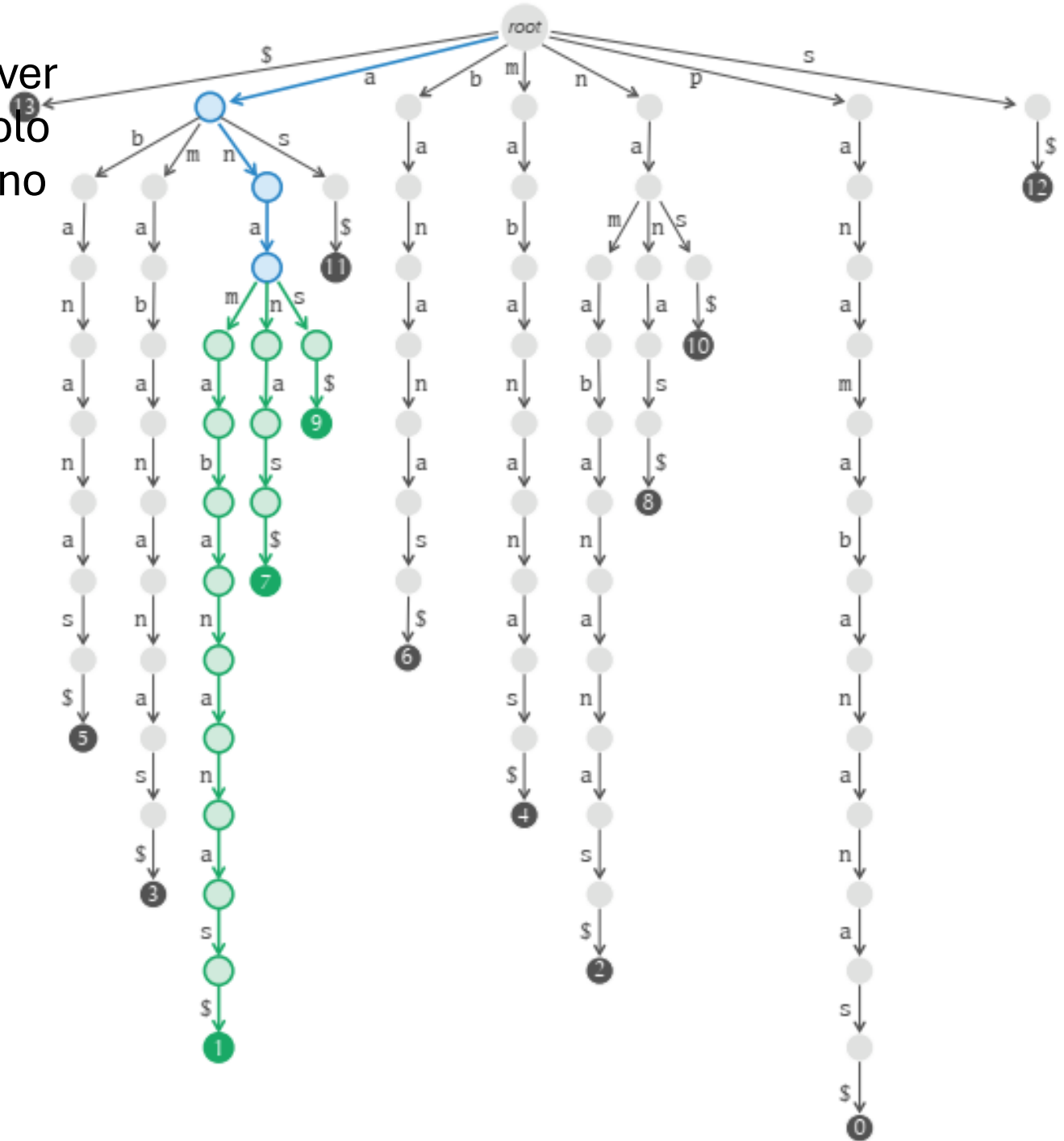
- grep dovrebbe (nel peggiore dei casi) comparare ogni posizione della sequenza con ogni posizione del genoma: complessità O(nxm)

- un albero di suffissi va direttamente alla ramificazione per ogni posizione della sequenza: complessità O(m).

Tuttavia deve ricordarsi tutto l'albero, quindi richiede un sacco di memoria!!



Gli alberi di suffissi permettono di non dover scorrere l’intero genoma ogni volta, ma solo percorrere quei percorsi che corrispondono alla nostra sequenza.



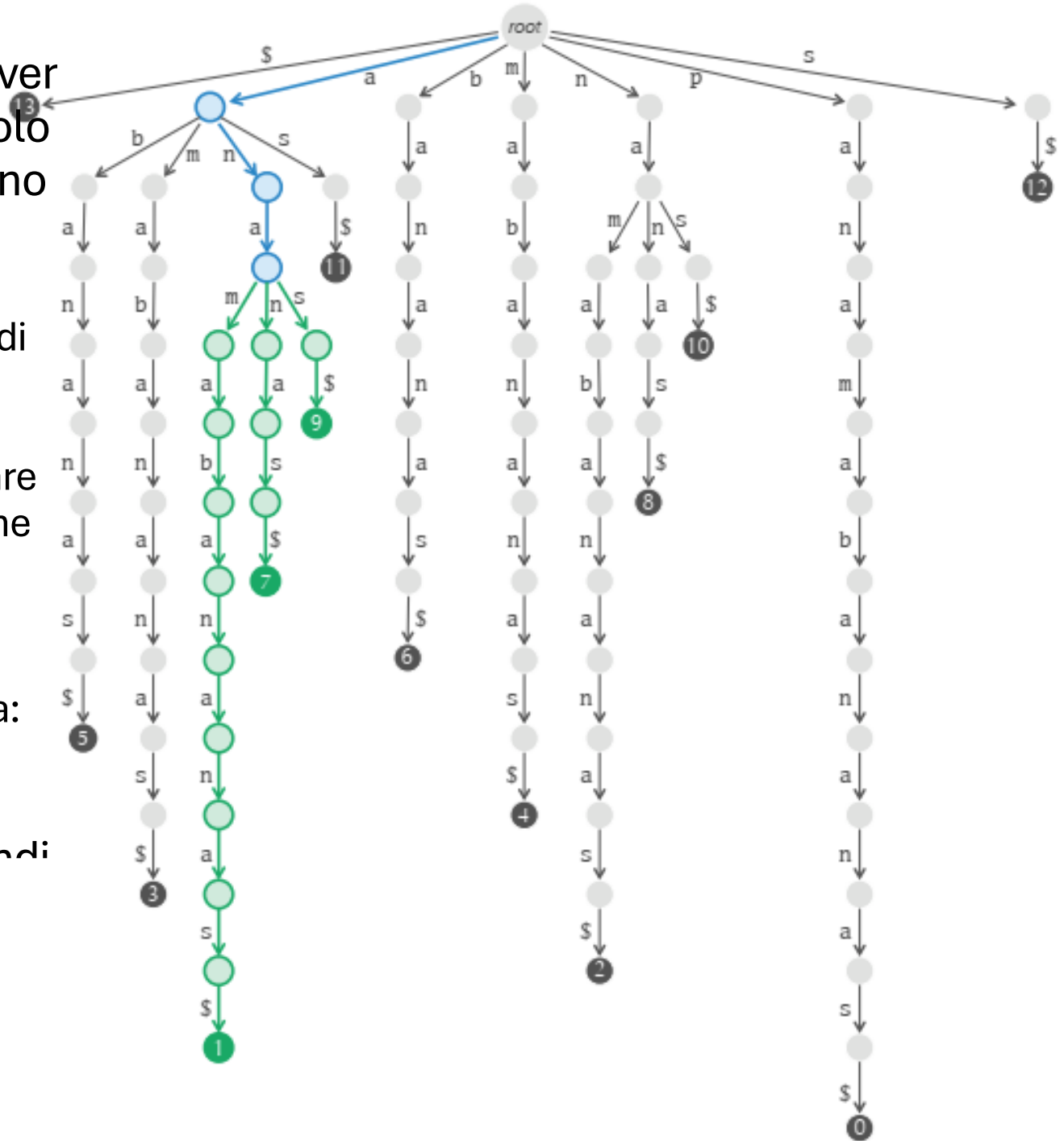
Gli alberi di suffissi permettono di non dover scorrere l'intero genoma ogni volta, ma solo percorrere quei percorsi che corrispondono alla nostra sequenza.

Per una sequenza di lunghezza ***m*** e un genoma di lunghezza ***n***:

- grep dovrebbe (nel peggiore dei casi) comparare ogni posizione della sequenza con ogni posizione del genoma: complessità O(nxm)

- un albero di suffissi va direttamente alla ramificazione per ogni posizione della sequenza: complessità O(m).

Tuttavia deve ricordarsi tutto l'albero, quindi



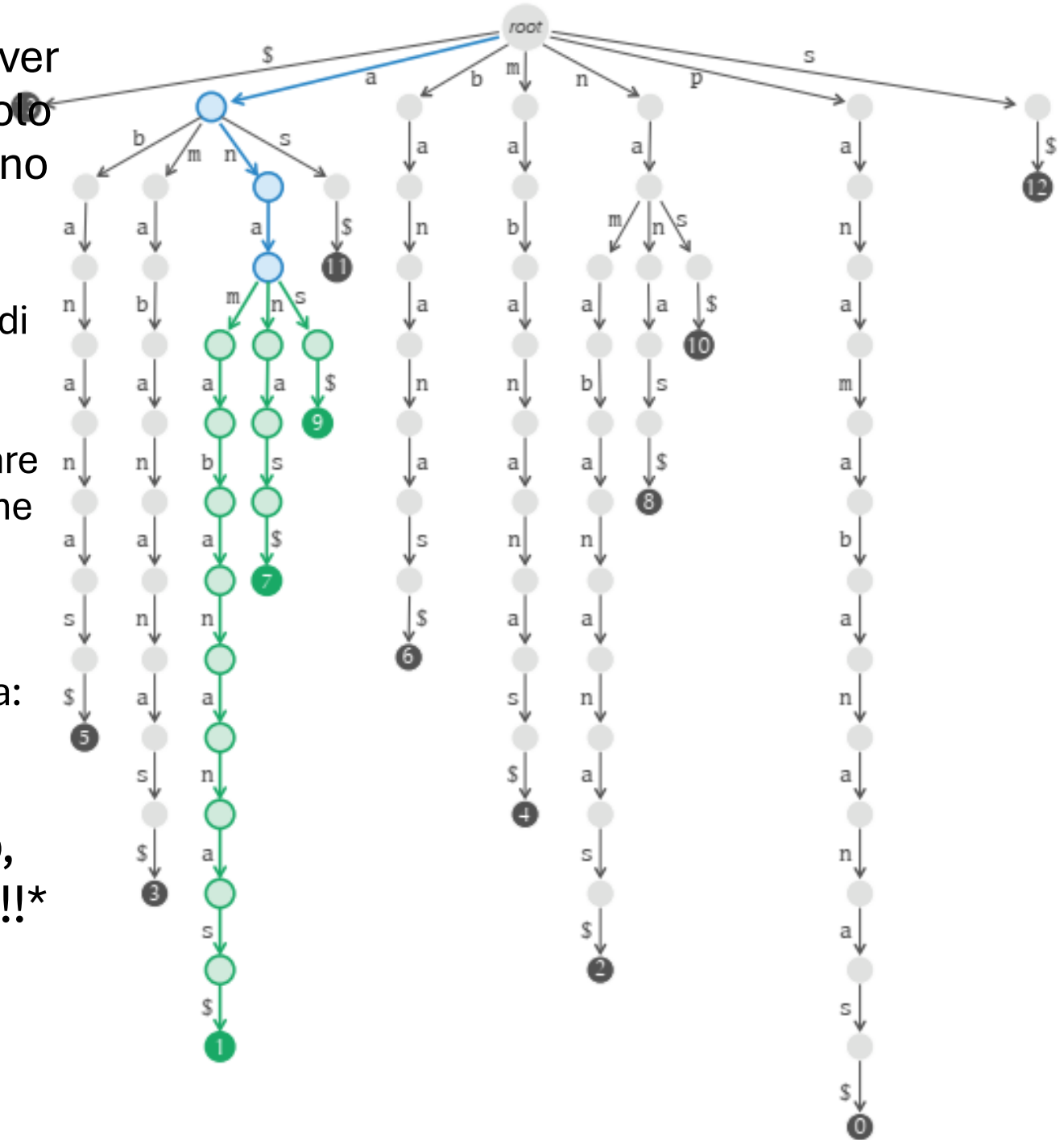
Gli alberi di suffissi permettono di non dover scorrere l'intero genoma ogni volta, ma solo percorrere quei percorsi che corrispondono alla nostra sequenza.

Per una sequenza di lunghezza ***m*** e un genoma di lunghezza ***n***:

- grep dovrebbe (nel peggiore dei casi) comparare ogni posizione della sequenza con ogni posizione del genoma: complessità O(nxm)

- un albero di suffissi va direttamente alla ramificazione per ogni posizione della sequenza: complessità O(m).

Tuttavia deve ricordarsi tutto l'albero, quindi richiede un sacco di memoria!!*



*Ad esempio per un genoma come quello umano servirebbero 60Gb di memoria.

Gli «alberi di suffissi» possono essere riorganizzati in «array di suffissi»

Suffissi ordinati alfanumericamente*	Posizione di inizio
$	13
abananas$	5
amabananas$	3
anamabananas$	1
ananas$	7
anas$	9
as$	11
bananas$	6
mabananas$	4
namabananas$	2
nanas$	8
nas$	10
panamabananas$	0
s$	12

***alfanumericamente:** alfabeticamente ma tenendo conto anche di numeri e simboli

Gli «alberi di suffissi» possono essere riorganzzati in «array di suffissi»

Suffissi ordinati alfanumericamente*	Posizione di inizio
$	13
abananas$	5
amabananas$	3
anamabananas$	1
ananas$	7
anas$	9
as$	11
bananas$	6
mabananas$	4
namabananas$	2
nanas$	8
nas$	10
panamabananas$	0
s$	12

Se volessi salvarla in memoria, dovrei creare una tabella di L righe e di L*L/2 caratteri, che per il genoma umano di sono 3000000000 righe e circa 10^{18} caratteri!

***alfanumericamente**: alfabeticamente ma tenendo conto anche di numeri e simboli

Gli «alberi di suffissi» possono essere riorganzzati in «array di suffissi»

Suffissi ordinati alfanumericamente*	Posizione di inizio
$	13
abananas$	5
amabananas$	3
anamabananas$	1
ananas$	7
anas$	9
as$	11
bananas$	6
mabananas$	4
namabananas$	2
nanas$	8
nas$	10
panamabananas$	0
s$	12

Ma possiamo evitare di salvarla con un algoritmo intelligente!!

***alfanumericamente**: alfabeticamente ma tenendo conto anche di numeri e simboli

Gli «alberi di suffissi» possono essere riorganzzati in «array di suffissi»

Suffissi ordinati alfanumericamente*	Posizione di inizio
$	13
abananas$	5
amabananas$	3
anamabananas$	1
ananas$	7
anas$	9
as$	11
bananas$	6
mabananas$	4
namabananas$	2
nanas$	8
nas$	10
panamabananas$	0
s$	12

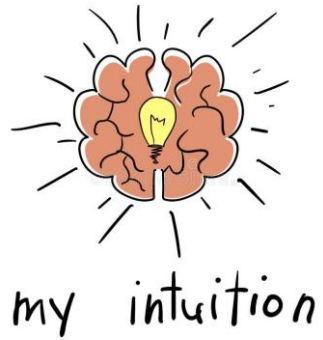
Ma possiamo evitare di salvarla con un algoritmo intelligente!!

***alfanumericamente**: alfabeticamente ma tenendo conto anche di numeri e simboli

Mappare con gli «array di suffissi»

Esempio: mappare ***anas*** *con* ***«binary search»***

Suffissi ordinati alfanumericamente*	Posizione di inizio
$	13
abananas$	5
amabananas$	3
anamabananas$	1
ananas$	7
anas$	9
as$	11
bananas$	6
mabananas$	4
namabananas$	2
nanas$	8
nas$	10
panamabananas$	0
s$	12



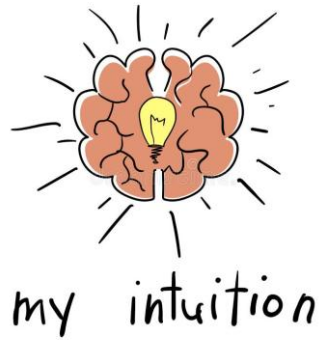
Due intuizioni per un algoritmo di mappatura (*binary search*):

- ***Non dobbiamo ricordare*** tutta la tabella, perché dalle posizioni possiamo trovare subito i suffissi!

Mappare con gli «array di suffissi»

Esempio: mappare ***anas*** *con* ***«binary search»***

Suffissi ordinati alfanumericamente*	Posizione di inizio
$	13
abananas$	5
amabananas$	3
anamabananas$	1
ananas$	7
anas$	9
as$	11
bananas$	6
mabananas$	4
namabananas$	2
nanas$	8
nas$	10
panamabananas$	0
s$	12



Due intuizioni per un algoritmo di mappatura (*binary search*):

- ***Non dobbiamo ricordare*** tutta la tabella, perché dalle posizioni possiamo trovare subito i suffissi!
- Possiamo cercare esplorando «a caso» ***dividendo a metà il genoma*** fino a che non troviamo la nostra sequenze

Mappare con gli «array di suffissi»

Esempio: mappare ***anas*** *con* ***«binary search»***

Suffissi ordinati alfanumericamente*	Posizione di inizio
$	13
abananas$	5
amabananas$	3
anamabananas$	1
ananas$	7
anas$	9
as$	11
bananas$	6
mabananas$	4
namabananas$	2
nanas$	8
nas$	10
panamabananas$	0
s$	12

1) Costruiamo l'array.

Mappare con gli «array di suffissi»

Esempio: mappare ***anas***

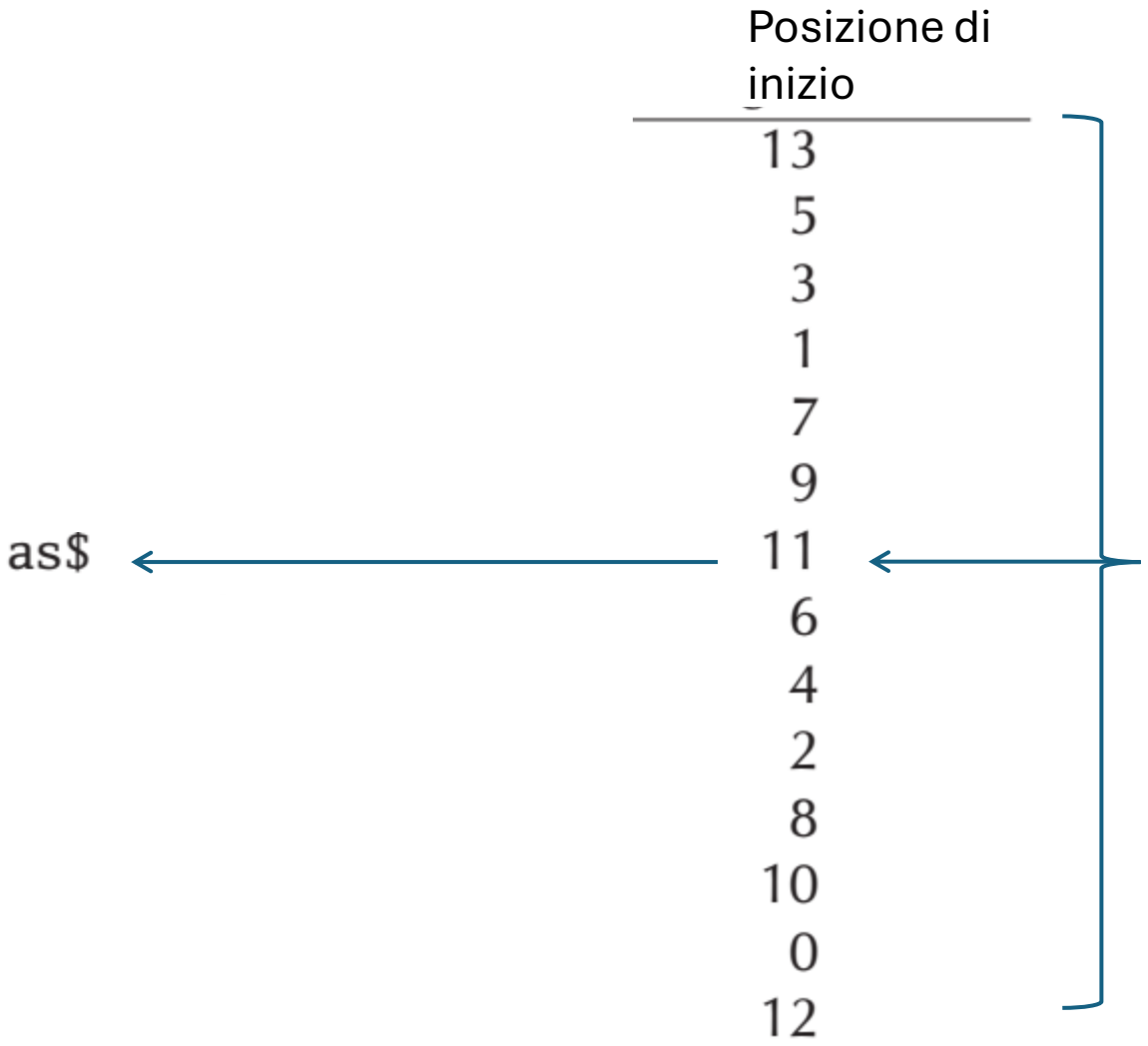
Posizione di inizio
13
5
3
1
7
9
11
6
4
2
8
10
0
12

2) Ricordiamo le posizioni e dimentichiamo i suffissi!!!

***alfanumericamente**: alfabeticamente ma tenendo conto anche di numeri e simboli

Mappare con gli «array di suffissi»

Esempio: mappare ***anas***

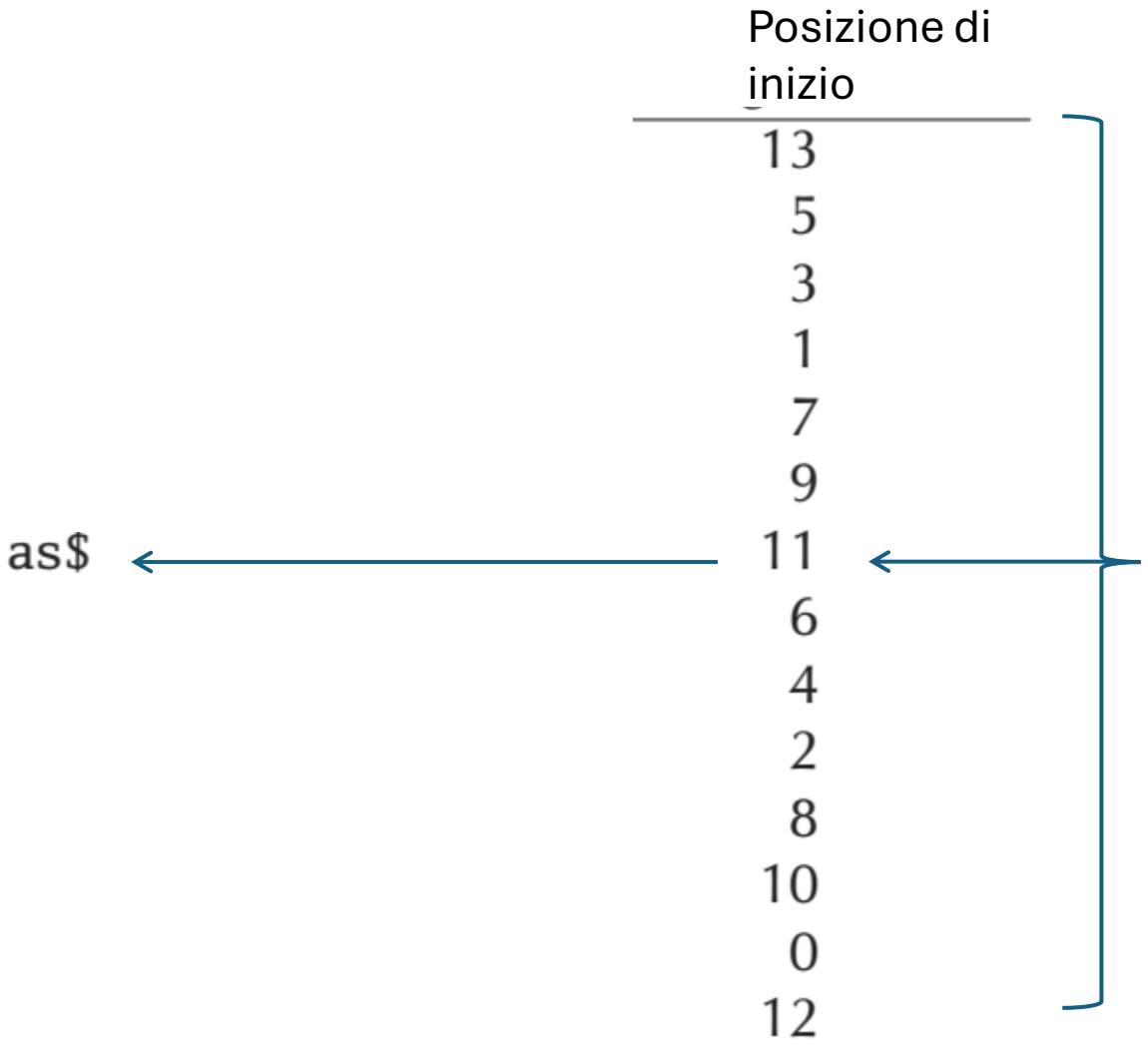


3) Guardiamo alla posizione «di mezzo» (qui 11). Visto che conosciamo la posizione, possiamo estrarre al volo il suffisso in quella posizione.

***alfanumericamente**: alfabeticamente ma tenendo conto anche di numeri e simboli

Mappare con gli «array di suffissi»

Esempio: mappare ***anas***



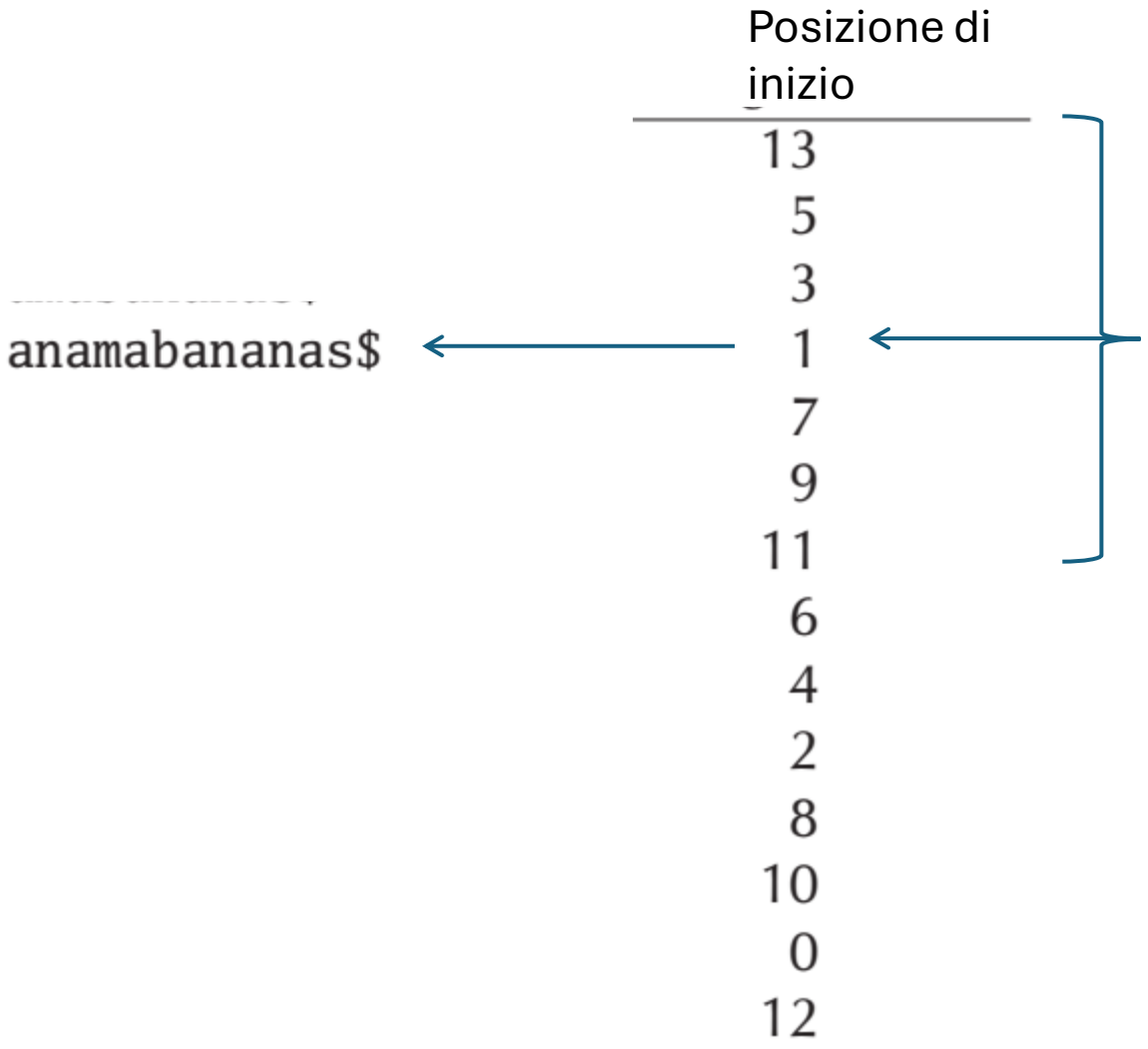
4) Il suffisso in posizione 11 ha valore alfanumerico maggiore o minore della mia sequenza ***anas***?

anas dovrebbe venir prima, quindi cerchiamo sopra!

***alfanumericamente**: alfabeticamente ma tenendo conto anche di numeri e simboli

Mappare con gli «array di suffissi»

Esempio: mappare ***anas***

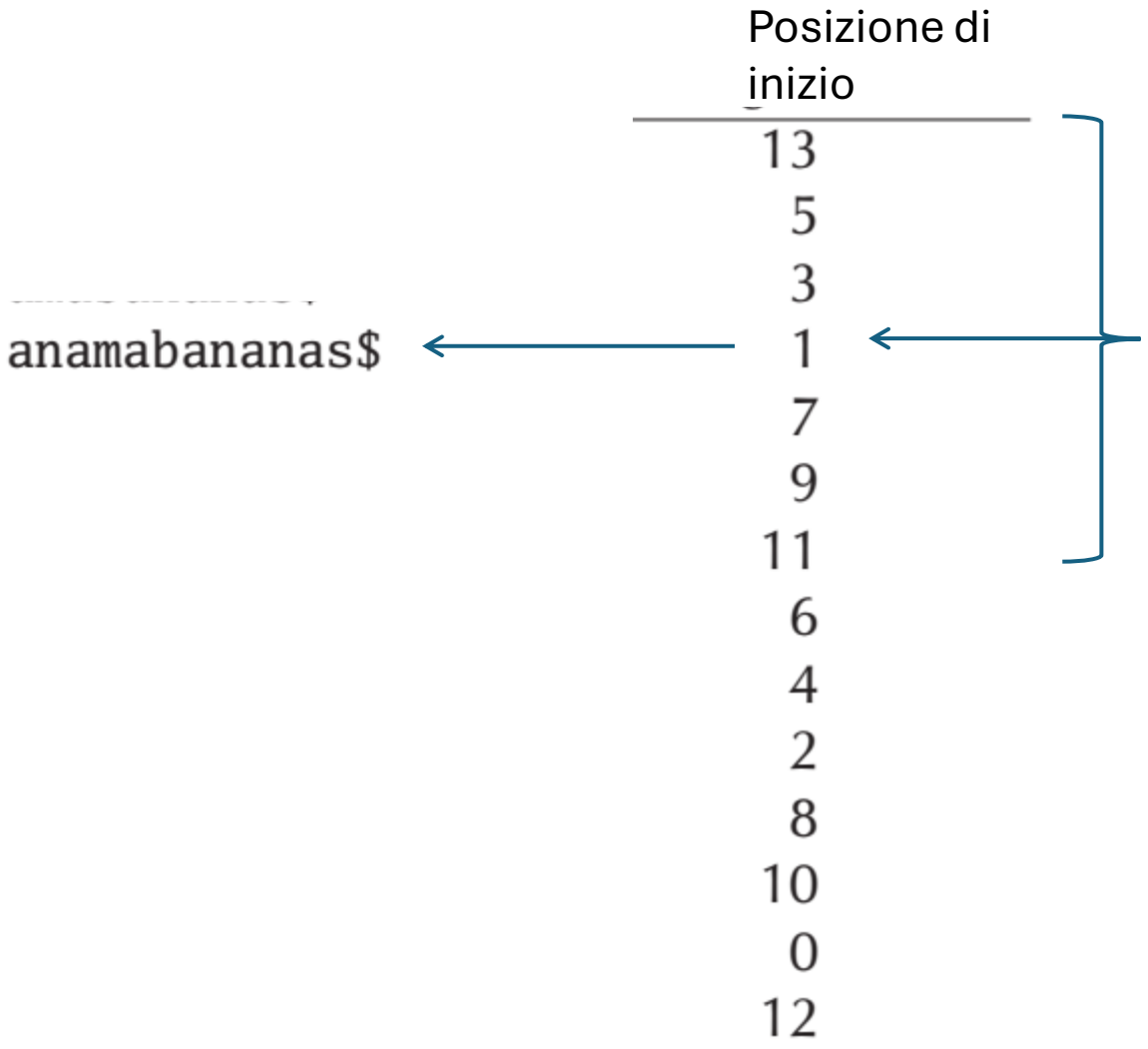


4) Cerchiamo nel mezzo del nuovo intervallo (posizione 1), estraendo il suffisso corrispondente

***alfanumericamente**: alfabeticamente ma tenendo conto anche di numeri e simboli

Mappare con gli «array di suffissi»

Esempio: mappare ***anas***



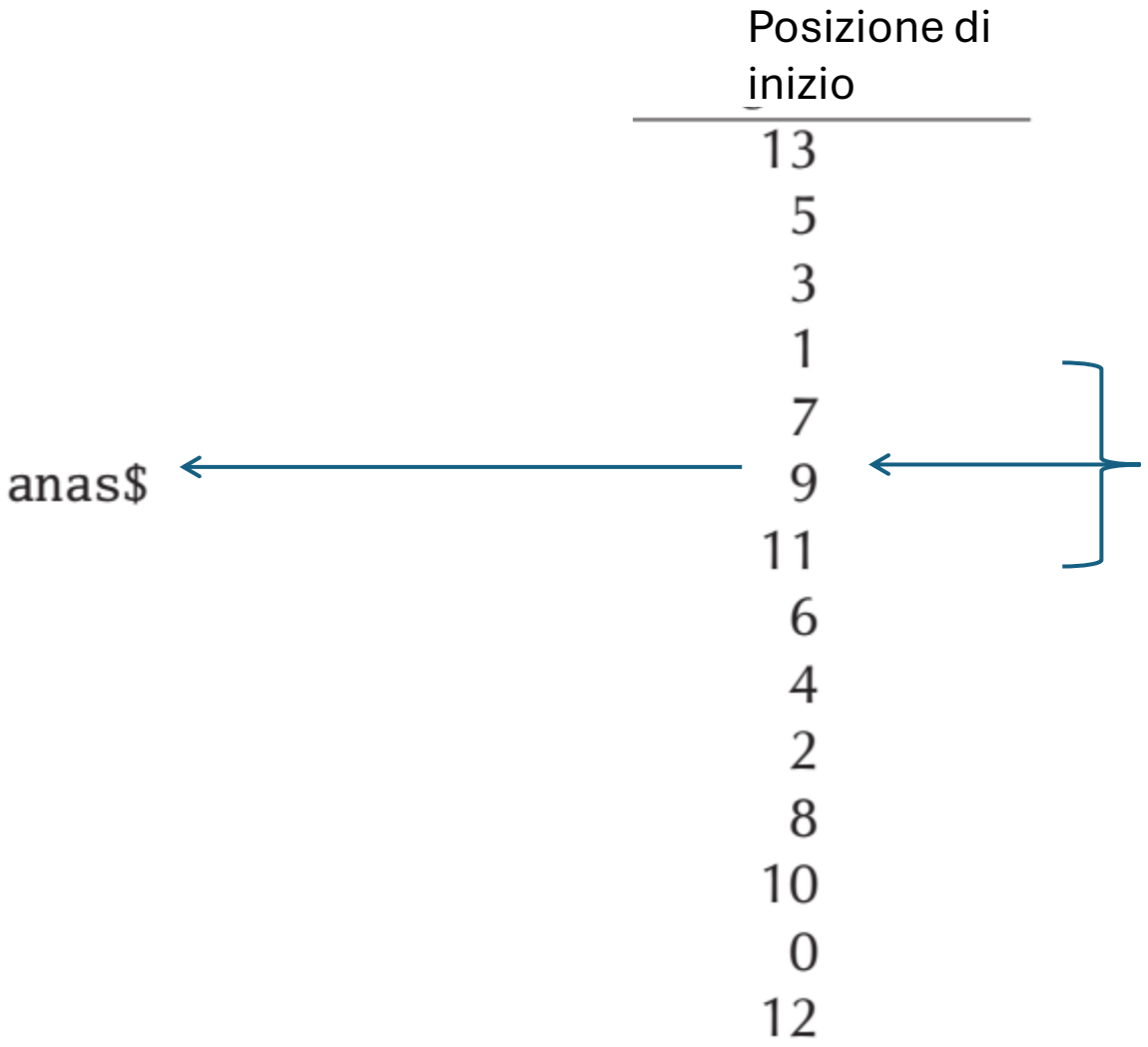
4) Cerchiamo nel mezzo del nuovo intervallo (posizione 1), estraendo il suffisso corrispondente.

anas dovrebbe venire dopo (s viene dopo la m di anamabananas)

***alfanumericamente**: alfabeticamente ma tenendo conto anche di numeri e simboli

Mappare con gli «array di suffissi»

Esempio: mappare ***anas***



5) Cerchiamo nel mezzo del nuovo intervallo (posizione 9), e troviamo ***anas***$

Mappare con gli «array di suffissi»

Esempio: mappare ***anas***

Suffissi ordinati alfanumericamente*	Posizione di inizio
$	13
abananas$	5
amabananas$	3
anamabananas$	1
ananas$	7
anas$	9
as$	11
bananas$	6
mabananas$	4
namabananas$	2
nanas$	8
nas$	10
panamabananas$	0
s$	12

Non è necessario tenere in memoria l'intera tabella!

Possiamo ricalcolarla al volo grazie al fatto che conosciamo le posizioni dei suffissi!

Pseudocodice

```
PATTERNMATCHINGWITHSUFFIXARRAY(Text, Pattern, SUFFIXARRAY)
    minIndex ← 0
    maxIndex ← |Text|
    while minIndex < maxIndex
        midIndex ← (minIndex + maxIndex) / 2
        if Pattern > suffix of Text starting at position SUFFIXARRAY(midIndex)
            minIndex ← midIndex + 1
        else
            maxIndex ← midIndex
    first ← minIndex
    maxIndex ← |Text|
    while minIndex < maxIndex
        midIndex ← (minIndex + maxIndex) / 2
        if Pattern < suffix of Text starting at position SUFFIXARRAY(midIndex)
            maxIndex ← midIndex
        else
            minIndex ← midIndex + 1
    last ← maxIndex
    if first > last
        return "Pattern does not appear in Text"
    else
        return (first, last)
```

Gli array di suffissi sono una buona soluzione

e fino ai primi anni 2000 erano il metodo usato per mappare

Tuttavia la stessa operazione va rifatta moltissime volte: un piccolo sequenziatore Mi-seq può produrre 30 milioni di reads, mentre una singola run di un sequenziatore Novaseq (Illumina) può produrre $20x10^9$ di sequenze!

Se doveste salvare efficientemente queste immagini risparmiando memoria, come fareste?

- Se volete approfondire l’argomento (assolutamente non necessario per il corso) consiglio: Information Theory, A tutorial Introduction, James V Stone

Se doveste salvare efficientemente queste immagini risparmiando memoria, come fareste?

- Se volete approfondire l’argomento (assolutamente non necessario per il corso) consiglio: Information Theory, A tutorial Introduction, James V Stone

La prima immagine non è altro che 00000000011000000000

- Se volete approfondire l’argomento (assolutamente non necessario per il corso) consiglio: Information Theory, A tutorial Introduction, James V Stone

La prima immagine non è altro che 00000000011000000000

Potremmo comprimerla ricordando solo le posizioni degli 1

- Se volete approfondire l’argomento (assolutamente non necessario per il corso) consiglio: Information Theory, A tutorial Introduction, James V Stone

Run-length encoding

La seconda immagine è anche rappresentabile come:

000000000111111110000110000010101 01

- Se volete approfondire l'argomento (assolutamente non necessario per il corso) consiglio: Information Theory, A tutorial Introduction, James V Stone

Run-length encoding

La seconda immagine è anche rappresentabile come:
000000000111111110000110000010101 01

Visto che gli 0 e gli 1 sono in lunghe file, potremmo comprimerla ricordando quanti 0 abbiamo, e poi quanti 1: 908140....

- Se volete approfondire l'argomento (assolutamente non necessario per il corso) consiglio: Information Theory, A tutorial Introduction, James V Stone

Run-length encoding

Sequenza originale	sequenza compressa
TTTTTGGGAAAACCCCCCA	5T3G4A6C1A

Run-length encoding

Sequenza originale

TTTTTGGGAAAACCCCCCA

sequenza compressa

5T3G4A6C1A

Run-length encoding

Sequenza originale

TTTTTGGGAAAACCCCCCA

sequenza compressa

5T3G4A6C1A

Numero di ripetizioni

Carattere

Questo è come sono compressi alcuni comuni formati di immagine: file .TIFF e .BMP

Run-length encoding

Sequenza originale	sequenza compressa
TTTTTGGGAAAACCCCCCA	5T3G4A6C1A

Tuttavia il genoma non appare cosi, ma ben piu’ disordinato, con ripetizioni anche complesse ogni tanto:

…AGCTAGCGTGCATGGGATATATAGAGCGCGCGCTGAGTGACTAGCTATATA…

Run-length encoding

Sequenza originale	sequenza compressa
TTTTTGGGAAAACCCCCCA	5T3G4A6C1A

Tuttavia il genoma non appare cosi, ma ben piu' disordinato, con ripetizioni anche complesse ogni tanto:

...AGCTAGCGTGCATGGGATATATAGAGCGCGCGCTGAGTGACTAGCTATATA...

Potremmo ordinarlo:
AAAAAAAAAAAAAAAAACCCCCCCCCCCCCCCCCGGGGGGGTTTTTTTTTT

Sarebbe una buona idea?

La trasformata di Burrow-Wheeler

O come piace pensarla a me, la trasformata di Heng Li*

* Autore di molti dei tool che useremo a lezione e di altre meraviglie come PSMC, etc.

La trasformata di Burrow-Wheeler

Cyclic Rotations
panamabananas$
$panamabananas
s$panamabanana
as$panamabanan
nas$panamabana
anas$panamaban
nanas$panamaba
ananas$panamab
bananas$panama
abananas$panam
mabananas$pana
amabananas$pan
namabananas$pa
anamabananas$p

La trasformata di Burrow-Wheeler

Cyclic Rotations	*M*("panamabananas$")
panamabananas$	$ p a n a m a b a n a n a **s**
$panamabananas	a b a n a n a s $ p a n a **m**
s$panamabanana	a m a b a n a n a s $ p a **n**
as$panamabanan	a n a m a b a n a n a s $ **p**
nas$panamabana	a n a n a s $ p a n a m a **b**
anas$panamaban	a n a s $ p a n a m a b a **n**
nanas$panamaba	a s $ p a n a m a b a n a **n**
ananas$panamab	b a n a n a s $ p a n a m **a**
bananas$panama	m a b a n a n a s $ p a n **a**
abananas$panam	n a m a b a n a n a s $ p **a**
mabananas$pana	n a n a s $ p a n a m a b **a**
amabananas$pan	n a s $ p a n a m a b a n **a**
namabananas$pa	p a n a m a b a n a n a s **$**
anamabananas$p	s $ p a n a m a b a n a n **a**

La trasformata di Burrow-Wheeler

Cyclic Rotations	*M*("panamabananas$")
panamabananas$	$ p a n a m a b a n a n a **s**
$panamabananas	a b a n a n a s $ p a n a **m**
s$panamabanana	a m a b a n a n a s $ p a **n**
as$panamabanan	a n a m a b a n a n a s $ **p**
nas$panamabana	a n a n a s $ p a n a m a **b**
anas$panamaban	a n a s $ p a n a m a b a **n**
nanas$panamaba	a s $ p a n a m a b a n a **n**
ananas$panamab	b a n a n a s $ p a n a m **a**
bananas$panama	m a b a n a n a s $ p a n **a**
abananas$panam	n a m a b a n a n a s $ p **a**
mabananas$pana	n a n a s $ p a n a m a b **a**
amabananas$pan	n a s $ p a n a m a b a n **a**
namabananas$pa	p a n a m a b a n a n a s **$**
anamabananas$p	s $ p a n a m a b a n a n **a**

La trasformata di ***panamabananas$*** è l'ultima colonna: **smnpbnnaaaa$a**

La trasformata di Burrow-Wheeler

Cyclic Rotations	*M*("panamabananas$")
panamabananas$	$ p a n a m a b a n a n a **s**
$panamabananas	a b a n a n a s $ p a n a **m**
s$panamabanana	a m a b a n a n a s $ p a **n**
as$panamabanan	a n a m a b a n a n a s $ **p**
nas$panamabana	a n a n a s $ p a n a m a **b**
anas$panamaban	a n a s $ p a n a m a b a **n**
nanas$panamaba	a s $ p a n a m a b a n a **n**
ananas$panamab	b a n a n a s $ p a n a m **a**
bananas$panama	m a b a n a n a s $ p a n **a**
abananas$panam	n a m a b a n a n a s $ p **a**
mabananas$pana	n a n a s $ p a n a m a b **a**
amabananas$pan	n a s $ p a n a m a b a n **a**
namabananas$pa	p a n a m a b a n a n a s **$**
anamabananas$p	s $ p a n a m a b a n a n **a**

BWT(***panamabananas$)*** = **smnpbnnaaaa$a**

```
nd Corey (1).  They kindly made their manuscript availa ..... a
nd criticism, especially on interatomic distances.  We  ..... a
nd cytosine.  The sequence of bases on a single chain d ..... a
nd experimentally (3,4) that the ratio of the amounts o ..... u
nd for this reason we shall not comment on it.  We wish ..... a
nd guanine (purine) with cytosine (pyrimidine).  In oth ..... a
nd ideas of Dr.  M. H. F. Wilkins, Dr.  R. E. Franklin  ..... a
nd its water content is rather high.  At lower water co ..... a
nd pyrimidine bases.  The planes of the bases are perpe ..... a
nd stereochemical arguments.  It has not escaped our no ..... a
nd that only specific pairs of bases can bond together  ..... u
nd the atoms near it is close to Furberg's 'standard co ..... a
nd the bases on the inside, linked together by hydrogen ..... a
nd the bases on the outside.  In our opinion, this stru ..... a
nd the other a pyrimidine for bonding to occur.  The hy ..... a
nd the phosphates on the outside.  The configuration of ..... a
nd the ration of guanine to cytosine, are always very c ..... a
nd the same axis (see diagram).  We have made the usual ..... u
nd their co-workers at King's College, London.  One of  ..... a
```

FIGURE 9.9 A few consecutive rows selected from M(*Text*), where *Text* is Watson and Crick's 1953 paper on the double helix. Rows beginning with "nd..." often end with "...a" because of the common occurrence of the word "and" in English, which causes runs of "a" in BWT(*Text*).

```
nd Corey (1).  They kindly made their manuscript availa ..... a
nd criticism, especially on interatomic distances.  We  ..... a
nd cytosine.  The sequence of bases on a single chain d ..... a
nd experimentally (3,4) that the ratio of the amounts o ..... u
nd for this reason we shall not comment on it.  We wish ..... a
nd guanine (purine) with cytosine (pyrimidine).  In oth ..... a
nd ideas of Dr.  M. H. F. Wilkins, Dr.  R. E. Franklin  ..... a
nd its water content is rather high.  At lower water co ..... a
nd pyrimidine bases.  The planes of the bases are perpe ..... a
nd stereochemical arguments.  It has not escaped our no ..... a
nd that only specific pairs of bases can bond together  ..... u
nd the atoms near it is close to Furberg's 'standard co ..... a
nd the bases on the inside, linked together by hydrogen ..... a
nd the bases on the outside.  In our opinion, this stru ..... a
nd the other a pyrimidine for bonding to occur.  The hy ..... a
nd the phosphates on the outside.  The configuration of ..... a
nd the ration of guanine to cytosine, are always very c ..... a
nd the same axis (see diagram).  We have made the usual ..... u
nd their co-workers at King's College, London.  One of  ..... a
```

Visto che **and** è molto comune in inglese, molte righe che cominciano con «**nd**» finiranno con «**a**». Quindi potrei provare a comprimere l'ultima colonna (dove ci sono molte a in fila) con run length encoding! Ma riuscirò mai a decomprimerla?

Esercizio:

- Ogni coppia estrae un bigliettino contenente una frase
- Costruite la trasformata di Burrow-Wheeler per le frasi nei bigliettini
- Per gli studenti in remoto, il vostro bigliettino dice ALLIGALLI

- Due sole regole:
  - Usate carta e penna, cosi che poi possiamo verificare se corretta dalla tabella
  - Non fate sbirciare agli altri gruppi!

(nella prossima fase dovranno decodificarle!)

Oh mi son perso, mi sento un citrullo…

Signor Matteo, peschi un bigliettino, ordini i suffissi in ordine alfabetico, e prenda l’ultima colonna. Ricordi, il $ va alla fine!

Esercizio:

- Ogni coppia estrae un bigliettino contenente una frase
- Costruite la trasformata di Burrow-Wheeler per le frasi nei bigliettini
- Comprimetela con run-length-encoding

- Due sole regole:
  - Usate carta e penna, cosi che poi possiamo verificare se corretta dalla tabella
  - Non fate sbirciare agli altri gruppi!

(nella prossima fase dovranno decodificarle!)

Esercizio:

Ora consegnate i bigliettini ad una coppia a caso

Come decodificare la trasformata di Burrow-Wheeler?

Decodificare la trasformata di Burrow-Wheeler

$	p	a	n	a	m	a	b	a	n	a	n	a	s
a_1	b	a	n	a	n	a	s	$	p	a	n	a	m
a_2	m	a	b	a	n	a	n	a	s	$	p	a	n
a_3	n	a	m	a	b	a	n	a	n	a	s	$	p
a_4	n	a	n	a	s	$	p	a	n	a	m	a	b
a_5	n	a	s	$	p	a	n	a	m	a	b	a	n
a_6	s	$	p	a	n	a	m	a	b	a	n	a	n
b	a	n	a	n	a	s	$	p	a	n	a	m	a
m	a	b	a	n	a	n	a	s	$	p	a	n	a
n	a	m	a	b	a	n	a	n	a	s	$	p	a
n	a	n	a	s	$	p	a	n	a	m	a	b	a
n	a	s	$	p	a	n	a	m	a	b	a	n	a
p	a	n	a	m	a	b	a	n	a	n	a	s	$
s	$	p	a	n	a	m	a	b	a	n	a	n	a

Decodificare la trasformata di Burrow-Wheeler

$ p a n a m a b a n a n a s
a_1 b a n a n a s $ p a n a m
a_2 m a b a n a n a s $ p a n
a_3 n a m a b a n a n a s $ p
a_4 n a n a s $ p a n a m a b
a_5 n a s $ p a n a m a b a n
a_6 s $ p a n a m a b a n a n
b a n a n a s $ p a n a m a
m a b a n a n a s $ p a n a
n a m a b a n a n a s $ p a
n a n a s $ p a n a m a b a
n a s $ p a n a m a b a n a
p a n a m a b a n a n a s $
s $ p a n a m a b a n a n a

L'ordine con cui una lettera uguale compare nella prima colonna viene definito **rango**

$pa_3na_2ma_1ba_4na_5na_6s$

Decodificare la trasformata di Burrow-Wheeler

	$	p	a	n	a	m	a	b	a	n	a	s
Rango 1	a_1	b	a	n	a	n	a	s	$	p	a	n	a	m
Rango 2	a_2	m	a	b	a	n	a	n	a	s	$	p	a	n
Rango 3	a_3	n	a	m	a	b	a	n	a	n	a	s	$	p
Rango 4	a_4	n	a	n	a	s	$	p	a	n	a	m	a	b
Rango 5	a_5	n	a	s	$	p	a	n	a	m	a	b	a	n
Rango 6	a_6	s	$	p	a	n	a	m	a	b	a	n	a	n
	b	a	n	a	n	a	s	$	p	a	n	a	m	a
	m	a	b	a	n	a	n	a	s	$	p	a	n	a
	n	a	m	a	b	a	n	a	n	a	s	$	p	a
	n	a	n	a	s	$	p	a	n	a	m	a	b	a
	n	a	s	$	p	a	n	a	m	a	b	a	n	a
	p	a	n	a	m	a	b	a	n	a	n	a	s	$
	s	$	p	a	n	a	m	a	b	a	n	a	n	a

L'ordine con cui una lettera uguale compare nella prima colonna viene definito **rango**

$$pa_3na_2ma_1ba_4na_5na_6s\$$$

$	p	a	n	a	m	a	b	a	n	a	s
a_1	b	a	n	a	n	a	s	$	p	a	n	a	m
a_2	m	a	b	a	n	a	n	a	s	$	p	a	n
a_3	n	a	m	a	b	a	n	a	n	a	s	$	p
a_4	n	a	n	a	s	$	p	a	n	a	m	a	b
a_5	n	a	s	$	p	a	n	a	m	a	b	a	n
a_6	s	$	p	a	n	a	m	a	b	a	n	a	n
b	a	n	a	n	a	s	$	p	a	n	a	m	a_1
m	a	b	a	n	a	n	a	s	$	p	a	n	a
n	a	m	a	b	a	n	a	n	a	s	$	p	a
n	a	n	a	s	$	p	a	n	a	m	a	b	a
n	a	s	$	p	a	n	a	m	a	b	a	n	a
p	a	n	a	m	a	b	a	n	a	n	a	s	$
s	$	p	a	n	a	m	a	b	a	n	a	n	a

$pa_3na_2ma_1ba_4na_5na_6s\$$

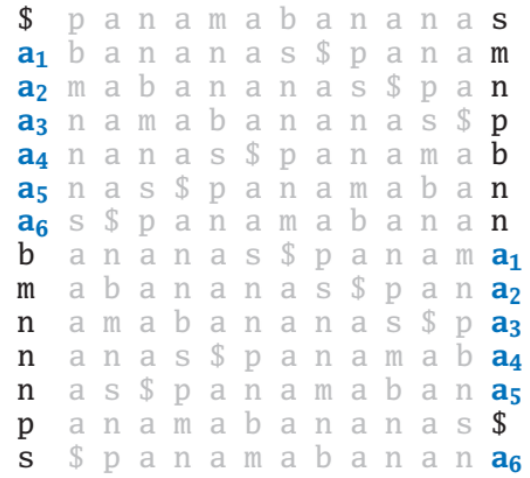
Notate che a_1 (la a prima di «banana») è anche la prima **a** nell'ultima colonna (la nostra trasformata).

$	p	a	n	a	m	a	b	a	n	a	n	a	s
a_1	b	a	n	a	n	a	s	$	p	a	n	a	m
a_2	m	a	b	a	n	a	n	a	s	$	p	a	n
a_3	n	a	m	a	b	a	n	a	n	a	s	$	p
a_4	n	a	n	a	s	$	p	a	n	a	m	a	b
a_5	n	a	s	$	p	a	n	a	m	a	b	a	n
a_6	s	$	p	a	n	a	m	a	b	a	n	a	n
b	a	n	a	n	a	s	$	p	a	n	a	m	a_1
m	a	b	a	n	a	n	a	s	$	p	a	n	a_2
n	a	m	a	b	a	n	a	n	a	s	$	p	a_3
n	a	n	a	s	$	p	a	n	a	m	a	b	a_4
n	a	s	$	p	a	n	a	m	a	b	a	n	a_5
p	a	n	a	m	a	b	a	n	a	n	a	s	$
s	$	p	a	n	a	m	a	b	a	n	a	n	a_6

$pa_3na_2ma_1ba_4na_5na_6s\$$

Questo non è un caso, le stesse lettere mantengono lo stesso rango nella prima e nell'ultima colonna.

Per capire perché, notate che se spostassimo le a dall'inizio alla fine (cioè guardiamo alla prima colonna e all'ultima + la a) a determinare l'ordine sono comunque le stesse lettere nella seconda colonna (e a scorrere): b,m,n,n,s.

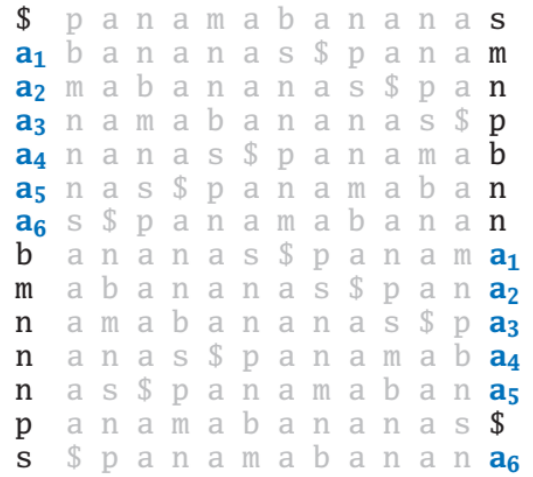


Prima colonna

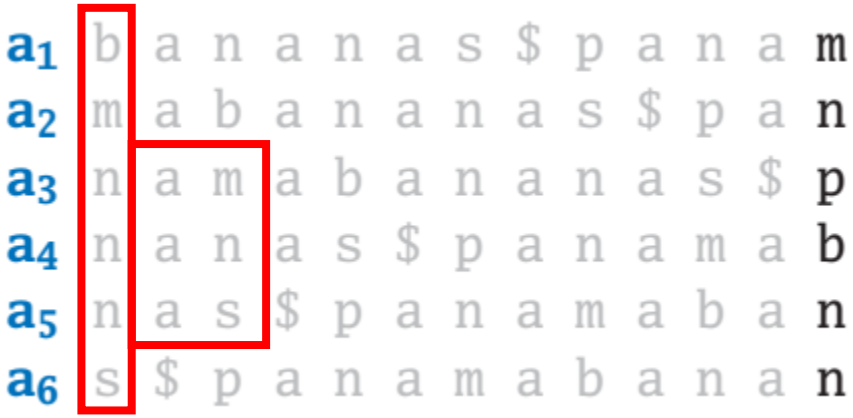
Spostare le ***a*** alla fine

Ultima colonna

Per capire perché, notate che se spostassimo le a dall'inizio alla fine (cioè guardiamo alla prima colonna e all'ultima + la a) a determinare l'ordine sono comunque le stesse lettere nella seconda colonna (e a scorrere): b,m,n,n,s.



Lettere che contano per l'ordine



Prima colonna

Spostare le ***a*** alla fine

Ultima colonna

$	p	a	n	a	m	a	b	a	n	a	n	a	s
a_1	b	a	n	a	n	a	s	$	p	a	n	a	m
a_2	m	a	b	a	n	a	n	a	s	$	p	a	n
a_3	n	a	m	a	b	a	n	a	n	a	s	$	p
a_4	n	a	n	a	s	$	p	a	n	a	m	a	b
a_5	n	a	s	$	p	a	n	a	m	a	b	a	n
a_6	s	$	p	a	n	a	m	a	b	a	n	a	n
b	a	n	a	n	a	s	$	p	a	n	a	m	a_1
m	a	b	a	n	a	n	a	s	$	p	a	n	a_2
n	a	m	a	b	a	n	a	n	a	s	$	p	a_3
n	a	n	a	s	$	p	a	n	a	m	a	b	a_4
n	a	s	$	p	a	n	a	m	a	b	a	n	a_5
p	a	n	a	m	a	b	a	n	a	n	a	s	$
s	$	p	a	n	a	m	a	b	a	n	a	n	a_6

$pa_3na_2ma_1ba_4na_5na_6s\$$

Proprietà First-Last

Uno specifico carattere mantiene lo stesso rango nella prima e nell'ultima colonna.

Possiamo utilizzare proprietà First-Last per ricostruire il testo originale dalla trasformata di Burrow-Wheeler

Come fare? Partiamo dalla trasformata: ***ard$rcaaaabb***

Possiamo utilizzare proprietà First-Last per ricostruire il testo originale dalla trasformata di Burrow-Wheeler

Come fare? Partiamo dalla trasformata: ***ard$rcaaaabb***

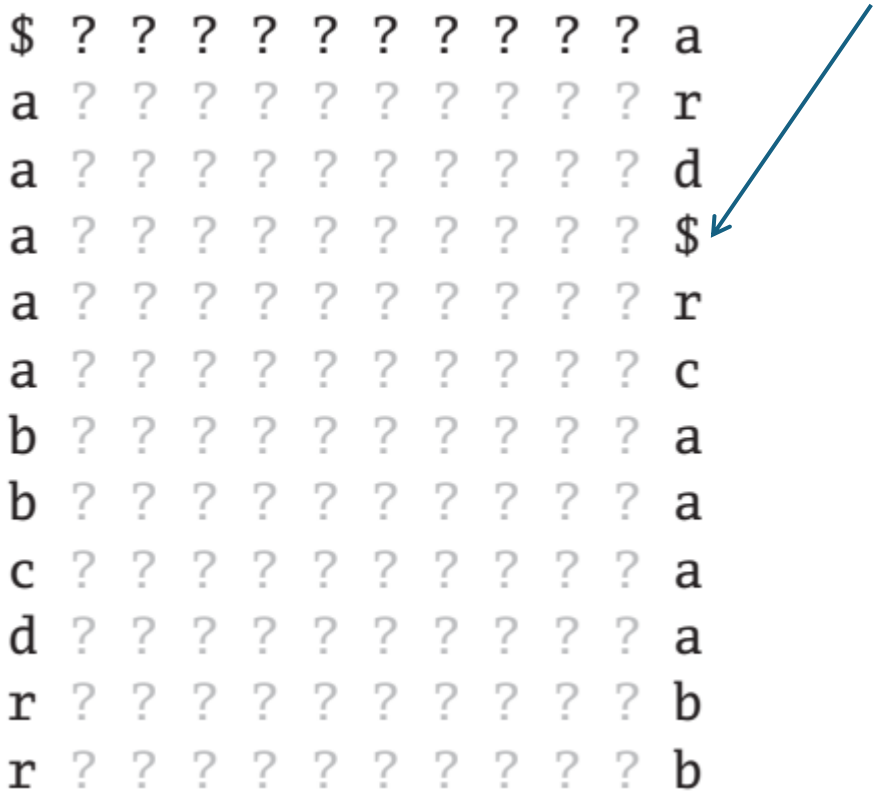
1) Ordiniamola alfanumericamente ottenendo la prima colonna

```
$ ? ? ? ? ? ? ? ? ? ? a
a ? ? ? ? ? ? ? ? ? ? r
a ? ? ? ? ? ? ? ? ? ? d
a ? ? ? ? ? ? ? ? ? ? $
a ? ? ? ? ? ? ? ? ? ? r
a ? ? ? ? ? ? ? ? ? ? c
b ? ? ? ? ? ? ? ? ? ? a
b ? ? ? ? ? ? ? ? ? ? a
c ? ? ? ? ? ? ? ? ? ? a
d ? ? ? ? ? ? ? ? ? ? a
r ? ? ? ? ? ? ? ? ? ? b
r ? ? ? ? ? ? ? ? ? ? b
```

Possiamo utilizzare proprietà First-Last per ricostruire il testo originale dalla trasformata di Burrow-Wheeler

Come fare? Partiamo dalla trasformata: ***ard$rcaaaabb***

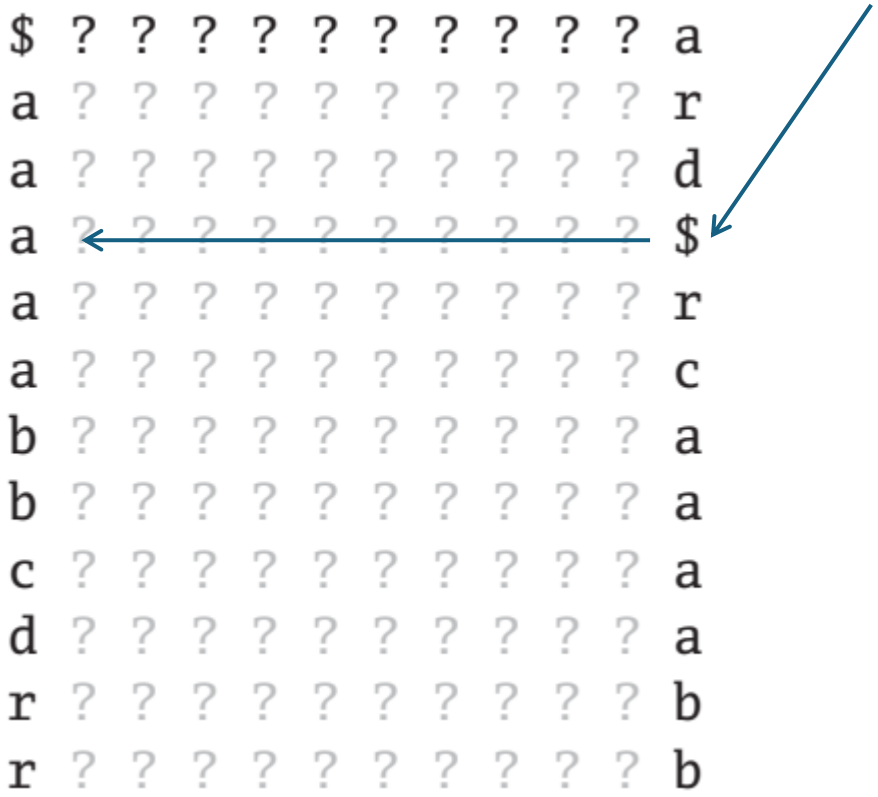
2) Sappiamo che l'ultimo carattere è $. Cerchiamolo.



Possiamo utilizzare proprietà First-Last per ricostruire il testo originale dalla trasformata di Burrow-Wheeler

Come fare? Partiamo dalla trasformata: ***ard$rcaaaabb***

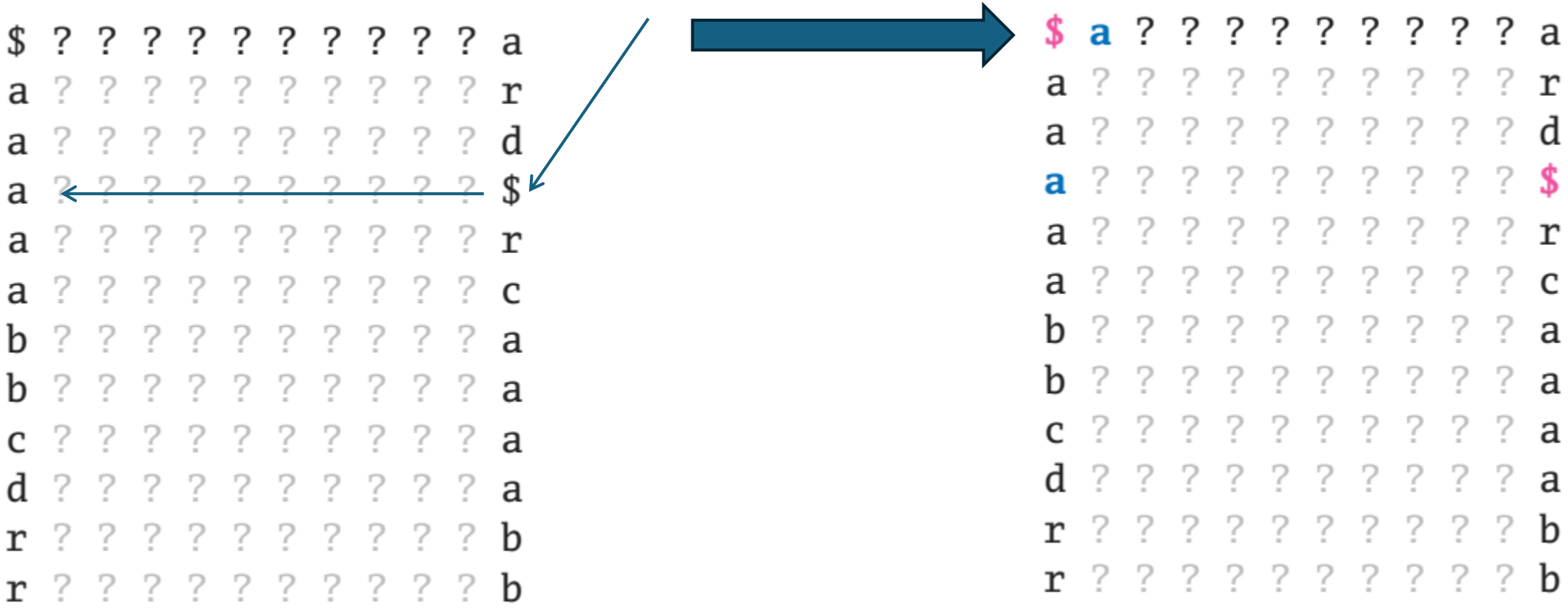
2) Sappiamo che l'ultimo carattere è $. Cerchiamolo. Quindi il primo carattere della sequenza doveva essere ***a!***



Possiamo utilizzare proprietà First-Last per ricostruire il testo originale dalla trasformata di Burrow-Wheeler

Come fare? Partiamo dalla trasformata: ***ard$rcaaaabb***

2) Sappiamo che l'ultimo carattere è $. Cerchiamolo. Quindi il primo carattere della sequenza doveva essere ***a!***



Possiamo utilizzare proprietà First-Last per ricostruire il testo originale dalla trasformata di Burrow-Wheeler

Come fare? Partiamo dalla trasformata: ***ard$rcaaaabb***

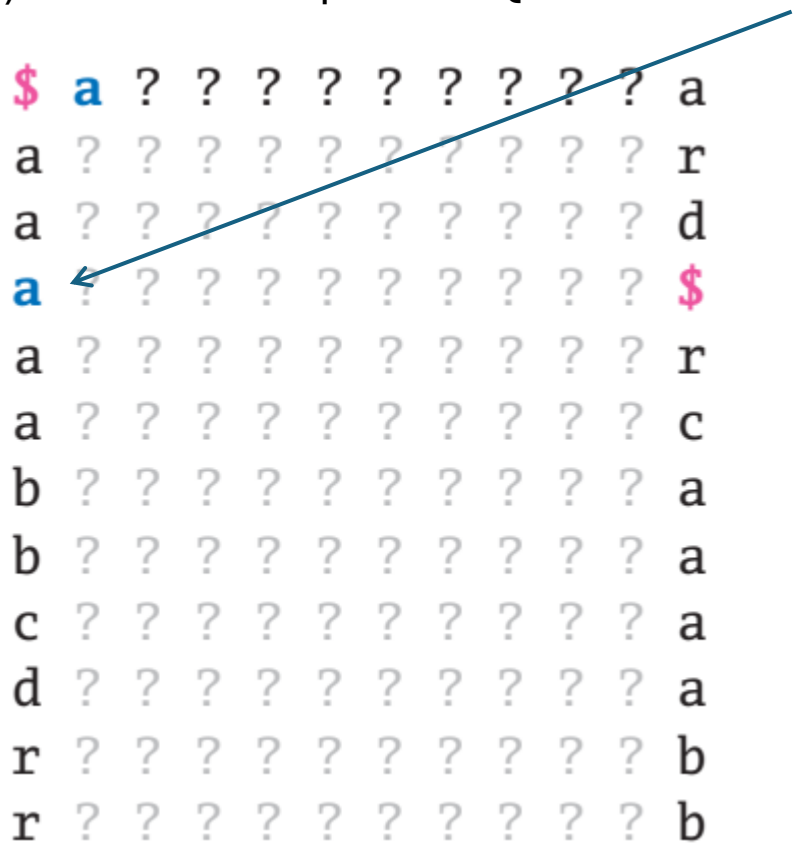
3) Cosa viene dopo la **a**?

$	**a**	?	?	?	?	?	?	?	?	?	a
a	?	?	?	?	?	?	?	?	?	?	r
a	?	?	?	?	?	?	?	?	?	?	d
a	?	?	?	?	?	?	?	?	?	?	$
a	?	?	?	?	?	?	?	?	?	?	r
a	?	?	?	?	?	?	?	?	?	?	c
b	?	?	?	?	?	?	?	?	?	?	a
b	?	?	?	?	?	?	?	?	?	?	a
c	?	?	?	?	?	?	?	?	?	?	a
d	?	?	?	?	?	?	?	?	?	?	a
r	?	?	?	?	?	?	?	?	?	?	b
r	?	?	?	?	?	?	?	?	?	?	b

Possiamo utilizzare proprietà First-Last per ricostruire il testo originale dalla trasformata di Burrow-Wheeler

Come fare? Partiamo dalla trasformata: ***ard$rcaaaabb***

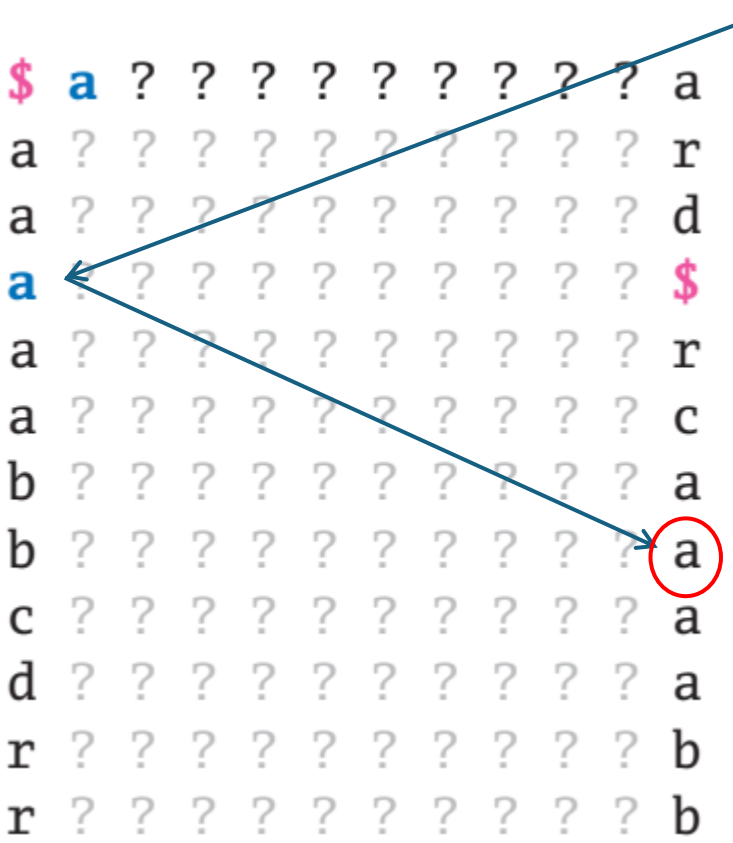
3) Cosa viene dopo la **a**? Questa **a** è la terza nella prima colonna.



Possiamo utilizzare proprietà First-Last per ricostruire il testo originale dalla trasformata di Burrow-Wheeler

Come fare? Partiamo dalla trasformata: ***ard$rcaaaabb***

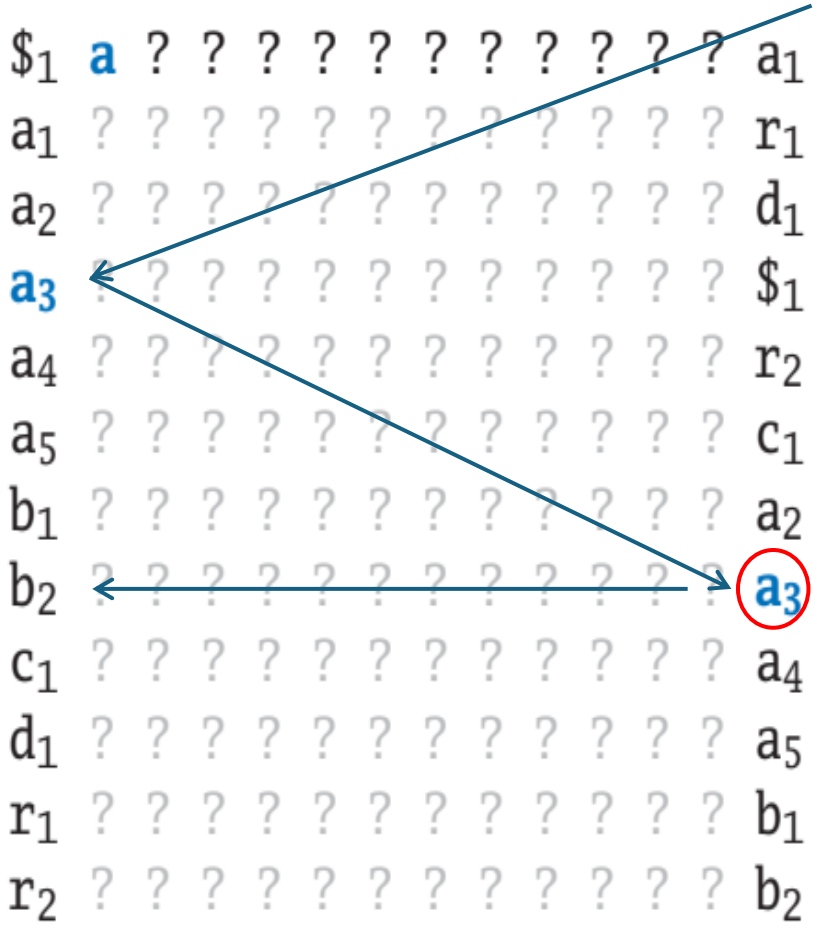
3) Cosa viene dopo la **a**? Questa **a** è la terza nella prima colonna. Dunque, grazie alla proprietà First-Last deve anche essere la terza nell'ultima colonna.



Possiamo utilizzare proprietà First-Last per ricostruire il testo originale dalla trasformata di Burrow-Wheeler

Come fare? Partiamo dalla trasformata: ***ard$rcaaaabb***

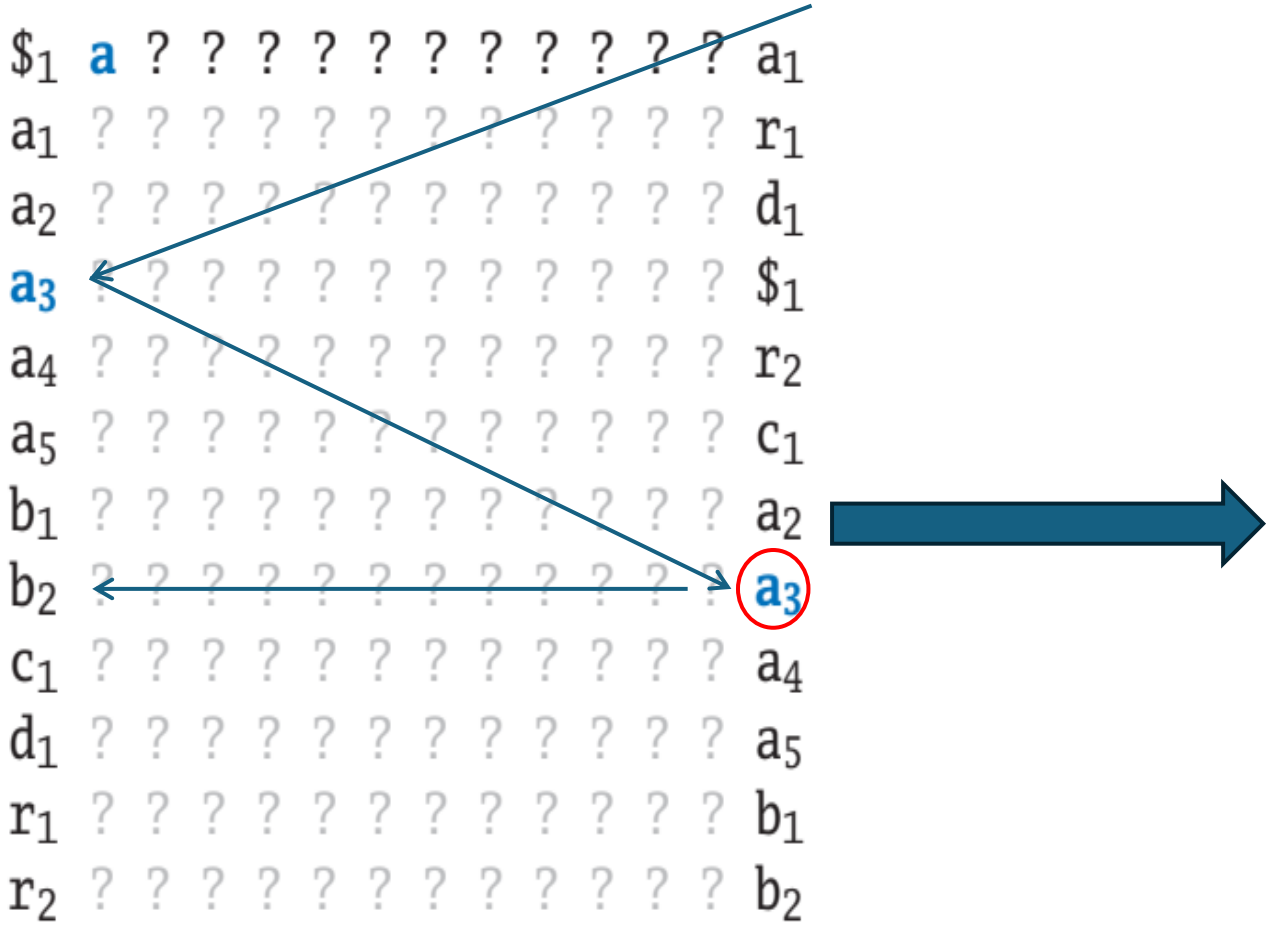
3) Cosa viene dopo la **a**? Questa **a** è la terza nella prima colonna. Dunque, grazie alla proprietà First-Last deve anche essere la terza nell'ultima colonna. Questo implica che fosse seguita nella sequenza originaria da una **b.**

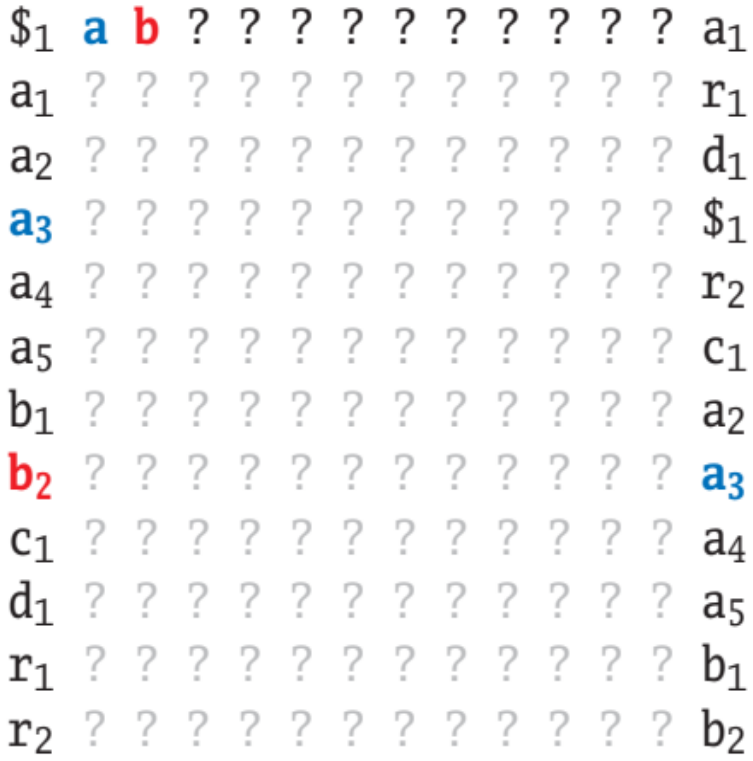


Possiamo utilizzare proprietà First-Last per ricostruire il testo originale dalla trasformata di Burrow-Wheeler

Come fare? Partiamo dalla trasformata: ***ard$rcaaaabb***

3) Cosa viene dopo la **a**? Questa **a** è la terza nella prima colonna. Dunque, grazie alla proprietà First-Last deve anche essere la terza nell'ultima colonna. Questo implica che fosse seguita nella sequenza originaria da una **b.**



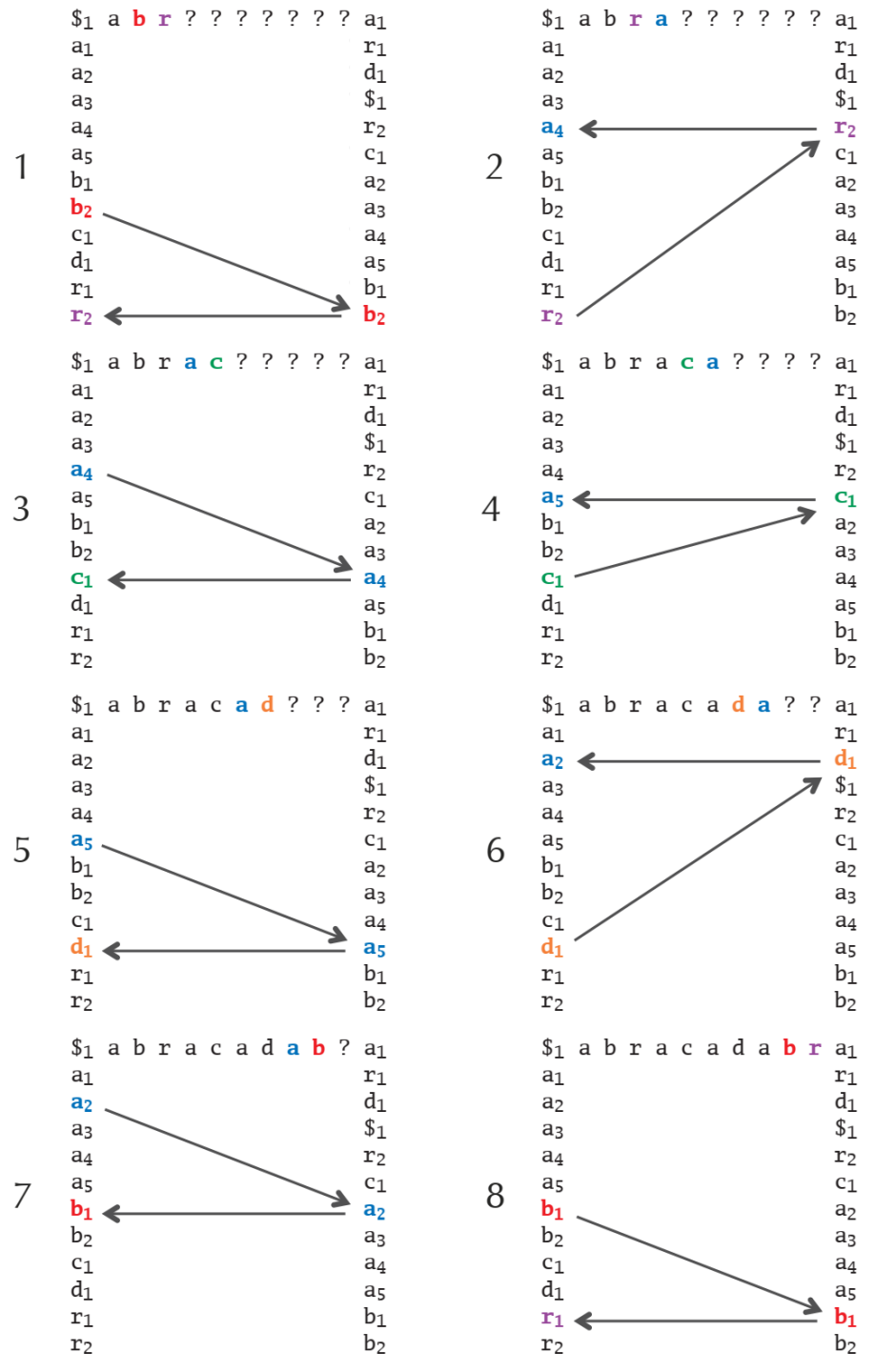


Questo processo può essere reiterato fino a risolvere l'intera sequenza:

abracadabra

quindi

BWT(abracadabra$)= *ard$rcaaaabb*



Esercizio:

- Scambiate i bigliettini con la trasformata di Burrow-Wheeler compressa con run-length-encoding
- Decodificatela!

Avete appreso come comprimere e decomprimere un genoma!

Questo algoritmo è alla base dei metodi di compressione e di «retrieval» per dati genomici piu' utilizzati (bgzip2, tabix).

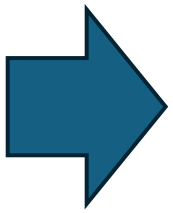
Come possiamo «mappare» sequenze: ritrovare un «pattern» dentro il nostro genoma?

Come ritrovare la nostra sequenza dentro BWT?

Idea simile ai suffix array: tutte le sequenze simili occorrono insieme

Esempio: dove è ***ana***?

Tutte le sequenze con ***ana***



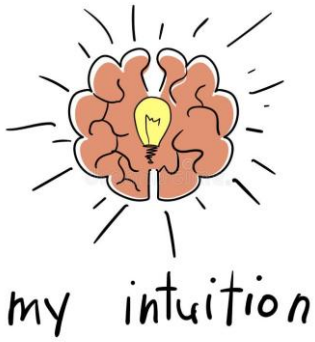
M(*Text*)	SUFFIXARRAY(*Text*)
$ p a n a m a b a n a n a s	13
a b a n a n a s $ p a n a m	5
a m a b a n a n a s $ p a n	3
a n a m a b a n a n a s $ p	**1**
a n a n a s $ p a n a m a b	**7**
a n a s $ p a n a m a b a n	**9**
a s $ p a n a m a b a n a n	11
b a n a n a s $ p a n a m a	6
m a b a n a n a s $ p a n a	4
n a m a b a n a n a s $ p a	2
n a n a s $ p a n a m a b a	8
n a s $ p a n a m a b a n a	10
p a n a m a b a n a n a s $	0
s $ p a n a m a b a n a n a	12

Come ritrovare la nostra sequenza dentro BWT?

Idea simile ai suffix array: tutte le sequenze simili occorrono insieme

Esempio: dove è ***ana***?

Tutte le sequenze con ***ana***



Possiamo cominciare dalla prima lettera ***a*** e piano piano trovare la successiva.

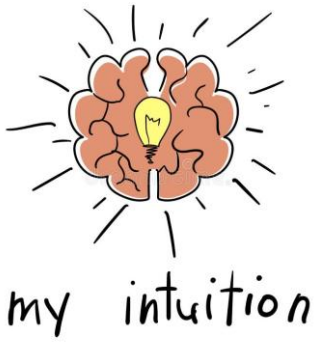
M(*Text*)	SUFFIXARRAY(*Text*)
$ p a n a m a b a n a n a s	13
a b a n a n a s $ p a n a m	5
a m a b a n a n a s $ p a n	3
a n a m a b a n a n a s $ p	**1**
a n a n a s $ p a n a m a b	**7**
a n a s $ p a n a m a b a n	**9**
a s $ p a n a m a b a n a n	11
b a n a n a s $ p a n a m a	6
m a b a n a n a s $ p a n a	4
n a m a b a n a n a s $ p a	2
n a n a s $ p a n a m a b a	8
n a s $ p a n a m a b a n a	10
p a n a m a b a n a n a s $	0
s $ p a n a m a b a n a n a	12

Come ritrovare la nostra sequenza dentro BWT?

Idea simile ai suffix array: tutte le sequenze simili occorrono insieme

Esempio: dove è ***ana***?

Tutte le sequenze con ***ana***



Possiamo cominciare dalla prima lettera ***a*** e piano piano trovare la successiva.
Ma non vogliamo ricostruire il suffix array intero! E allora?

M(*Text*)	SUFFIXARRAY(*Text*)
$ p a n a m a b a n a n a s	13
a b a n a n a s $ p a n a m	5
a m a b a n a n a s $ p a n	3
a n a m a b a n a n a s $ p	**1**
a n a n a s $ p a n a m a b	**7**
a n a s $ p a n a m a b a n	**9**
a s $ p a n a m a b a n a n	11
b a n a n a s $ p a n a m a	6
m a b a n a n a s $ p a n a	4
n a m a b a n a n a s $ p a	2
n a n a s $ p a n a m a b a	8
n a s $ p a n a m a b a n a	10
p a n a m a b a n a n a s $	0
s $ p a n a m a b a n a n a	12

Ritrovare la nostra sequenza ***ana***

1) Ordiniamo BWT(genoma) per ottenere la prima colonna
2) Selezioniamo l'**ultimo** carattere (***a***)

$_1$	p	a	n	a	m	a	b	a	n	a	n	a s$_1$
a$_1$	b	a	n	a	n	a	s	$	p	a	n	a m$_1$
a$_2$	m	a	b	a	n	a	n	a	s	$	p	a n$_1$
a$_3$	n	a	m	a	b	a	n	a	n	a	s	$ p$_1$
a$_4$	n	a	n	a	s	$	p	a	n	a	m	a b$_1$
a$_5$	n	a	s	$	p	a	n	a	m	a	b	a n$_2$
a$_6$	s	$	p	a	n	a	m	a	b	a	n	a n$_3$
b$_1$	a	n	a	n	a	s	$	p	a	n	a	m a$_1$
m$_1$	a	b	a	n	a	n	a	s	$	p	a	n a$_2$
n$_1$	a	m	a	b	a	n	a	n	a	s	$	p a$_3$
n$_2$	a	n	a	s	$	p	a	n	a	m	a	b a$_4$
n$_3$	a	s	$	p	a	n	a	m	a	b	a	n a$_5$
p$_1$	a	n	a	m	a	b	a	n	a	n	a	s $_1$
s$_1$	$	p	a	n	a	m	a	b	a	n	a	n a$_6$

Ritrovare la nostra sequenza ***ana***

1) Ordiniamo BWT(genoma) per ottenere la prima colonna
2) Selezioniamo l'**ultimo** carattere (***a***)
3) Teniamo solo quelle righe per cui la ***a*** è preceduta da una ***n***

$_1$	p	a	n	a	m	a	b	a	n	a	n	a	s$_1$
a$_1$	b	a	n	a	n	a	s	$	p	a	n	a	m$_1$
a$_2$	m	a	b	a	n	a	n	a	s	$	p	a	**n$_1$**
a$_3$	n	a	m	a	b	a	n	a	n	a	s	$	p$_1$
a$_4$	n	a	n	a	s	$	p	a	n	a	m	a	b$_1$
a$_5$	n	a	s	$	p	a	n	a	m	a	b	a	**n$_2$**
a$_6$	s	$	p	a	n	a	m	a	b	a	n	a	**n$_3$**
b$_1$	a	n	a	n	a	s	$	p	a	n	a	m	a$_1$
m$_1$	a	b	a	n	a	n	a	s	$	p	a	n	a$_2$
n$_1$	a	m	a	b	a	n	a	n	a	s	$	p	a$_3$
n$_2$	a	n	a	s	$	p	a	n	a	m	a	b	a$_4$
n$_3$	a	s	$	p	a	n	a	m	a	b	a	n	a$_5$
p$_1$	a	n	a	m	a	b	a	n	a	n	a	s	$_1$
s$_1$	$	p	a	n	a	m	a	b	a	n	a	n	a$_6$

Ritrovare la nostra sequenza *ana*

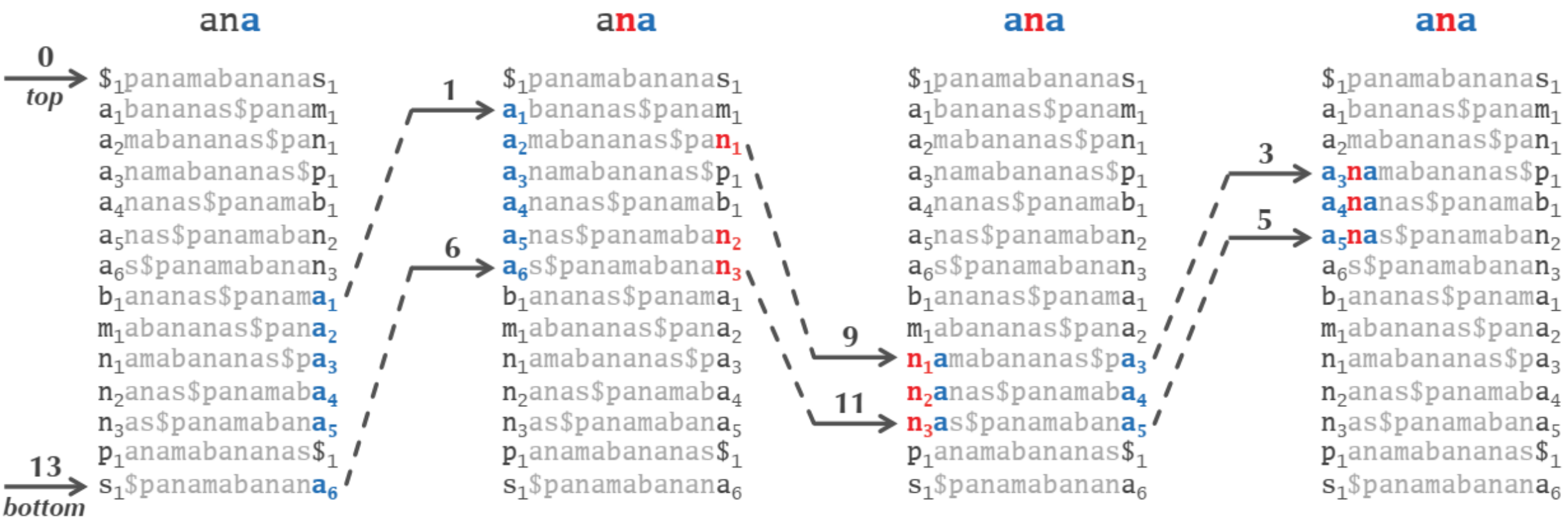
1) Ordiniamo BWT(genoma) per ottenere la prima colonna
2) Selezioniamo l'**ultimo** carattere (*a*)
3) Teniamo solo quelle righe per cui la *a* è preceduta da una *n*
4) E poi andiamo a tenere solo quelle righe per le quali le *n* (ora in prima colonna), sono precedute da delle *a* (in ultima colonna)

$_1$	p	a	n	a	m	a	b	a	n	a	n	a	s$_1$
a$_1$	b	a	n	a	n	a	s	$	p	a	n	a	m$_1$
a$_2$	m	a	b	a	n	a	n	a	s	$	p	a	n$_1$
a$_3$	n	a	m	a	b	a	n	a	n	a	s	$	p$_1$
a$_4$	n	a	n	a	s	$	p	a	n	a	m	a	b$_1$
a$_5$	n	a	s	$	p	a	n	a	m	a	b	a	n$_2$
a$_6$	s	$	p	a	n	a	m	a	b	a	n	a	n$_3$
b$_1$	a	n	a	n	a	s	$	p	a	n	a	m	a$_1$
m$_1$	a	b	a	n	a	n	a	s	$	p	a	n	a$_2$
n$_1$	a	m	a	b	a	n	a	n	a	s	$	p	a$_1$
n$_2$	a	n	a	s	$	p	a	n	a	m	a	b	a$_2$
n$_3$	a	s	$	p	a	n	a	m	a	b	a	n	a$_3$
p$_1$	a	n	a	m	a	b	a	n	a	n	a	s	$_1$
s$_1$	$	p	a	n	a	m	a	b	a	n	a	n	a$_6$

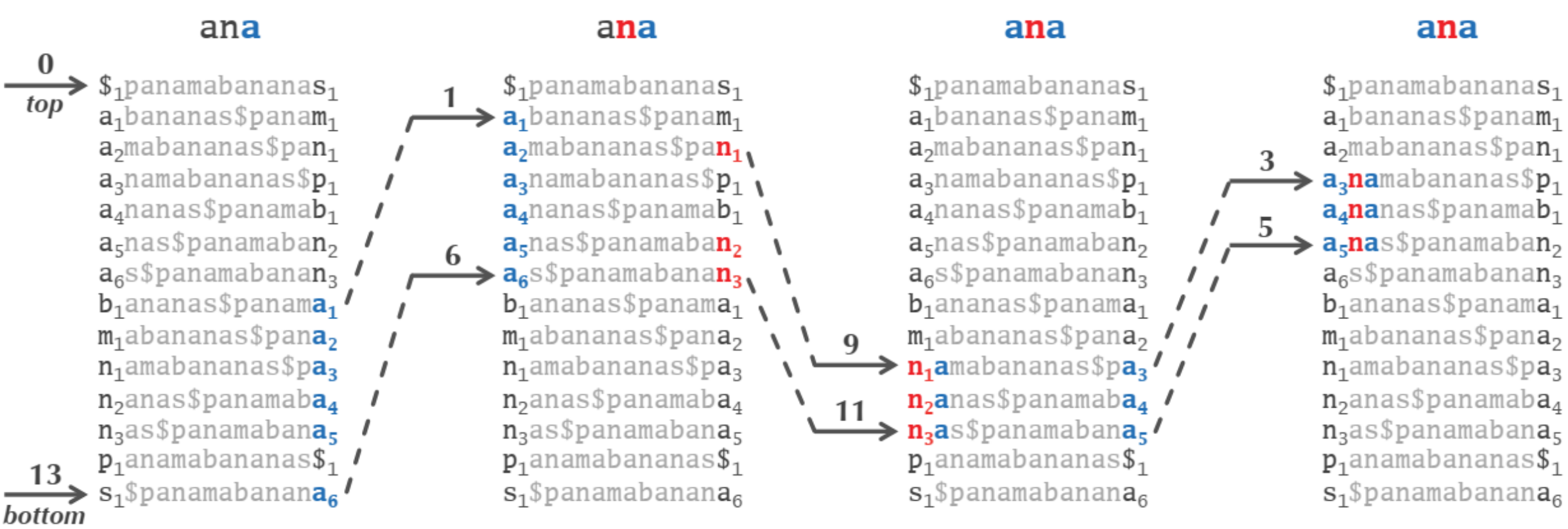
Ritrovare la nostra sequenza *ana*

1) Ordiniamo BWT(genoma) per ottenere la prima colonna
2) Selezioniamo l'**ultimo** carattere (*a*)
3) Teniamo solo quelle righe per cui la *a* è preceduta da una *n*
4) E poi andiamo a tenere solo quelle righe per le quali le *n* (ora in prima colonna), sono precedute da delle *a* (in ultima colonna)

$_1$	p	a	n	a	m	a	b	a	n	a	n	a s$_1$
a$_1$	b	a	n	a	n	a	s	$	p	a	n	a m$_1$
a$_2$	m	a	b	a	n	a	n	a	s	$	p	a n$_1$
a$_3$	n	a$_1$	m	a	b	a	n	a	n	a	s	$ p$_1$
a$_4$	n	a$_2$	n	a	s	$	p	a	n	a	m	a b$_1$
a$_5$	n	a$_3$	s	$	p	a	n	a	m	a	b	a n$_2$
a$_6$	s	$	p	a	n	a	m	a	b	a	n	a n$_3$
b$_1$	a	n	a	n	a	s	$	p	a	n	a	m a$_1$
m$_1$	a	b	a	n	a	n	a	s	$	p	a	n a$_2$
n$_1$	a	m	a	b	a	n	a	n	a	s	$	p a$_3$
n$_2$	a	n	a	s	$	p	a	n	a	m	a	b a$_4$
n$_3$	a	s	$	p	a	n	a	m	a	b	a	n a$_5$
p$_1$	a	n	a	m	a	b	a	n	a	n	a	s $_1$
s$_1$	$	p	a	n	a	m	a	b	a	n	a	n a$_6$

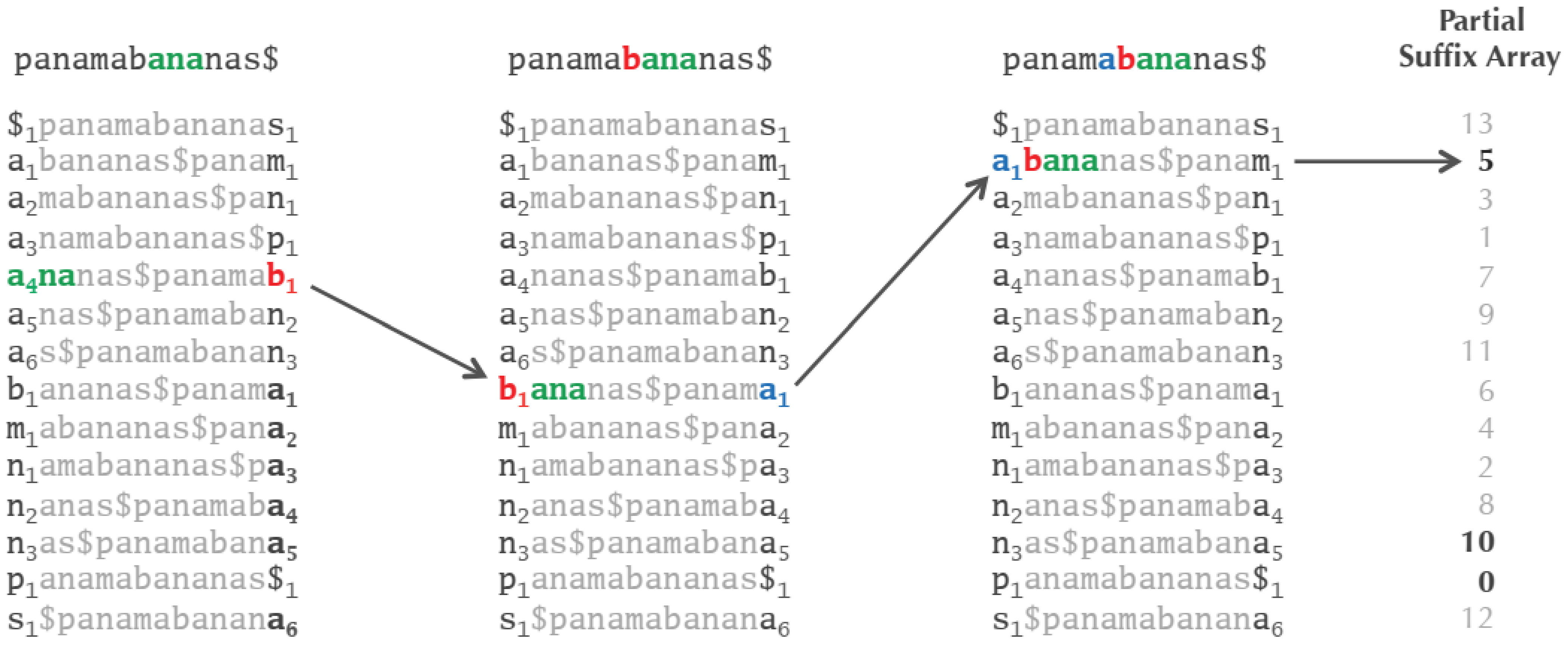


- Ricostruzione del percorso. Notate che per «risparmiare memoria» possiamo solo ricordare il primo e l'ultimo rango del carattere investigato (invece che tutte le posizioni)



OK, ma dove sono le mie sequenze? Come trovo le posizioni?

In pratica si combina questo approccio con degli «array di suffissi parziali»*



*Nell’esempio, una posizione ogni 5. In pratica, piu’ spesso, una ogni 100. Questo permette molta meno memoria di un array di suffissi (1%) ed evitare di dover percorrere il percorso fino all’inizio (posizione 0) per trovare la posizione della sequenza

Come permettere delle «mutazioni» differenti?

- Esempio: vogliamo trovare il pattern ***acttggct*** con al massimo un nucleotide diverso dalla referenza.
- Cosa fareste, usando la nozione che BWT è velocissimo a cercare brevi sequenze?

Come permettere delle «mutazioni» differenti?

- Esempio: vogliamo trovare il pattern ***acttggct*** con al massimo un nucleotide diverso dalla referenza.
- Cosa fareste, usando la nozione che BWT è velocissimo a cercare brevi sequenze?
- Potremmo cercare un **seed** – una sequenza minima che deve essere contenuta. Come disegnare questo **seed**?

Come permettere delle «mutazioni» differenti?

- Esempio: vogliamo trovare il pattern ***acttggct*** con al massimo un nucleotide diverso dalla referenza.
- Cosa fareste, usando la nozione che BWT è velocissimo a cercare brevi sequenze?
- Potremmo cercare un **seed** – una sequenza minima che deve essere contenuta. Come disegnare questo **seed**?
- Almeno metà della sequenza dovrebbe essere identica! Cerchiamo i due potenziali seed (le due metà)! Se trovate, estendiamo!

Pattern acttggct

Text ...ggcacactaggctcc...

Come permettere delle «mutazioni» differenti?

Due sequenze di lunghezza ***n*** che differiscono al massimo ***d*** mutazioni condividono necessariamente un k-mer di lunghezza k=n/(d+1)

Pattern acttaggctcgggataatcc

Text ...actaagtctcgggataagcc...

Come permettere delle «mutazioni» differenti?

Due sequenze di lunghezza ***n*** che differiscono al massimo ***d*** mutazioni condividono necessariamente un k-mer di lunghezza k=n/(d+1)

Pattern acttaggctcgggataatcc

Text ...actaagtctcgggataagcc...

acttaggctcgggataatccgga

Possiamo spezzare la nostra sequenza in frammenti di lunghezza k per ottenere i potenziali **seeds** da cercare!

Il seed è esteso fino ad eccedere il numero di mismatch

Sequenza ricercata con d=1 ggctcgggat

Genoma gtctcgggat

gggat seed

Il seed è esteso fino ad eccedere il numero di mismatch

Sequenza ricercata con d=1 ggctcgggat

Genoma gtctcgggat

gggat seed

cgggat estensione

Il seed è esteso fino ad eccedere il numero di mismatch

Sequenza ricercata con d=1 ggctcgggat

Genoma gtctcgggat

gggat seed

cgggat estensione

tcgggat

Il seed è esteso fino ad eccedere il numero di mismatch

Sequenza ricercata con d=1

Genoma

ggctcgggat

gtctcgggat

gggat seed

cgggat estensione

tcgggat

ctcgggat

Il seed è esteso fino ad eccedere il numero di mismatch

Sequenza ricercata con d=1 ggctcgggat

Genoma gtctcgggat

gggat seed

cgggat estensione

tcgggat

ctcgggat

tctcgggat Una differenza (mismatch) trovato

Il seed è esteso fino ad eccedere il numero di mismatch

Sequenza ricercata con d=1 ggctcgggat

Genoma gtctcgggat

gggat	seed
cgggat	estensione
tcgggat	
ctcgggat	
tctcgggat	Una differenza (mismatch) trovato
gtctcgggat	Nuova estensione finché: • Finisce la sequenza • Troviamo un altro mismatch

Esercizio

Ogni gruppo dia il proprio messaggio codificato ad una terza coppia.

Il primo gruppo (quello che ha compresso la sequenza usando la trasformata di Burrow-Wheeler) e conosce il messaggio, assegni una parola di 4 lettere (con 1 errore) da trovare al nuovo gruppo.

Il terzo gruppo, ritrovi questa sequenza.

Conclusioni della mappatura di sequenze corte

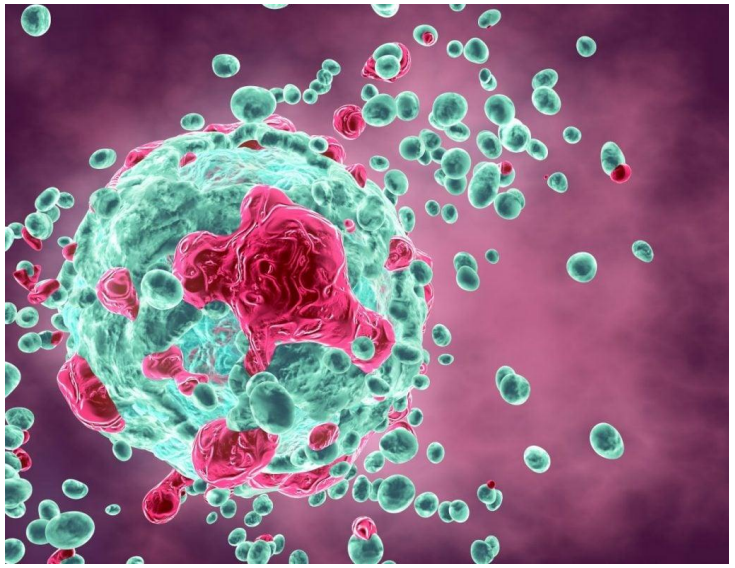
- Metodi basati sulla Trasformata di Burrow-Wheeler sono fino a 50 volte piu' veloci e occupano un decimo della memoria di metodi basati su «suffix arrays».

- I principali metodi di compressione dei dati genomici e di estrazione sequenze (bgzip e tabix) sono basati sulla trasformata di Burrow-Wheeler
  - Solo ***bwa*** nelle varie versioni ha oltre 100.000 citazioni!

Quasi ogni analisi di genomica necessita mappatura

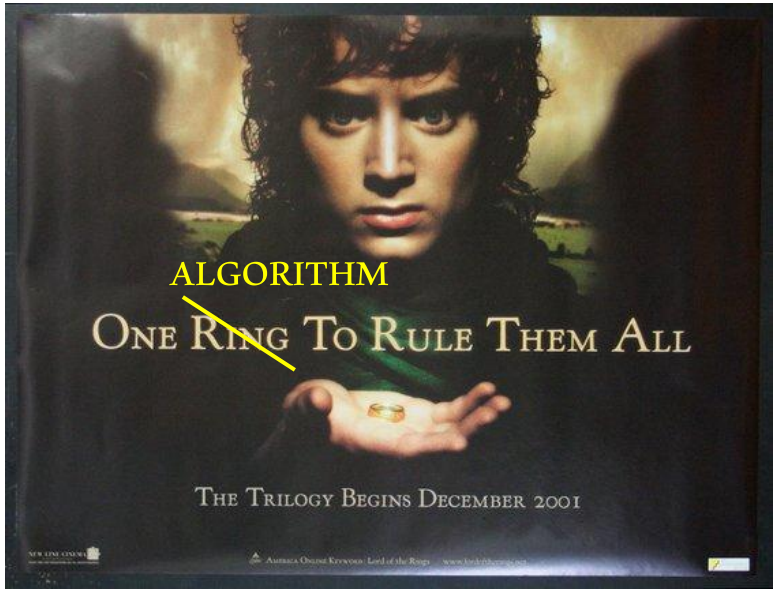
Espressione genica

Quali geni sono espressi in questa linea tumorale?



Assemblaggio di genomi

«Polishing» con short reads

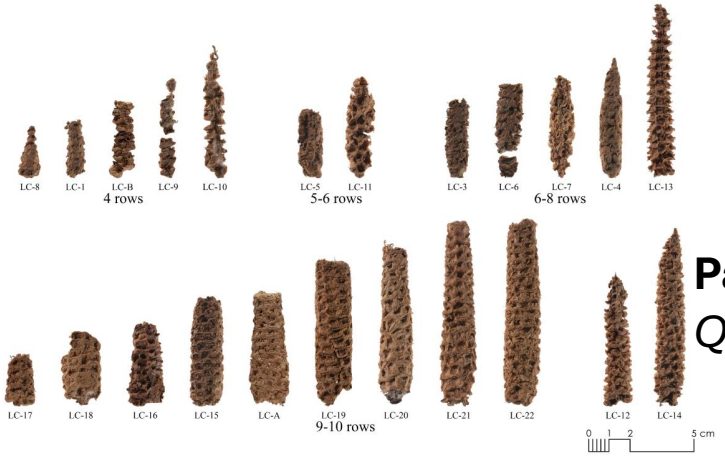


GWAS e genetica di popolazione

Quali varianti causano il diabete?

biobank uk

Improving the health of future generations

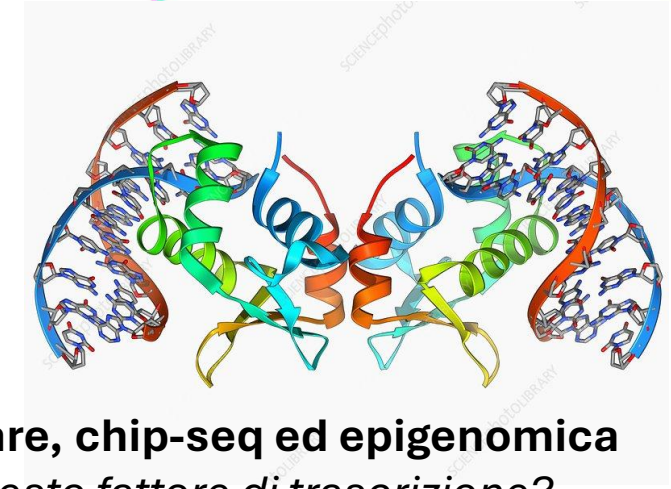


Paleogenomica e genetica forense

Questo osso è autentico ed antico?

Metagenomica

Quali specie sono presenti in questo campione?



Biologia molecolare, chip-seq ed epigenomica

Che geni regola questo fattore di trascrizione?

Quali regioni hanno questa modifica della cromatina?

Sommario di un mappatore di short reads

Per il genoma:

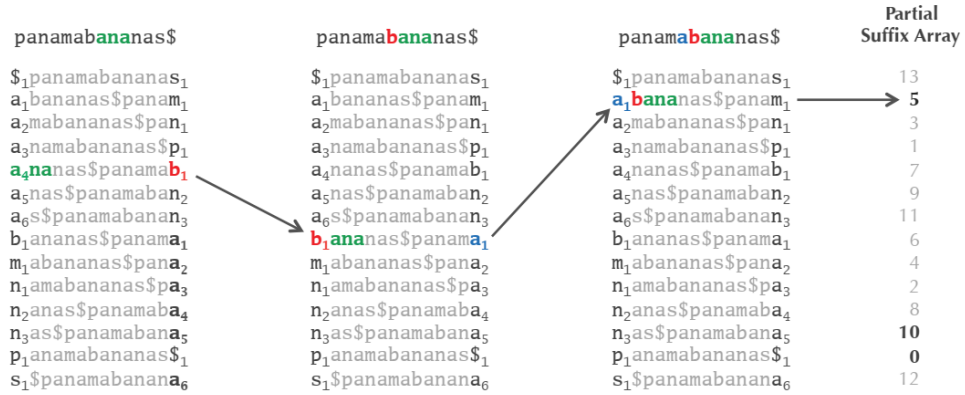
Compressione del genoma con BWT

Sommario di un mappatore di short reads

Per il genoma:

Compressione del genoma con BWT

Creazione degli indici (suffix array)

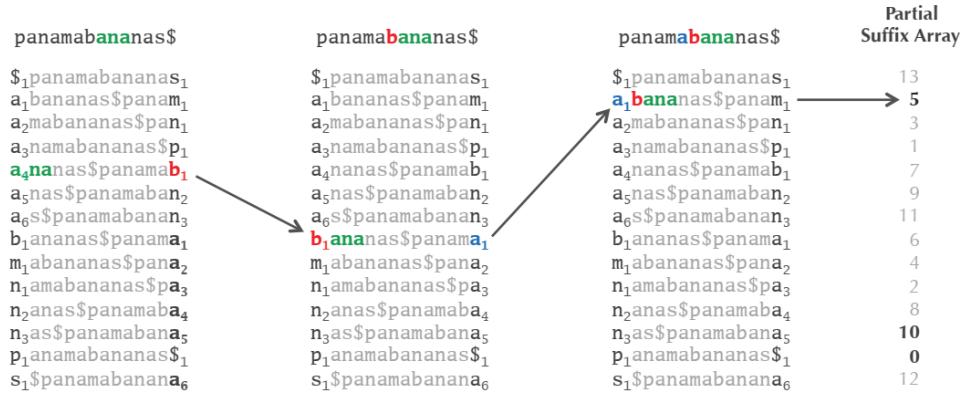


Sommario di un mappatore di short reads

Per il genoma:

Compressione del genoma con BWT

Creazione degli indici (suffix array)



Per ogni read:

identificazione di un seed

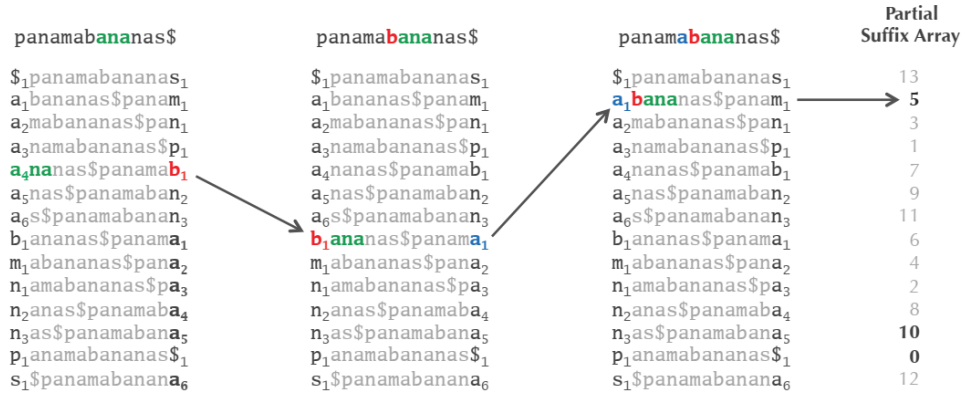
ACGTTGAC ← k=n/(d+1)

Sommario di un mappatore di short reads

Per il genoma:

Compressione del genoma con BWT

Creazione degli indici (suffix array)



Per ogni read:

identificazione di un seed

ACGTTGAC ← k=n/(d+1)

estensione in “avanti”* o a “ritroso”**
ignorando gli errori fino a **d+1** mismatches

→ TGACCGA

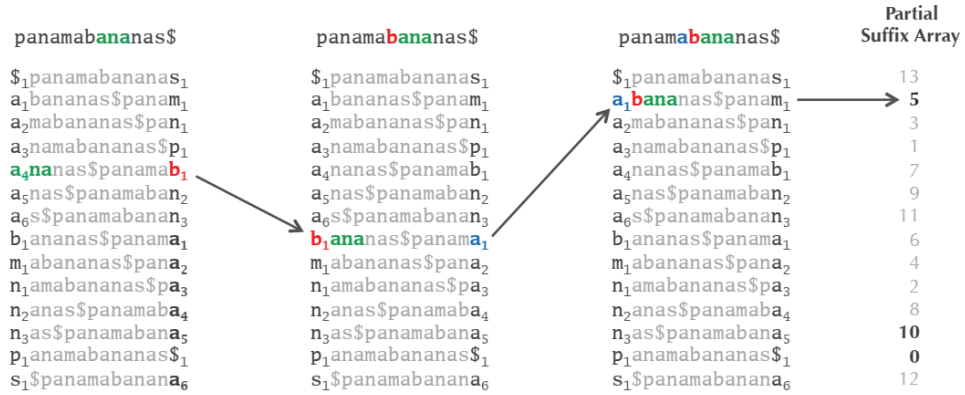
← TCTTTGAC

Sommario di un mappatore di short reads

Per il genoma:

Compressione del genoma con BWT

Creazione degli indici (suffix array)



Per ogni read:

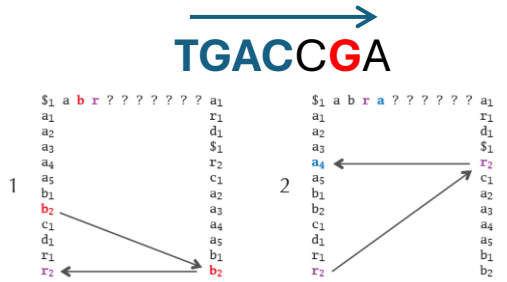
identificazione di un seed

ACGTTGAC ← k=n/(d+1)

estensione in “avanti”* o a “ritroso”**

ignorando gli errori fino a **d+1** mismatches

*come per la decompressione della BWT, «dalla prima all’ultima colonna»

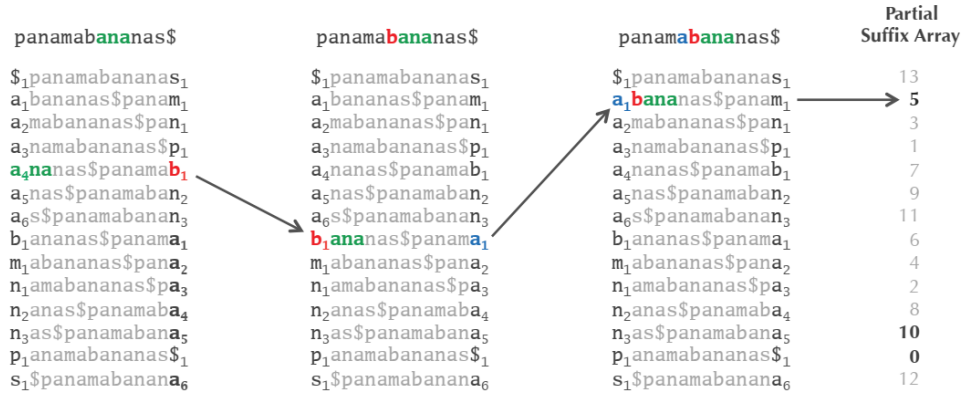


Sommario di un mappatore di short reads

Per il genoma:

Compressione del genoma con BWT

Creazione degli indici (suffix array)



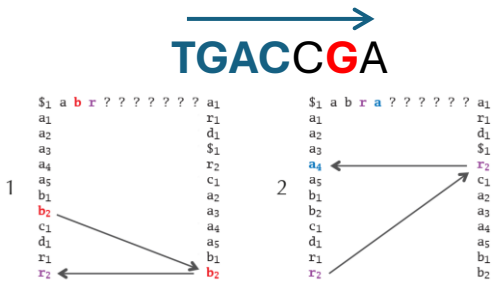
Per ogni read:

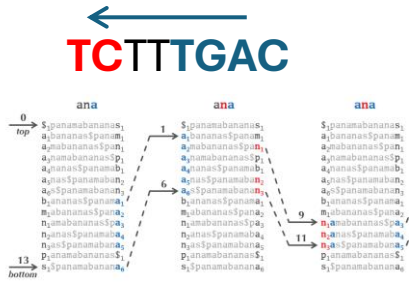
identificazione di un seed

ACGTTGAC ← k=n/(d+1)

estensione in “avanti”* o a “ritroso”**
ignorando gli errori fino a **d+1** mismatches

*come per la decompressione della BWT, «dalla prima all’ultima colonna»





** al contrario, «dall’ultima colonna alla prima»

Sommario di un mappatore di short reads

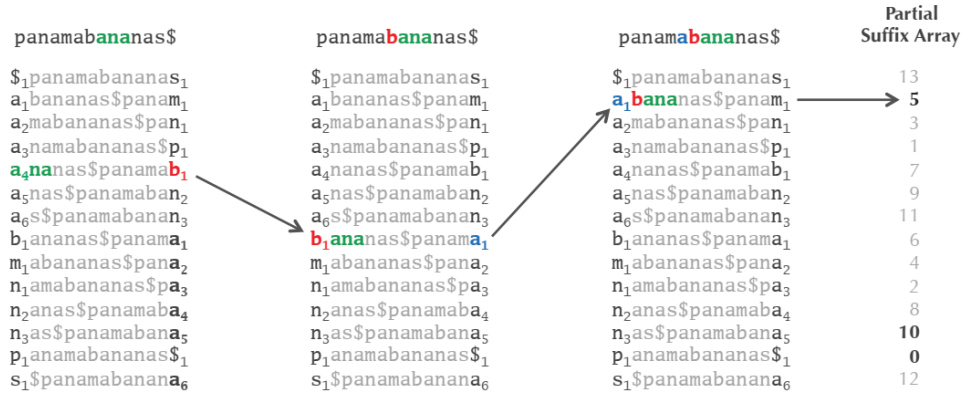
In BWA

```
bgzip genome.fasta
```

Per il genoma:

Compressione del genoma con BWT

Creazione degli indici (suffix array)



Per ogni read:

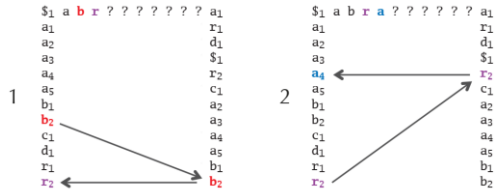
identificazione di un seed

ACGTTGAC ← k=n/(d+1)

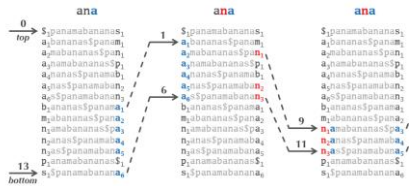
estensione in “avanti”* o a “ritroso”**
ignorando gli errori fino a **d+1** mismatches

*come per la decompressione della BWT, «dalla prima all’ultima colonna»

TGACCGA →



← TCTTTGAC



** al contrario, «dall’ultima colonna alla prima»

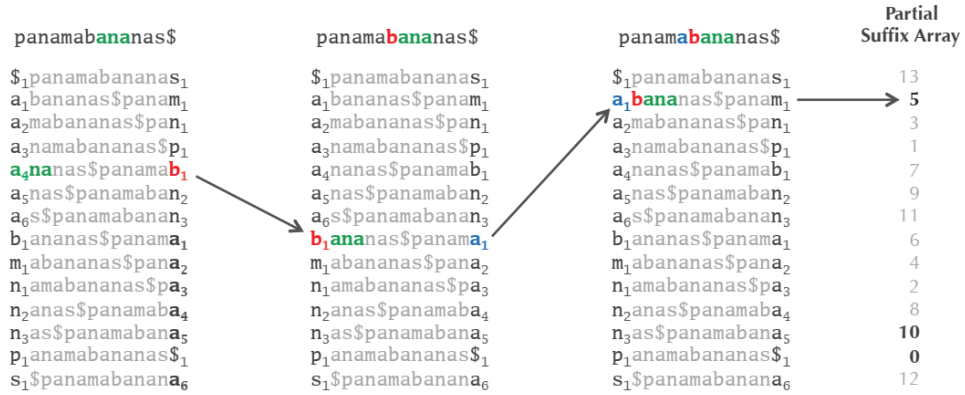
Sommario di un mappatore di short reads

In BWA

Per il genoma:

`bgzip genome.fasta` Compressione del genoma con BWT

`bwa index genome.fasta` Creazione degli indici (suffix array)



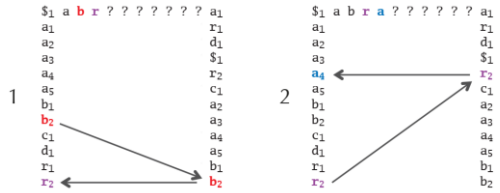
Per ogni read:

identificazione di un seed ACGTTGAC ← k=n/(d+1)

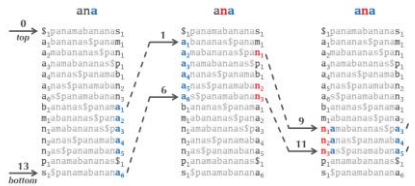
estensione in “avanti”* o a “ritroso”**
ignorando gli errori fino a **d+1** mismatches

*come per la decompressione della BWT, «dalla prima all’ultima colonna»

TGACCGA



TCTTTGAC



** al contrario, «dall’ultima colonna alla prima»

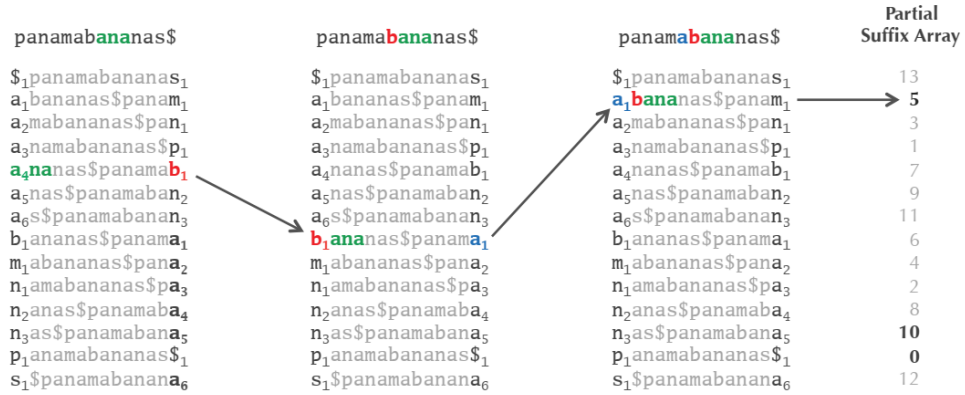
Sommario di un mappatore di short reads

In BWA

Per il genoma:

`bgzip genome.fasta` Compressione del genoma con BWT

`bwa index genome.fasta` Creazione degli indici (suffix array)



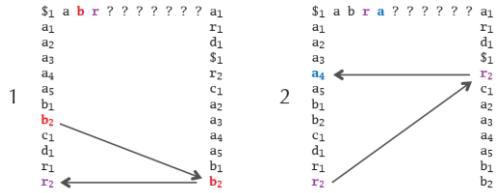
Per ogni read:

`bwa mem`

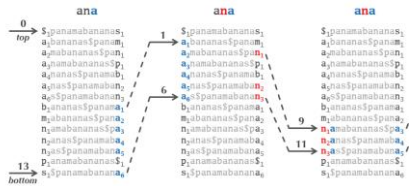
identificazione di un seed ACGTTGAC ← k=n/(d+1)

estensione in “avanti”* o a “ritroso”**
ignorando gli errori fino a **d+1** mismatches

TGACCGA →



← TCTTTGAC



*come per la decompressione della BWT, «dalla prima all’ultima colonna»

** al contrario, «dall’ultima colonna alla prima»

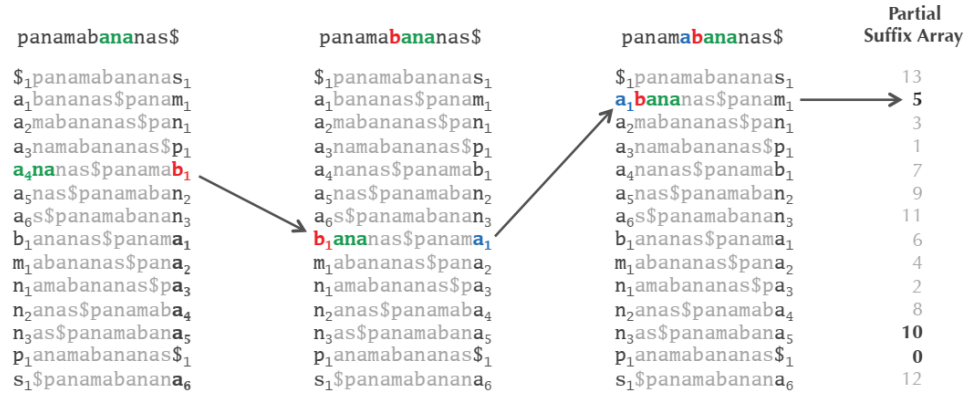
Sommario di un mappatore di short reads

In BWA

Per il genoma:

`bgzip genome.fasta` Compressione del genoma con BWT

`bwa index genome.fasta` Creazione degli indici (suffix array)



Per ogni read:

`bwa mem`

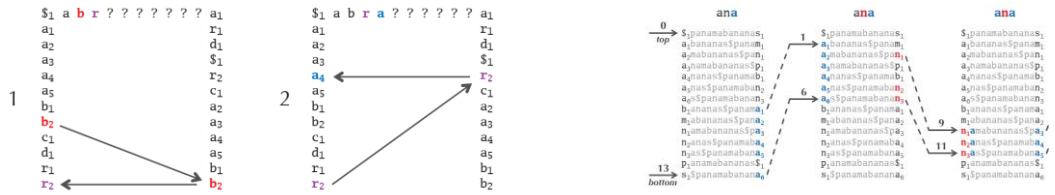
identificazione di un seed **ACGTTGAC** ← k=n/(d+1)

estensione in “avanti”* o a “ritroso”** ignorando gli errori fino a **d+1** mismatches

o se con indels algoritmi di dynamic programming per allineamento

TGACCGA

TCTTTGAC



*come per la decompressione della BWT, «dalla prima all’ultima colonna»

** al contrario, «dall’ultima colonna alla prima»

I principali software di mappatura per reads corte

bwa-aln	bwa-mem	bowtie	bowtie2
molto veloce	abbastanza veloce	Ultraveloce e poca memoria	Meno veloce di bowtie e piu' memoria
Reads ultra corte <50bp	Fino a qualche kb	Reads ultra corte <50bp	Reads corte

Il formato sam/bam

.sam e **.bam** sono lo stesso formato, con l'unica differenza che .bam è compresso (con Burrow-Wheeler!!), quindi non leggibile se non con tool specifici (`samtools`)

Formato tipico per tutti i file di mapping, sia da *long* che *short reads*.

«Se sono arrivato dove sono arrivato è grazie al fatto che conosco bene il formato bam»
Fritz Sedlazeck, autore di moltissimi strumenti bioinformatici

Il formato sam/bam

```
samtool view  bt2_ecoli1000.bam | head
```

```
r988    16      NC_000913.3     38063   24      35M     *       0       0       GCAGGAGCGTCCATCCATCGTTTGCCACTGAGTGA      45567778999:9;;<===>?@@@@AAAABCCCDE     AS:i:-8 XN:i:0  XM:i:2  XO:i:0  XG:i:0  NM:i:2  MD:Z:5T3A25     YT:Z:UU
r639    0       NC_000913.3     46755   24      35M     *       0       0       GCCATGATGACGCTGGCGCTGGCGGTTTTGGGGTA      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:-7 XN:i:0  XM:i:2  XO:i:0  XG:i:0  NM:i:2  MD:Z:33C0T0     YT:Z:UU
r408    16      NC_000913.3     53781   24      35M     *       0       0       ACAGTTTCACGCGGGCCTGTTCGTCAGTCATGATC      45567778999:9;;<===>?@@@@AAAABCCCDE     AS:i:-7 XN:i:0  XM:i:2  XO:i:0  XG:i:0  NM:i:2  MD:Z:0C9A24     YT:Z:UU
r375    0       NC_000913.3     56633   0       35M     *       0       0       TTCGCCACGTTGTTTCATCAGGTCCGCTTTAGTTT      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:-15        XN:i:0  XM:i:4  XO:i:0  XG:i:0  NM:i:4  MD:Z:31C0C0G0C0 YT:Z:UU
r33     16      NC_000913.3     64183   40      35M     *       0       0       NNNNNTTTGGTTTGCCGCATGATCTGATGACCACG      45567778999:9;;<===>?@@@@AAAABCCCDE     AS:i:-5 XN:i:0  XM:i:5  XO:i:0  XG:i:0  NM:i:5  MD:Z:0A0A0C0G0C30        YT:Z:UU
r364    0       NC_000913.3     65085   42      35M     *       0       0       CATTCGCAGATCGAACTGCACCACGTTCCAACCGA      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:35 YT:Z:UU
r694    16      NC_000913.3     68419   42      35M     *       0       0       TCGCCCATTGCTAATAGCGGCGATAAAGCTGTTCA      45567778999:9;;<===>?@@@@AAAABCCCDE     AS:i:-4 XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:12G22      YT:Z:UU
r980    0       NC_000913.3     72008   42      35M     *       0       0       CTGACGCCGTTGATTTCTGCGATCGGCGTGGTGGC      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:-4 XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:20C14      YT:Z:UU
r415    0       NC_000913.3     85596   42      35M     *       0       0       TACACATTTTTTCCGTCAAACAGTGAGGCTGGCCA      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:-4 XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:29A5       YT:Z:UU
r236    0       NC_000913.3     91139   42      35M     *       0       0       GAAGCTGCCACTCTGCCTGTTCATTTGCATTATTT      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:35 YT:Z:UU
```

Il formato sam/bam

```
samtool view  bt2_ecoli1000.bam | head
```

```
r988    16      NC_000913.3     38063   24      35M     *       0       0       GCAGGAGCGTCCATCCATCGTTTGCCACTGAGTGA      45567778999:9;;<===>?@@@@AAAABCCCDE     AS:i:-8 XN:i:0  XM:i:2  XO:i:0  XG:i:0  NM:i:2  MD:Z:5T3A25     YT:Z:UU
r639    0       NC_000913.3     46755   24      35M     *       0       0       GCCATGATGACGCTGGCGCTGGCGGTTTTGGGGTA      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:-7 XN:i:0  XM:i:2  XO:i:0  XG:i:0  NM:i:2  MD:Z:33C0T0     YT:Z:UU
r408    16      NC_000913.3     53781   24      35M     *       0       0       ACAGTTTCACGCGGGCCTGTTCGTCAGTCATGATC      45567778999:9;;<===>?@@@@AAAABCCCDE     AS:i:-7 XN:i:0  XM:i:2  XO:i:0  XG:i:0  NM:i:2  MD:Z:0C9A24     YT:Z:UU
r375    0       NC_000913.3     56633   0       35M     *       0       0       TTCGCCACGTTGTTTCATCAGGTCCGCTTTAGTTT      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:-15        XN:i:0  XM:i:4  XO:i:0  XG:i:0  NM:i:4  MD:Z:31C0C0G0C0 YT:Z:UU
r33     16      NC_000913.3     64183   40      35M     *       0       0       NNNNNTTTGGTTTGCCGCATGATCTGATGACCACG      45567778999:9;;<===>?@@@@AAAABCCCDE     AS:i:-5 XN:i:0  XM:i:5  XO:i:0  XG:i:0  NM:i:5  MD:Z:0A0A0C0G0C30        YT:Z:UU
r364    0       NC_000913.3     65085   42      35M     *       0       0       CATTCGCAGATCGAACTGCACCACGTTCCAACCGA      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:35 YT:Z:UU
r694    16      NC_000913.3     68419   42      35M     *       0       0       TCGCCCATTGCTAATAGCGGCGATAAAGCTGTTCA      45567778999:9;;<===>?@@@@AAAABCCCDE     AS:i:-4 XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:12G22      YT:Z:UU
r980    0       NC_000913.3     72008   42      35M     *       0       0       CTGACGCCGTTGATTTCTGCGATCGGCGTGGTGGC      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:-4 XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:20C14      YT:Z:UU
r415    0       NC_000913.3     85596   42      35M     *       0       0       TACACATTTTTTCCGTCAAACAGTGAGGCTGGCCA      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:-4 XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:29A5       YT:Z:UU
r236    0       NC_000913.3     91139   42      35M     *       0       0       GAAGGTGCCACTCTGCCTGTTCATTTGCATTATTT      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:35 YT:Z:UU
```

Colonna	Campo	Descrizione
1	QNAME	Nome della read
2	FLAG Binaria	Codifica per proprietà dell'allineamento
3	RNAME	Nome della sequenza (genoma o nome del cromosoma) di riferimento
4	POS	Posizione sulla sequenza di riferimento
5	MAPQ	Qualità della mappatura. 0=qualità incerta; 60 qualità ottima
6	CIGAR	Stringa CIGAR.
7	RNEXT	Per le reads paired-end, la sequenza di riferimento a cui è mappata la paired reads. Può essere *CHR1*, = (stesso cromosoma del mate), * (mate non allineato o assente)
8	PNEXT	Posizione del mate se paired-end sequencing
9	TLEN	Lunghezza del frammento inserito. Negativo se il mate o la direzione è inversa
10	SEQ	Sequenza
11	QUAL	Qualità della sequenza in PHRED score,come nel fastq

Il formato sam/bam

```
samtool view  bt2_ecoli1000.bam | head
```

```
r988    16      NC_000913.3     38063   24      35M     *       0       0       GCAGGAGCGTCCATCCATCGTTTGCCACTGAGTGA      45567778999:9;;<===>?@@@@AAAABCCCDE     AS:i:-8 XN:i:0  XM:i:2  XO:i:0  XG:i:0  NM:i:2  MD:Z:5T3A25     YT:Z:UU
r639    0       NC_000913.3     46755   24      35M     *       0       0       GCCATGATGACGCTGGCGCTGGCGGTTTTGGGGTA      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:-7 XN:i:0  XM:i:2  XO:i:0  XG:i:0  NM:i:2  MD:Z:33C0T0     YT:Z:UU
r408    16      NC_000913.3     53781   24      35M     *       0       0       ACAGTTTCACGCGGGCCTGTTCGTCAGTCATGATC      45567778999:9;;<===>?@@@@AAAABCCCDE     AS:i:-7 XN:i:0  XM:i:2  XO:i:0  XG:i:0  NM:i:2  MD:Z:0C9A24     YT:Z:UU
r375    0       NC_000913.3     56633   0       35M     *       0       0       TTCGCCACGTTGTTTCATCAGGTCCGCTTTAGTTT      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:-15        XN:i:0  XM:i:4  XO:i:0  XG:i:0  NM:i:4  MD:Z:31C0C0G0C0 YT:Z:UU
r33     16      NC_000913.3     64183   40      35M     *       0       0       NNNNNTTTGGTTTGCCGCATGATCTGATGACCACG      45567778999:9;;<===>?@@@@AAAABCCCDE     AS:i:-5 XN:i:0  XM:i:5  XO:i:0  XG:i:0  NM:i:5  MD:Z:0A0A0C0G0C30        YT:Z:UU
r364    0       NC_000913.3     65085   42      35M     *       0       0       CATTCGCAGATCGAACTGCACCACGTTCCAACCGA      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:35 YT:Z:UU
r694    16      NC_000913.3     68419   42      35M     *       0       0       TCGCCCATTGCTAATAGCGGCGATAAAGCTGTTCA      45567778999:9;;<===>?@@@@AAAABCCCDE     AS:i:-4 XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:12G22      YT:Z:UU
r980    0       NC_000913.3     72008   42      35M     *       0       0       CTGACGCCGTTGATTTCTGCGATCGGCGTGGTGGC      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:-4 XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:20C14      YT:Z:UU
r415    0       NC_000913.3     85596   42      35M     *       0       0       TACACATTTTTTCCGTCAAACAGTGAGGCTGGCCA      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:-4 XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:29A5       YT:Z:UU
r236    0       NC_000913.3     91139   42      35M     *       0       0       GAAGCTGCCACTCTGCCTGTTCATTTGCATTATTT      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:35 YT:Z:UU
```

Colonna	Campo	Descrizione
1	QNAME	Nome della read
2	FLAG Binaria	Codifica per proprietà dell'allineamento
3	RNAME	Nome della sequenza (genoma o nome del cromosoma) di riferimento
4	POS	Posizione sulla sequenza di riferimento
5	MAPQ	Qualità della mappatura. 0=qualità incerta; 60 qualità ottima
6	CIGAR	Stringa CIGAR.

Indica il numero di:
MATCH/MISMATCH: basi allineate, esattamente o con mismatch) con M
I o **D**: Inserzioni o delezioni
S e **H**: basi tagliate, presenti (Soft-clipping) o rimosse dalla sequenza (Hard-clipping)
Esempi:
50M= 50 basi allineate

Il formato sam/bam

```
samtool view  bt2_ecoli1000.bam | head
```

```
r988    16      NC_000913.3     38063   24      35M     *       0       0       GCAGGAGCGTCCATCCATCGTTTGCCACTGAGTGA      45567778999:9;;<===>?@@@@AAAABCCCDE     AS:i:-8 XN:i:0  XM:i:2  XO:i:0  XG:i:0  NM:i:2  MD:Z:5T3A25     YT:Z:UU
r639    0       NC_000913.3     46755   24      35M     *       0       0       GCCATGATGACGCTGGCGCTGGCGGTTTTGGGGTA      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:-7 XN:i:0  XM:i:2  XO:i:0  XG:i:0  NM:i:2  MD:Z:33C0T0     YT:Z:UU
r408    16      NC_000913.3     53781   24      35M     *       0       0       ACAGTTTCACGCGGGGCCTGTTCGTCAGTCATGATC      45567778999:9;;<===>?@@@@AAAABCCCDE     AS:i:-7 XN:i:0  XM:i:2  XO:i:0  XG:i:0  NM:i:2  MD:Z:0C9A24     YT:Z:UU
r375    0       NC_000913.3     56633   0       35M     *       0       0       TTCGCCACGTTGTTTCATCAGGTCCGCTTTAGTTT      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:-15        XN:i:0  XM:i:4  XO:i:0  XG:i:0  NM:i:4  MD:Z:31C0C0G0C0 YT:Z:UU
r33     16      NC_000913.3     64183   40      35M     *       0       0       NNNNNTTTGGTTTGCCGCATGATCTGATGACCACG      45567778999:9;;<===>?@@@@AAAABCCCDE     AS:i:-5 XN:i:0  XM:i:5  XO:i:0  XG:i:0  NM:i:5  MD:Z:0A0A0C0G0C30        YT:Z:UU
r364    0       NC_000913.3     65085   42      35M     *       0       0       CATTCGCAGATCGAACTGCACCACGTTCCAACCGA      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:35 YT:Z:UU
r694    16      NC_000913.3     68419   42      35M     *       0       0       TCGCCCATTGCTAATAGCGGCGATAAAGCTGTTCA      45567778999:9;;<===>?@@@@AAAABCCCDE     AS:i:-4 XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:12G22      YT:Z:UU
r980    0       NC_000913.3     72008   42      35M     *       0       0       CTGACGCCGTTGATTTCTGCGATCGGCGTGGTGGC      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:-4 XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:20C14      YT:Z:UU
r415    0       NC_000913.3     85596   42      35M     *       0       0       TACACATTTTTTCCGTCAAACAGTGAGGCTGGCCA      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:-4 XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:29A5       YT:Z:UU
r236    0       NC_000913.3     91139   42      35M     *       0       0       GAAGCTGCCACTCTGCCTGTTCATTTGCATTATTT      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:35 YT:Z:UU
```

Colonna	Campo	Descrizione
1	QNAME	Nome della read
2	FLAG Binaria	Codifica per proprietà dell'allineamento
3	RNAME	Nome della sequenza (genoma o nome del cromosoma) di riferimento
4	POS	Posizione sulla sequenza di riferimento
5	MAPQ	Qualità della mappatura. 0=qualità incerta; 60 qualità ottima
6	CIGAR	Stringa CIGAR.

Indica il numero di:
MATCH/MISMATCH: basi allineate, esattamente o con mismatch) con M
I o **D**: Inserzioni o delezioni
S e **H**: basi tagliate, presenti (Soft-clipping) o rimosse dalla sequenza (Hard-clipping)
Esempi:
25M2I10M1D40M=

```
Read:    ATCG...TCA**GG**TCA...TCA**?**TCA...
           ↑       ↑↑      ↑     ↑
         25M      2I      10M   1D (base mancante)
Riferimento: ATCG...TCA**--**TCA...TCA**G**TCA...
```

Il formato sam/bam

```
samtool view  bt2_ecoli1000.bam | head
```

```
r988    16      NC_000913.3     38063   24      35M     *       0       0       GCAGGAGCGTCCATCCATCGTTTGCCACTGAGTGA      45567778999:9;;<===>?@@@@AAAABCCCDE     AS:i:-8 XN:i:0  XM:i:2  XO:i:0  XG:i:0  NM:i:2  MD:Z:5T3A25     YT:Z:UU
r639    0       NC_000913.3     46755   24      35M     *       0       0       GCCATGATGACGCTGGCGCTGGCGGTTTTGGGGTA      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:-7 XN:i:0  XM:i:2  XO:i:0  XG:i:0  NM:i:2  MD:Z:33C0T0     YT:Z:UU
r408    16      NC_000913.3     53781   24      35M     *       0       0       ACAGTTTCACGCGGGCCTGTTCGTCAGTCATGATC      45567778999:9;;<===>?@@@@AAAABCCCDE     AS:i:-7 XN:i:0  XM:i:2  XO:i:0  XG:i:0  NM:i:2  MD:Z:0C9A24     YT:Z:UU
r375    0       NC_000913.3     56633   0       35M     *       0       0       TTCGCCACGTTGTTTCATCAGGTCCGCTTTAGTTT      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:-15        XN:i:0  XM:i:4  XO:i:0  XG:i:0  NM:i:4  MD:Z:31C0C0G0C0 YT:Z:UU
r33     16      NC_000913.3     64183   40      35M     *       0       0       NNNNNTTTGGTTTGCCGCATGATCTGATGACCACG      45567778999:9;;<===>?@@@@AAAABCCCDE     AS:i:-5 XN:i:0  XM:i:5  XO:i:0  XG:i:0  NM:i:5  MD:Z:0A0A0C0G0C30        YT:Z:UU
r364    0       NC_000913.3     65085   42      35M     *       0       0       CATTCGCAGATCGAACTGCACCACGTTCCAACCGA      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:35 YT:Z:UU
r694    16      NC_000913.3     68419   42      35M     *       0       0       TCGCCCATTGCTAATAGCGGCGATAAAGCTGTTCA      45567778999:9;;<===>?@@@@AAAABCCCDE     AS:i:-4 XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:12G22      YT:Z:UU
r980    0       NC_000913.3     72008   42      35M     *       0       0       CTGACGCCGTTGATTTCTGCGATCGGCGTGGTGGC      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:-4 XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:20C14      YT:Z:UU
r415    0       NC_000913.3     85596   42      35M     *       0       0       TACACATTTTTTCCGTCAAACAGTGAGGCTGGCCA      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:-4 XN:i:0  XM:i:1  XO:i:0  XG:i:0  NM:i:1  MD:Z:29A5       YT:Z:UU
r236    0       NC_000913.3     91139   42      35M     *       0       0       GAAGCTGCCACTCTGCCTGTTCATTTGCATTATTT      EDCCCBAAAA@@@@?>===<;;9:99987776554     AS:i:0  XN:i:0  XM:i:0  XO:i:0  XG:i:0  NM:i:0  MD:Z:35 YT:Z:UU
```

Colonna	Campo	Descrizione
1	QNAME	Nome della read
2	**FLAG Binaria**	**Codifica per proprietà dell'allineamento**

Utilizza un formato in bits, dove:

Bit	Description
0x1	template having multiple segments in sequencing
0x2	each segment properly aligned according to the aligner
0x4	segment unmapped
0x8	next segment in the template unmapped
0x10	SEQ being reverse complemented
0x20	SEQ of the next segment in the template being reversed
0x40	the first segment in the template
0x80	the last segment in the template
0x100	secondary alignment
0x200	not passing quality controls
0x400	PCR or optical duplicate
0x800	supplementary alignment

awk

- Comando (insieme di comandi) potentissimo
- Lavora linea dopo linea
- Pensato per tabelle
- Comando tipo:
- Estrae la colonna numero 5

```
cat FILE.TSV | awk  '{print $5}'
```

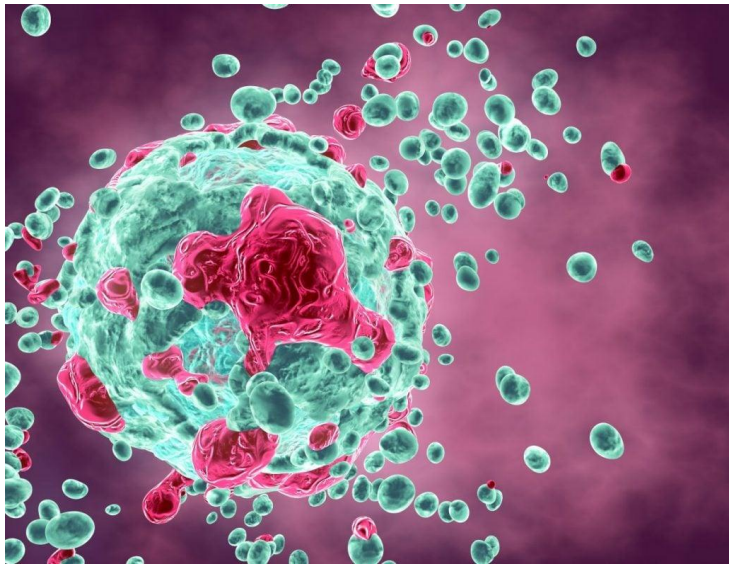
- Estrae le colonna numero 5, numero 6, e i primi 3 caratteri di 7

```
cat FILE.TSV | awk  '{print $5,$6,substr($7,1,3)}'
```

Quasi ogni analisi di genomica necessita mappatura

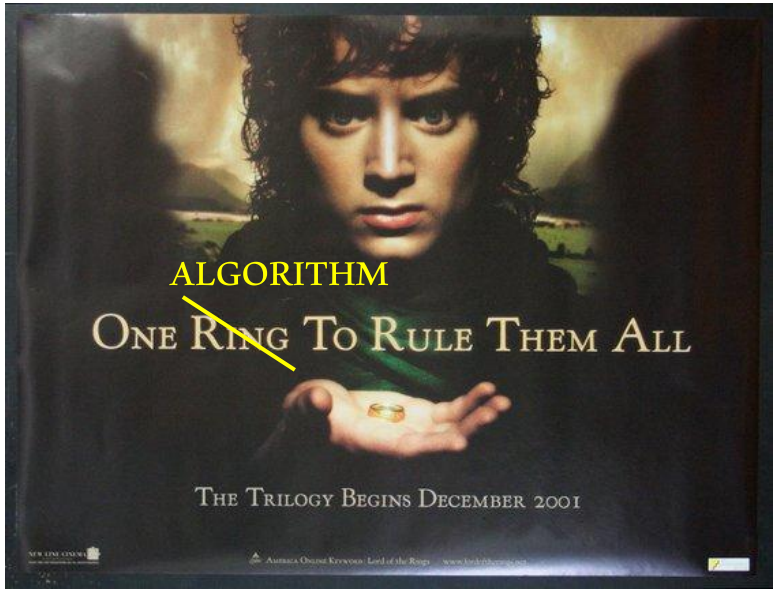
Espressione genica

Quali geni sono espressi in questa linea tumorale?



Assemblaggio di genomi

«Polishing» con short reads

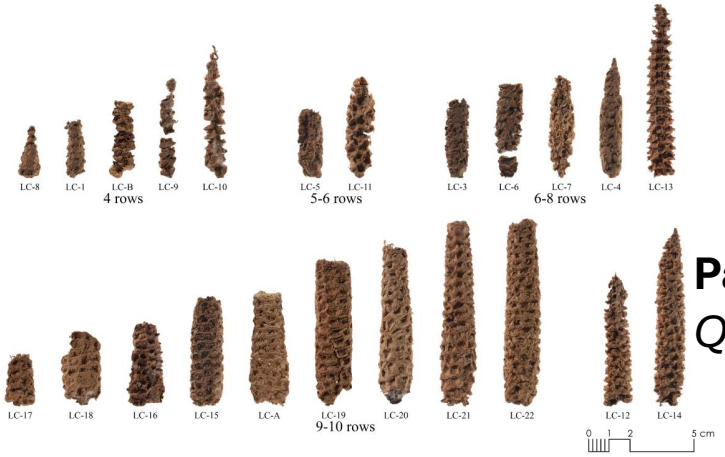


GWAS e genetica di popolazione

Quali varianti causano il diabete?

biobank uk

Improving the health of future generations

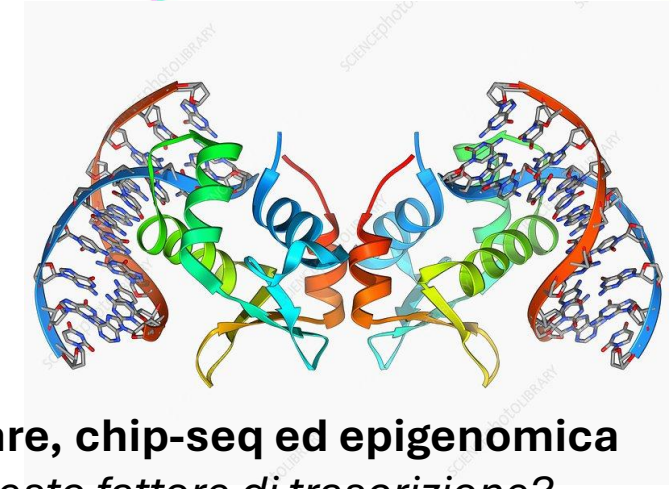


Paleogenomica e genetica forense

Questo osso è autentico ed antico?

Metagenomica

Quali specie sono presenti in questo campione?



Biologia molecolare, chip-seq ed epigenomica

Che geni regola questo fattore di trascrizione?

Quali regioni hanno questa modifica della cromatina?