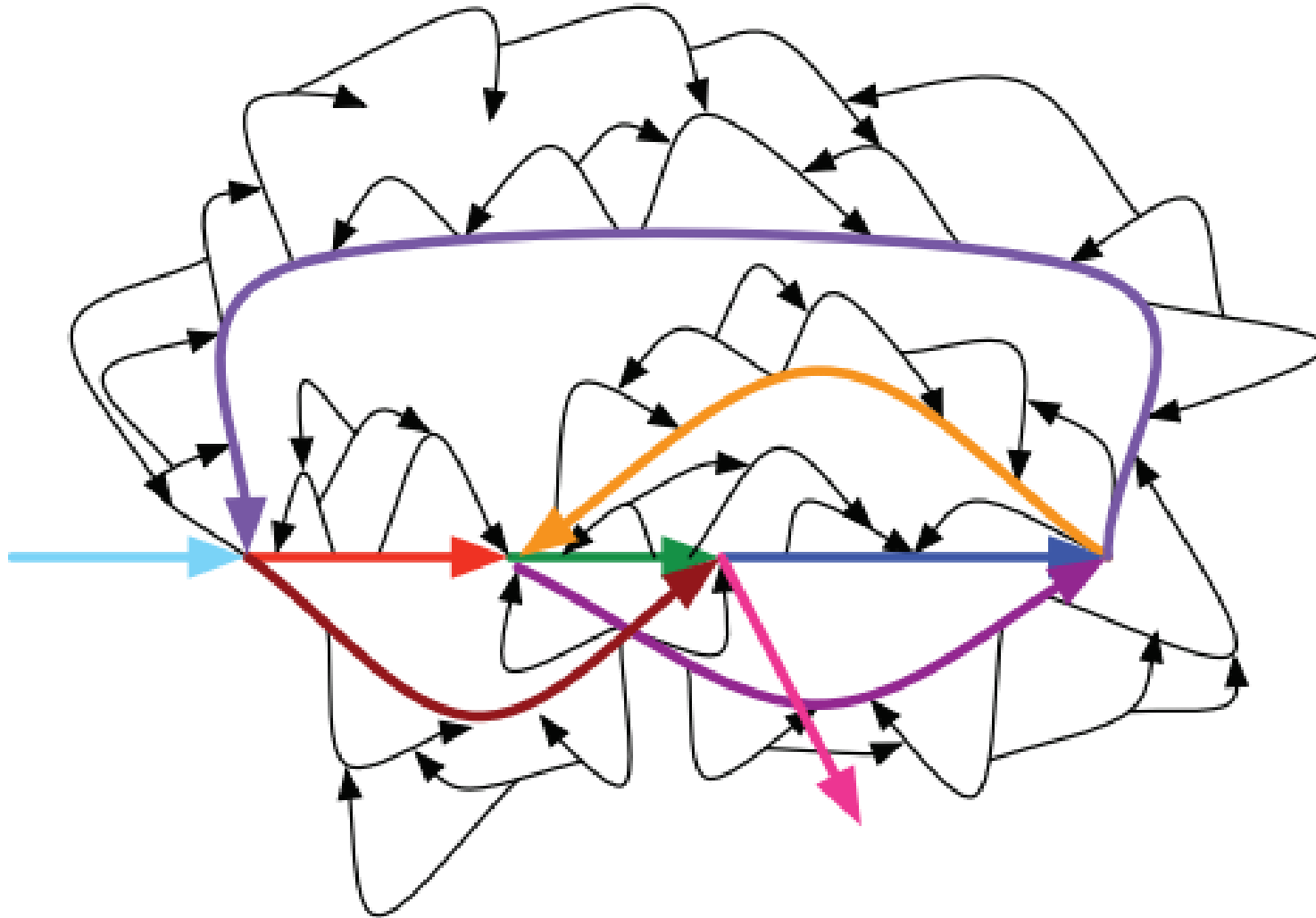


Come assemblare un genoma: difficoltà, pratica e valutazione di un genoma assemblato



Cosa abbiamo la volta scorsa

- Perché è difficile assemblare
- Algoritmi per assemblare:
 - Overlap-graphs
 - De Bruijn graphs
- Concetti generali delle scienze computazionali (“Teoria dei grafi”, algoritmi polinomiali ed intrattabili)

Questa lezione

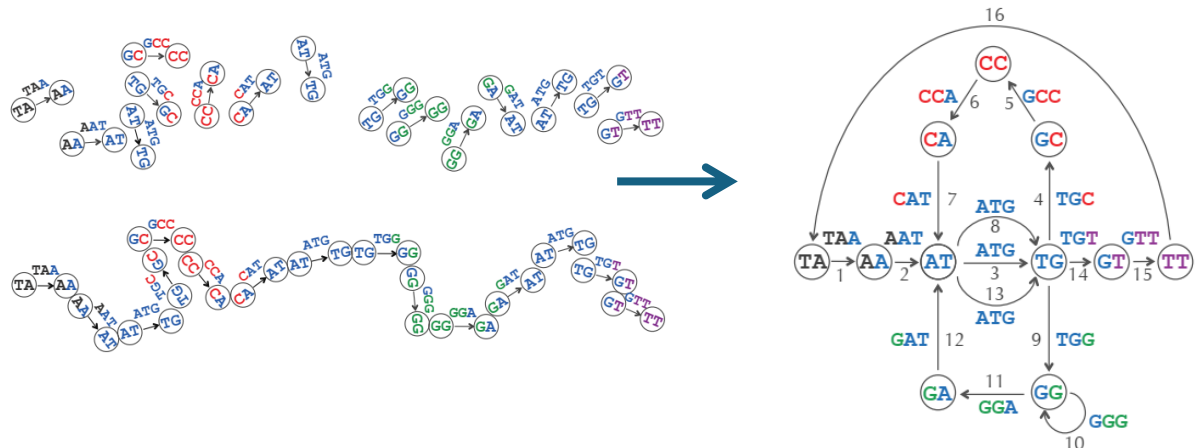
- Applicazioni specifiche a tipologie di dati di sequenziamento e problemi dei “genomi veri”
 - Il problema delle ripetizioni
 - Scaffolding
 - Assemblare con sequenze lunghe e corte
 - Valutazione degli assemblaggi
- Pratica di assemblaggio di genomi con strumenti bioinformatici

Due approcci per assemblare un genoma

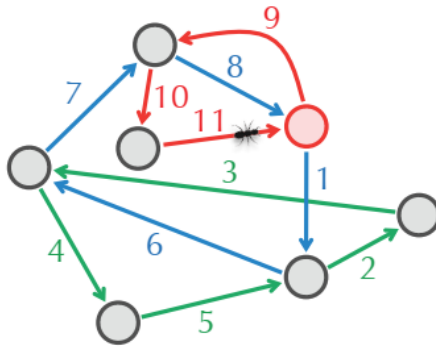
Esplorazione degli **Overlap-graph** per «forza bruta» ed usando al piu delle «euristiche»



Costruzione del **De Bruijn Graph**



Ricerca del «ciclo Euleriano»*



*con l'algoritmo di Leo la formica



Esercizio di assemblaggio manuale

Ricostruite la sequenza originale usando l'approccio degli ***Overlap-graphs*** e dei ***De Bruijn graphs***, partendo da questi 4-mer.

CGTA

ACGT

TAGG

GTAT

GCGA

GGTA

ATGC

GTAG

CGTG

AGGT

CGAT

ATGC

GATT

CACG

ATTA

ACGT

TTAT

TGCA

GCAC

TATG

Abbiamo i nostri algoritmi per assemblare: ma quali sono le difficoltà pratiche?



Abbiamo i nostri algoritmi per assemblare: ma quali sono le difficoltà pratiche?

- Errori di sequenziamento
- Ripetizioni Nel genoma
- Ripetizioni Del genoma (diploidia e poliploidia)

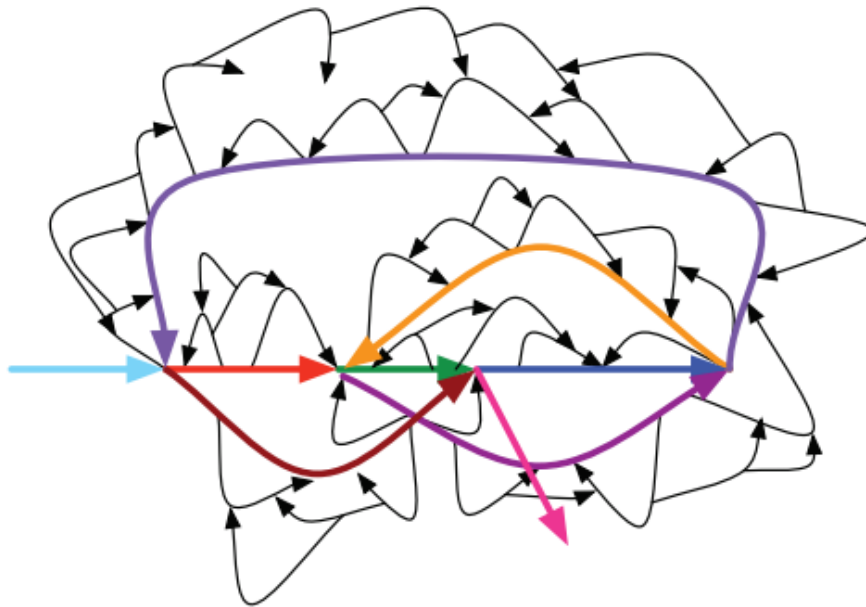


Problema 1: abbiamo errori nelle sequenze (o bolle)



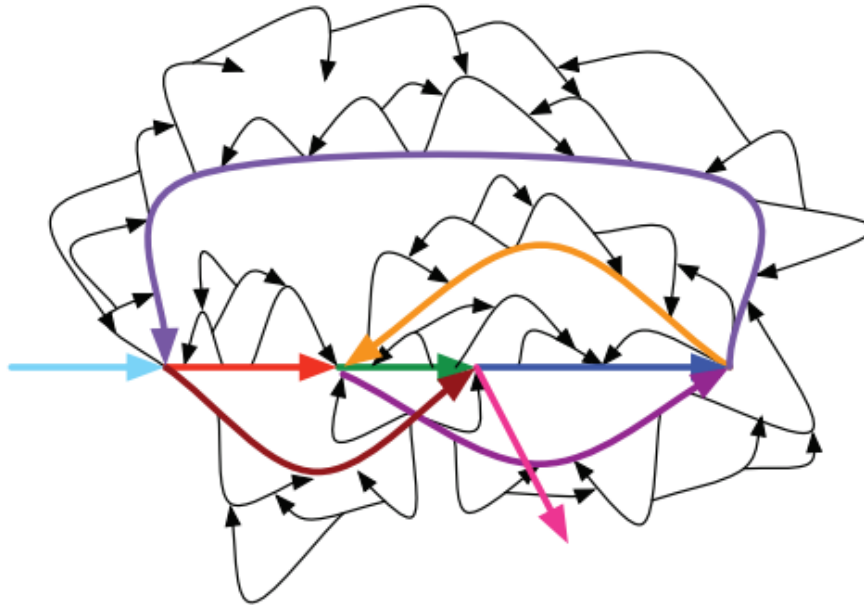
Una “bolla” causata da un errore di sequenziamento (una T letta come C)

Problema 1: abbiamo errori nelle sequenze (o bolle)



In dati reali ci sono molti errori di sequenziamento (quindi molte bolle)!

Problema 1: abbiamo errori nelle sequenze (o bolle)



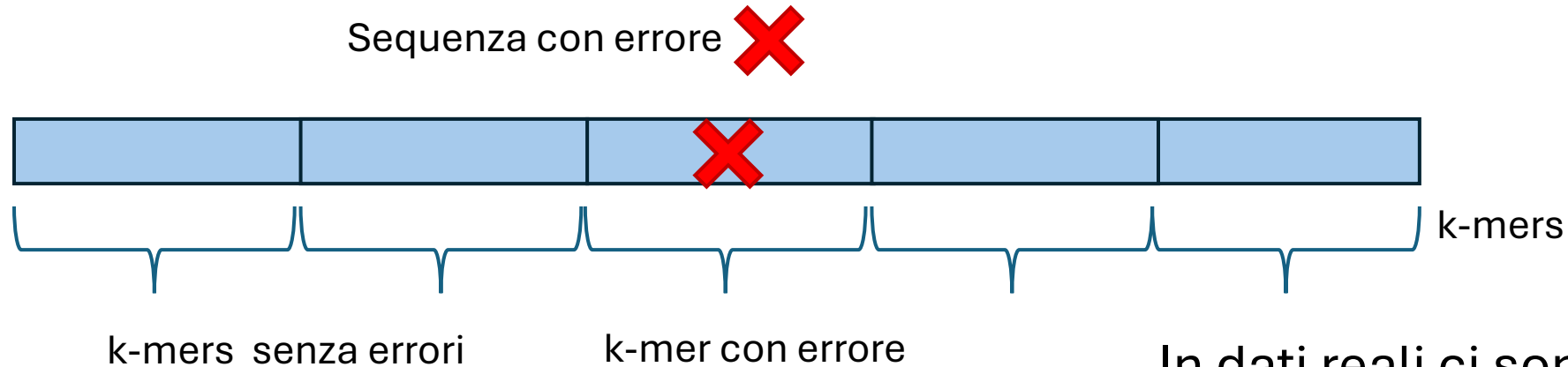
In dati reali ci sono molti errori di sequenziamento (quindi molte bolle)!

Assemblatori moderni correggono queste bolle sulla base di alcune “euristiche”:

- rimuovere i **kmers rari*** prima della costruzione del grafo
- costruire il grafo e poi rimuovere i percorsi che creano bolle e sono supportati da **kmers rari**

*gli errori di sequenziamento avvengono perlopiu' casualmente, quindi è abbastanza improbabile che molti errori avvengano nella stessa posizione. Pertanto generalmente gli errori originano kmer unici o molto rari

Problema 1: abbiamo errori nelle sequenze (o bolle)

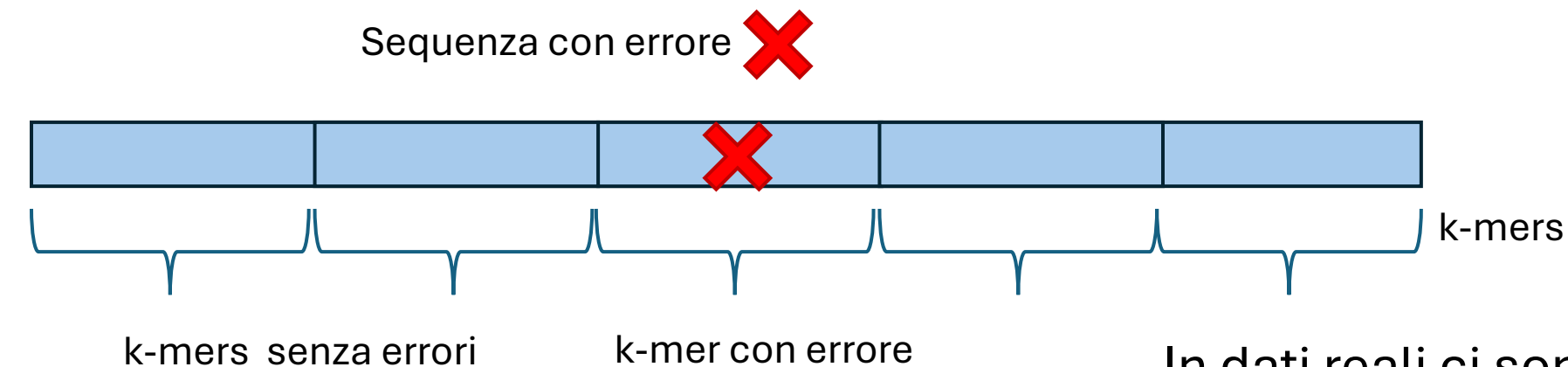


In dati reali ci sono molti errori di sequenziamento (quindi molte bolle)!

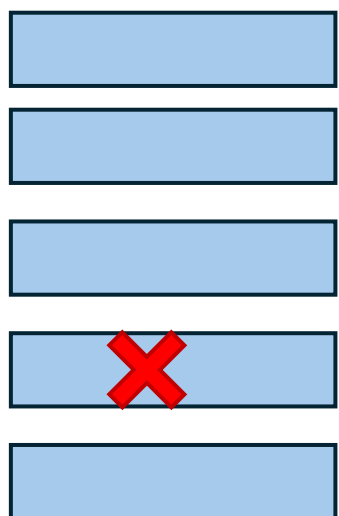
Assemblatori moderni correggono queste bolle sulla base di alcune “euristiche”:

- rimuovere i **kmers rari*** prima della costruzione del grafo
- costruire il grafo e poi rimuovere i percorsi che creano bolle e sono supportati da **kmers rari**
- **spezzare le reads in kmers piu' piccoli (soprattutto per de Bruijn graphs)**

Problema 1: abbiamo errori nelle sequenze (o bolle)



In dati reali ci sono molti errori di sequenziamento (quindi molte bolle)!

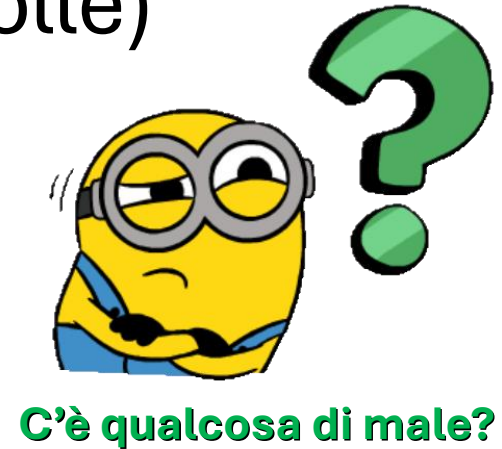
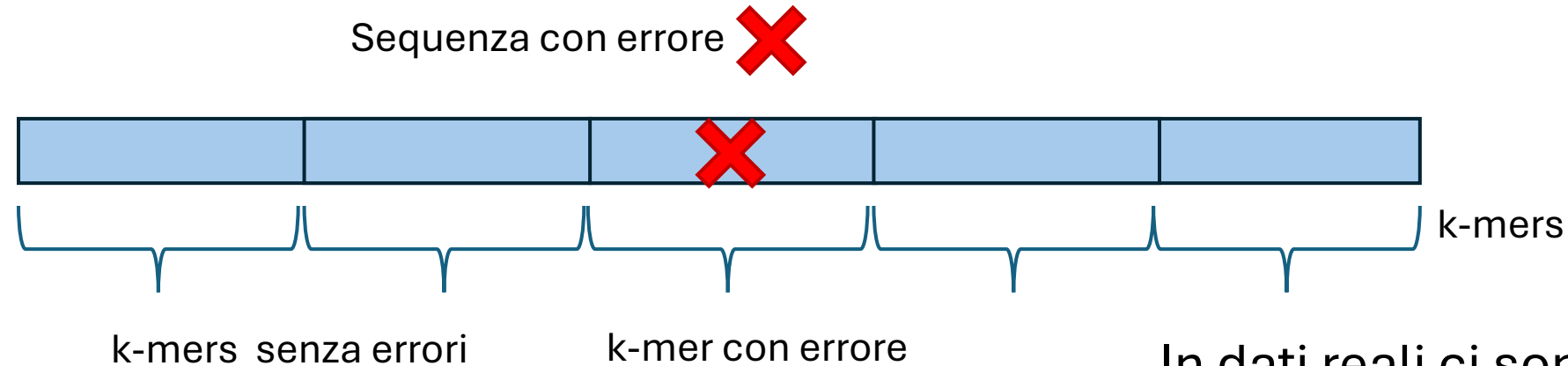


La maggior parte dei k-mers non hanno errori e così posso utilizzare quasi tutta la sequenza senza preoccuparmi dell'errore

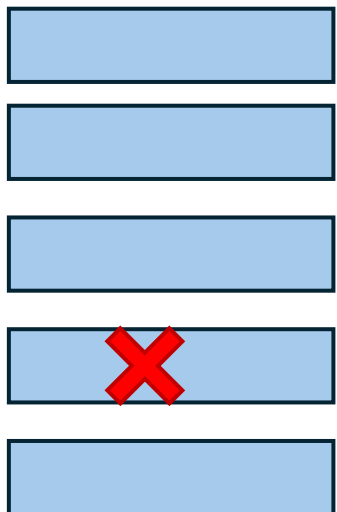
Assemblatori moderni correggono queste bolle sulla base di alcune “euristiche”:

- rimuovere i **kmers rari*** prima della costruzione del grafo
- costruire il grafo e poi rimuovere i percorsi che creano bolle e sono supportati da **kmers rari**
- **spezzare le reads in kmers più piccoli (per de Bruijn graphs)**

Problema 1: abbiamo errori nelle sequenze (o bolle)



In dati reali ci sono molti errori di sequenziamento (quindi molte bolle)!



La maggior parte dei k-mers non hanno errori e così posso utilizzare quasi tutta la sequenza senza preoccuparmi dell'errore

Assemblatori moderni correggono queste bolle sulla base di alcune “euristiche”:

- rimuovere i **kmers rari*** prima della costruzione del grafo
- costruire il grafo e poi rimuovere i percorsi che creano bolle e sono supportati da **kmers rari**
- **spezzare le reads in kmers più piccoli (per de Bruijn graphs)**

Soprattutto le tecnologie con long reads hanno molti errori

Library schematic	Output
Illumina  The diagram shows a horizontal DNA strand. The left end is a red segment labeled 'P5' in red text. The right end is an orange segment labeled 'P7' in orange text. The middle section of the strand is a wavy blue line.	Sequenze corte (50-250 bp) e relativamente accurate. Economico quindi alta coverage. Ad oggi ancora il 75% circa dei dati presenti.
PacBio  The diagram shows a horizontal DNA strand. At both the left and right ends, the strand forms a red circular loop labeled 'Hairpin adapters' in red text. The middle section of the strand is a wavy blue line.	Sequenze HiFi accurate (errore < 0.5%) e lunghe >10kb.
Oxford Nanopore  The diagram shows a horizontal DNA strand. On the left end, there is a purple motor protein (two spheres) labeled 'Motor protein' in purple text. Below the motor protein is a red segment labeled 'Tether oligo' in orange text. The middle section of the strand is a wavy blue line.	Sequenze ultra-lunghe di anche >100kb ma con errore <10%

Soprattutto le tecnologie con long reads hanno molti errori

Library schematic

Output

Spezzare in k-mers porta a perdere inevitabilmente informazione a lungo raggio, che come vedremo è fondamentale per le ripetizioni!



PacBio

Hairpin adapters



Sequenze HiFi accurate (errore < 0.5%) e lunghe >10kb.

Oxford Nanopore

Motor protein



Sequenze ultra-lunghe di anche >100kb ma con errore <10%

Soprattutto le tecnologie con long reads hanno molti errori

Library schematic

Output

Spezzare in k-mers porta a perdere inevitabilmente informazione a lungo raggio, che come vedremo è fondamentale per le ripetizioni!



PacBio

Hairpin adapters



Sequenze HiFi accurate (errore < 0.5%)
lunghe >10kb.

Oxford Nanopore

Motor protein

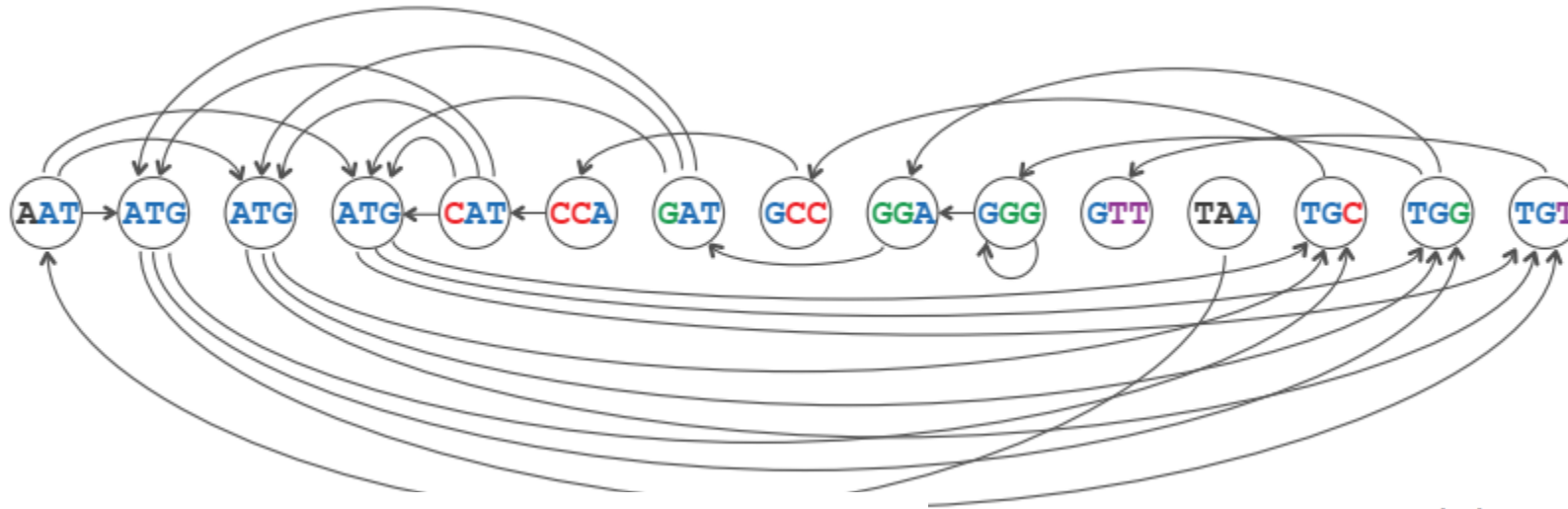


Sequenze ultra-lunghe di anche >100kb
ma con errore <10%

Short reads (che hanno meno errori) spesso sono processate come 25mers, mentre alcune long reads come 20mers visti i maggiori errori!!!! Quanto spreco!!!

Gli «overlap graphs» sono alla base di vari assemblatori moderni

(specializzati in sequenze molto lunghe e accurate).



ARTICLES

<https://doi.org/10.1038/s41592-020-01056-5>

nature | methods

Check for updates

Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm

Haoyu Cheng^{1,2}, Gregory T. Conception³, Xiaowen Feng^{1,2}, Haowen Zhang⁴ and Heng Li^{1,2}✉



Method

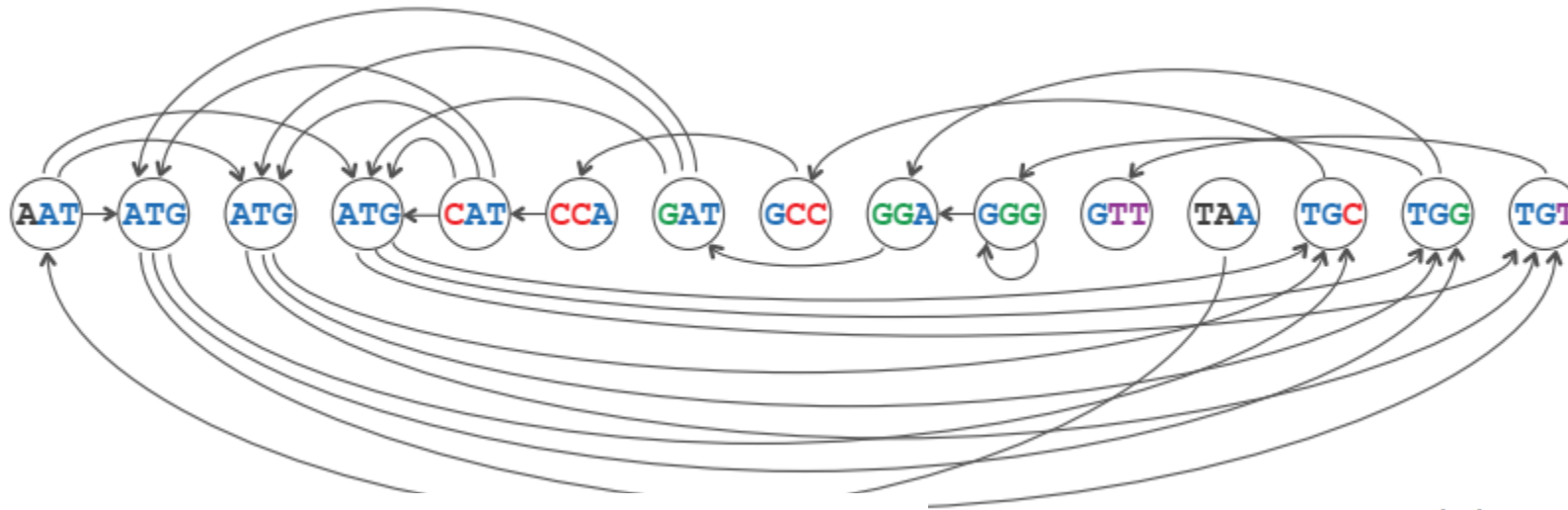
HiCanu: accurate assembly of segmental duplications, satellites, and allelic variants from high-fidelity long reads

Sergey Nurk,^{1,6} Brian P. Walenz,^{1,6} Arang Rhie,¹ Mitchell R. Vollger,² Glennis A. Logsdon,² Robert Grothe,³ Karen H. Miga,⁴ Evan E. Eichler,^{2,5} Adam M. Phillippy,¹ and Sergey Koren¹



Gli «overlap graphs» sono alla base di vari assemblatori moderni

(specializzati in sequenze molto lunghe e accurate).



Anche per questo gli overlap-graph sono nuovamente usati abbondantemente nonostante abbiamo visto che siano computazionalmente «intrattabili»



ARTICLES

<https://doi.org/10.1038/s41592-020-01056-5>

nature | methods

Check for updates

Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm

Haoyu Cheng^{1,2}, Gregory T. Conception³, Xiaowen Feng^{1,2}, Haowen Zhang⁴ and Heng Li^{1,2} ✉



Method

HiCanu: accurate assembly of segmental duplications, satellites, and allelic variants from high-fidelity long reads

Sergey Nurk,^{1,6} Brian P. Walenz,^{1,6} Arang Rhie,¹ Mitchell R. Vollger,² Glennis A. Logsdon,² Robert Grothe,³ Karen H. Miga,⁴ Evan E. Eichler,^{2,5} Adam M. Phillippy,¹ and Sergey Koren¹



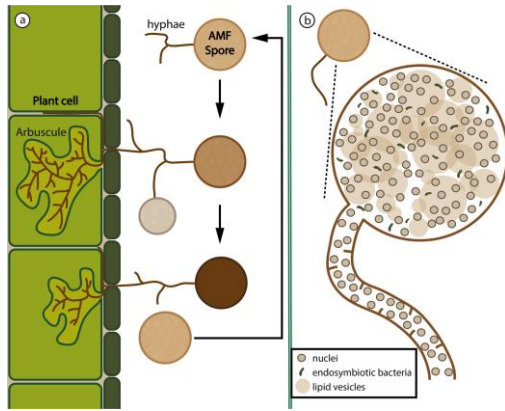
Gli assemblatori basati su grafi di De Bruijn sono ancora utili per le short reads, la metagenomica e per varie applicazioni specifiche come alcuni genomi estremamente ripetuti

SPAdes

Aimed at short reads, small genomes, metagenomes, single-cell data



Human
Microbiome
Project



Montoliu-Nerin et al., 2020

Verkko2 integrates proximity-ligation data with long-read De Bruijn graphs for efficient telomere-to-telomere genome assembly, phasing, and scaffolding

Dmitry Antipov,^{1,4} Mikko Rautiainen,^{2,4} Sergey Nurk,³ Brian P. Walenz,¹ Steven J. Solar,¹ Adam M. Phillippy,¹ and Sergey Koren¹



High-quality metagenome assembly from long accurate reads with metaMDBG *Nature Biotechnology* **42**, 1378–1383 (2024)

Gaëtan Benoit, Sébastien Raguideau, Robert James, Adam M. Phillippy, Rayan Chikhi & Christopher Quince

LJA

Multiplex de Bruijn graphs enable genome assembly from long, high-fidelity reads

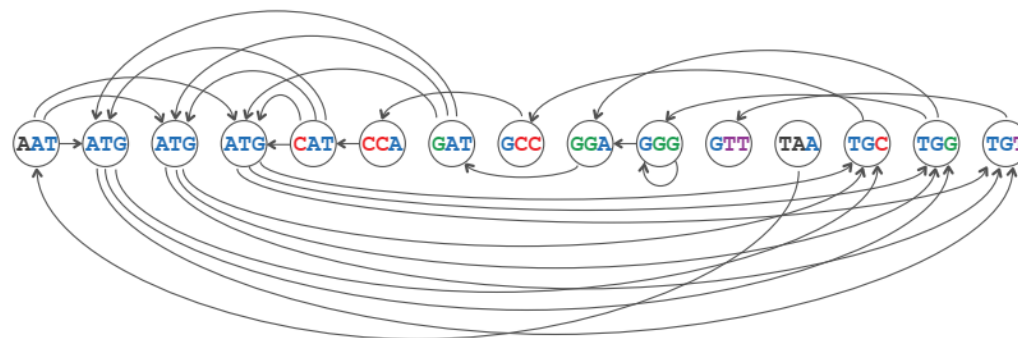
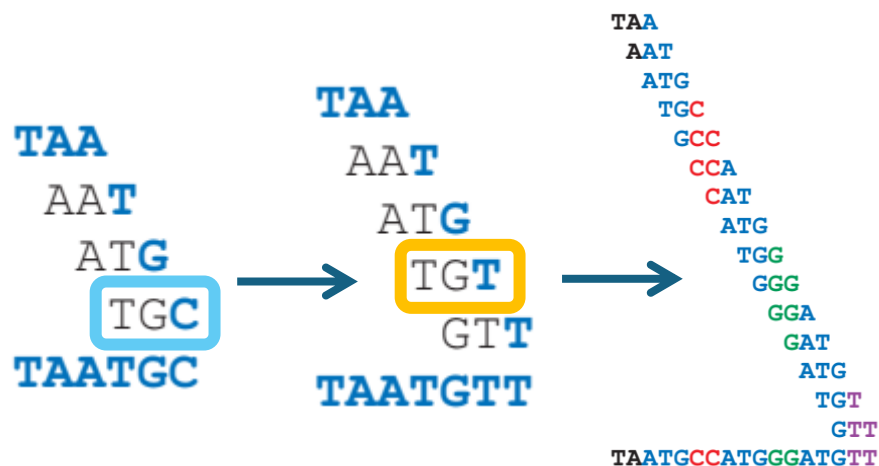
Anton Bankevich , Andrey V. Bzikadze, Mikhail Kolmogorov, Dmitry Antipov & Pavel A. Pevzner 

Nature Biotechnology **40**, 1075–1081 (2022) | [Cite this article](#)

Fast and feasible for large repeated genomes like barley

Due approcci per assemblare un genoma

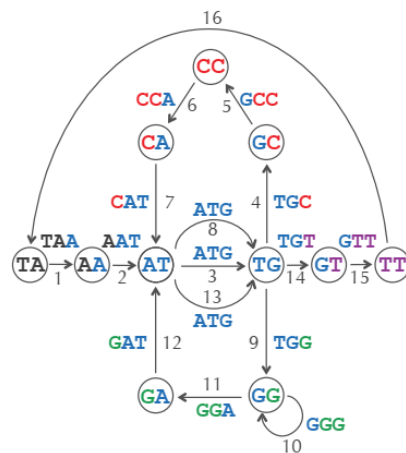
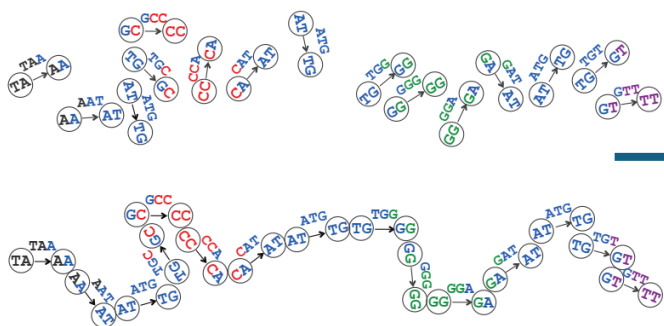
Esplorazione degli **Overlap-graph** per «forza bruta» ed usando al più delle «euristiche»



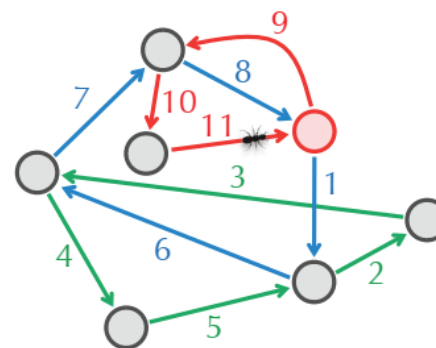
Applicazioni

Applicabili soprattutto a sequenze ultralunghe e accurate, ad esempio per genomi ultracompleti (Telomere 2 Telomere).

Costruzione del **De Bruijn Graph**



Ricerca del «ciclo Euleriano»*

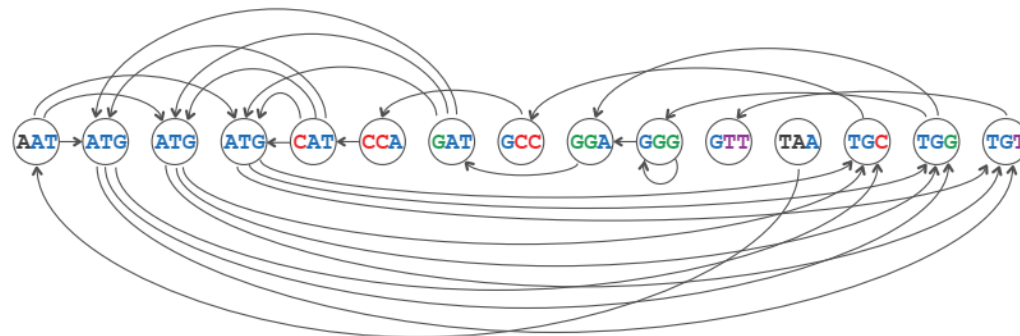
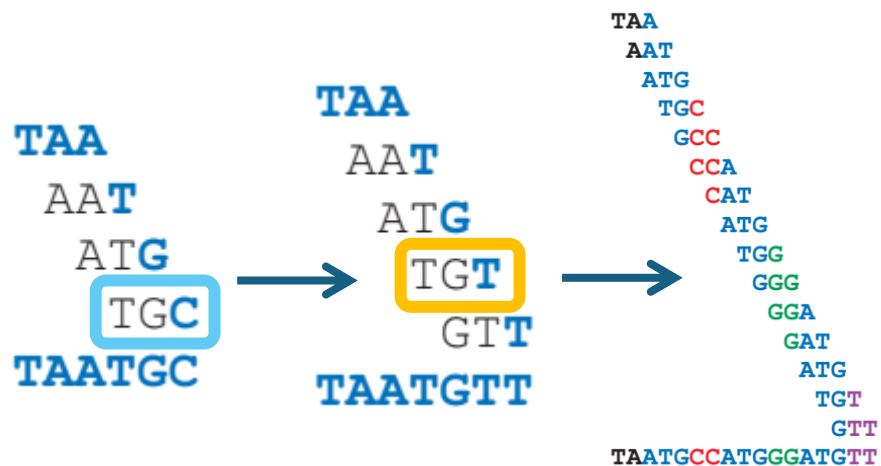


*con l'algoritmo di Leo la formica

Metagenomi, genomi microbici, dati vari incluse sequenze corte (illumina), ma anche phasing e genomi iper-ripetuti

Due approcci per assemblare un genoma

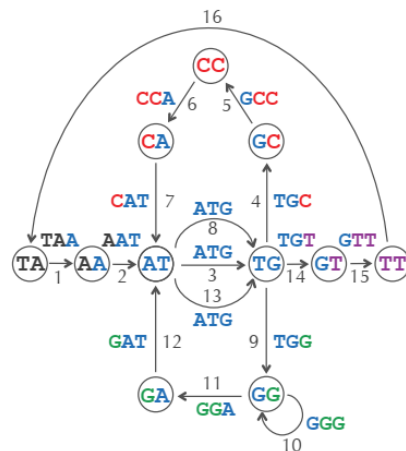
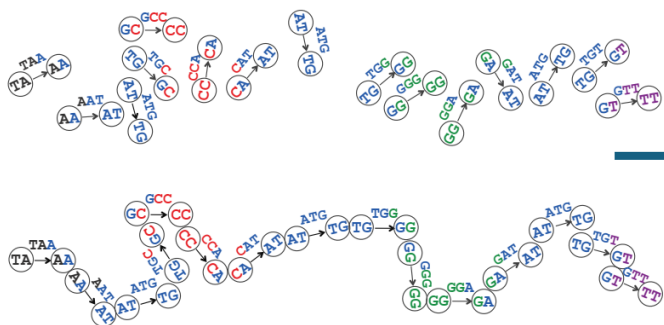
Esplorazione degli **Overlap-graph** per «forza bruta» ed usando al piu delle «euristiche»



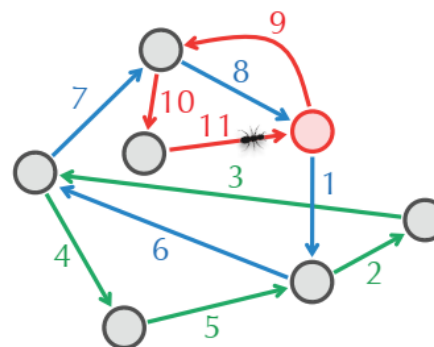
Applicazioni



Costruzione del **De Bruijn Graph**



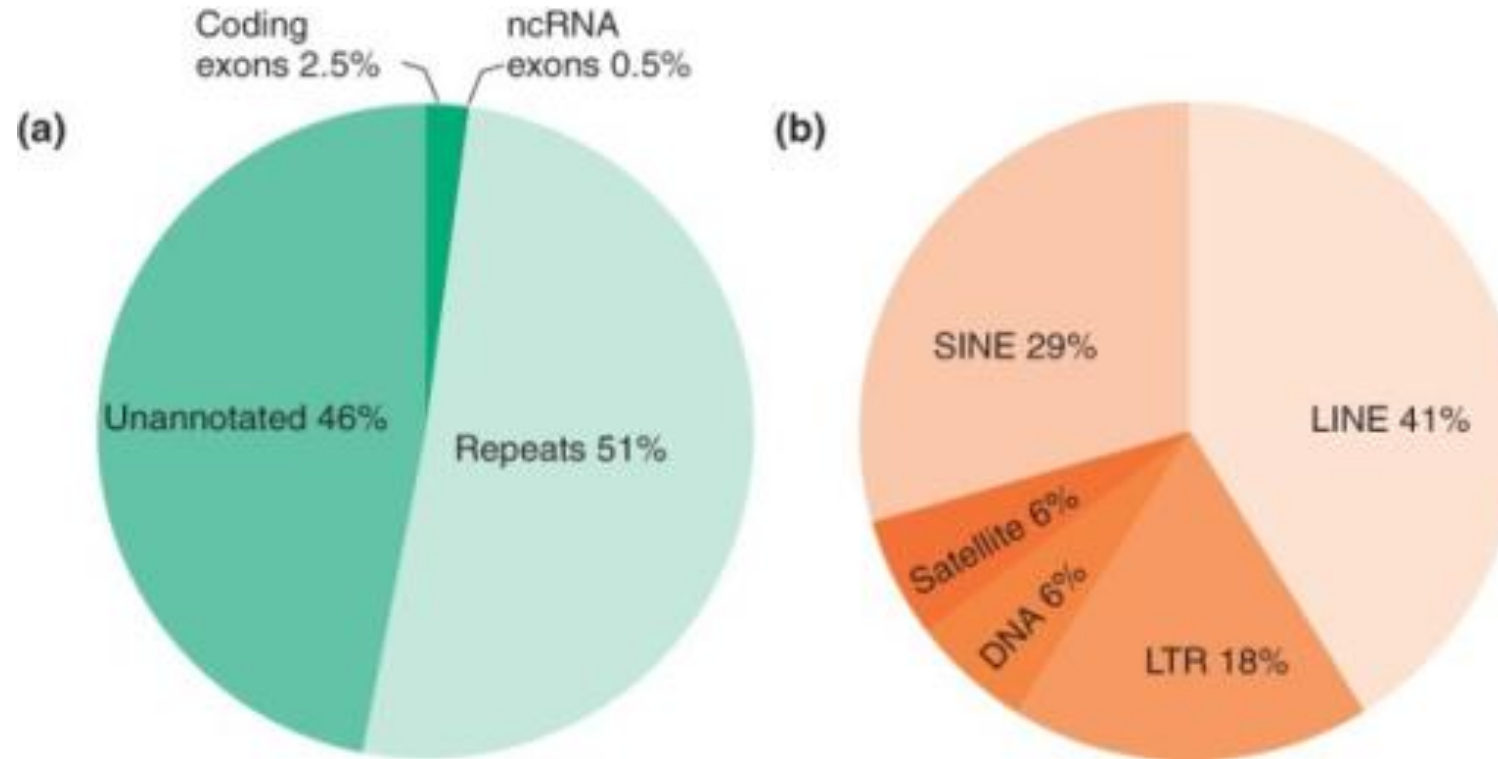
Ricerca del «ciclo Euleriano»*



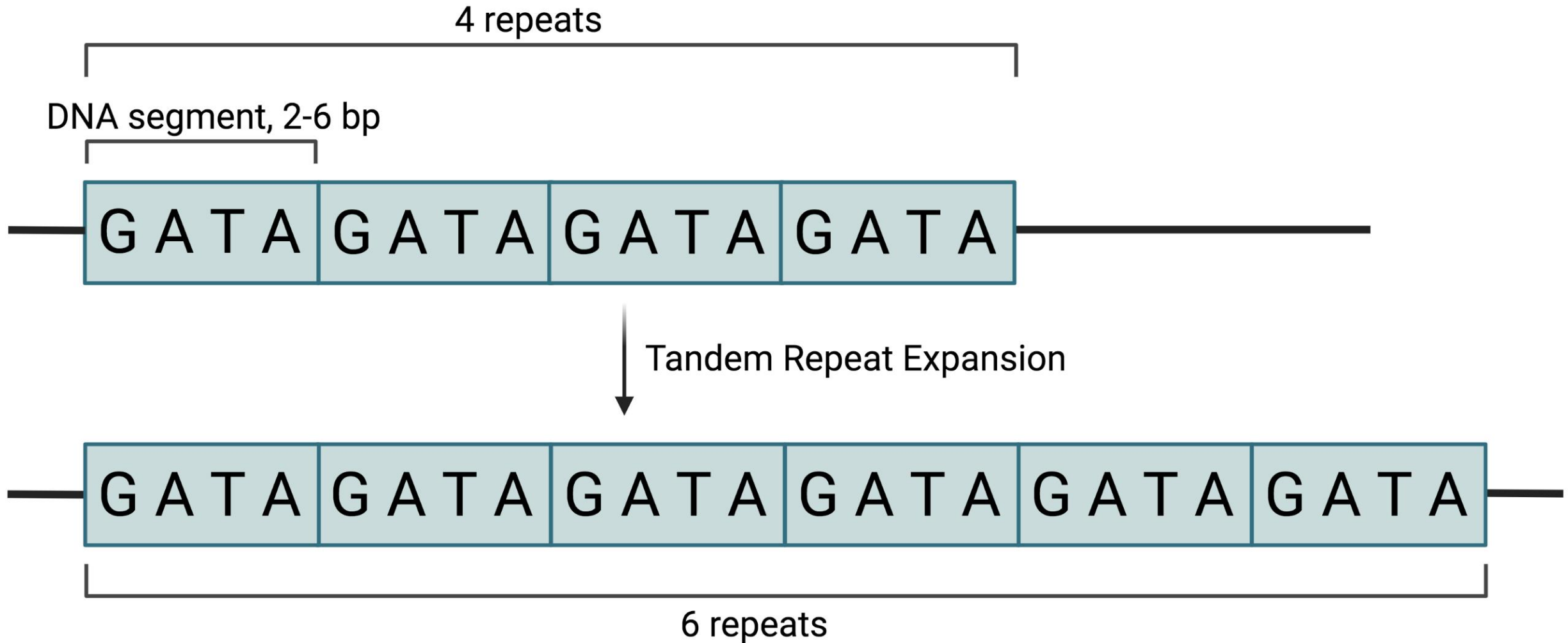
*con l'algoritmo di Leo la formica



Problema 2: Il genoma (umano) è altamente ripetitivo

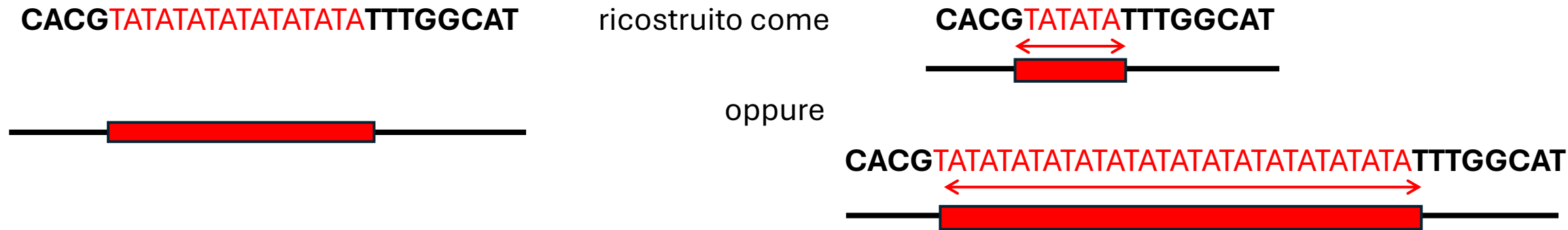


Short Tandem Repeat (STR) *o microsatelliti*



I problemi introdotti dalle ripetizioni per reads corte (o non abbastanza lunghe per “superare” la ripetizione)

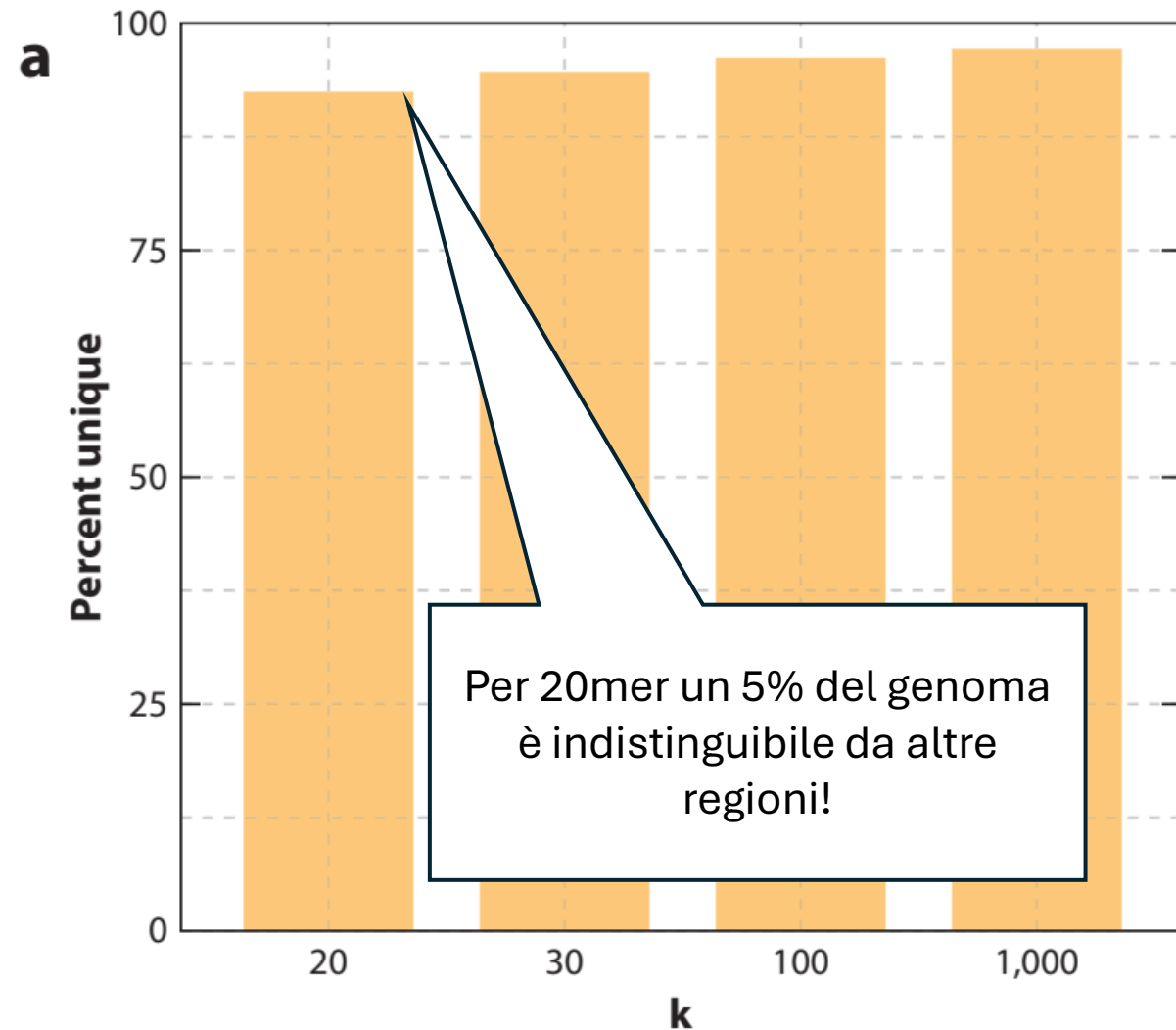
- Le zone ripetute sono ricostruite erroneamente



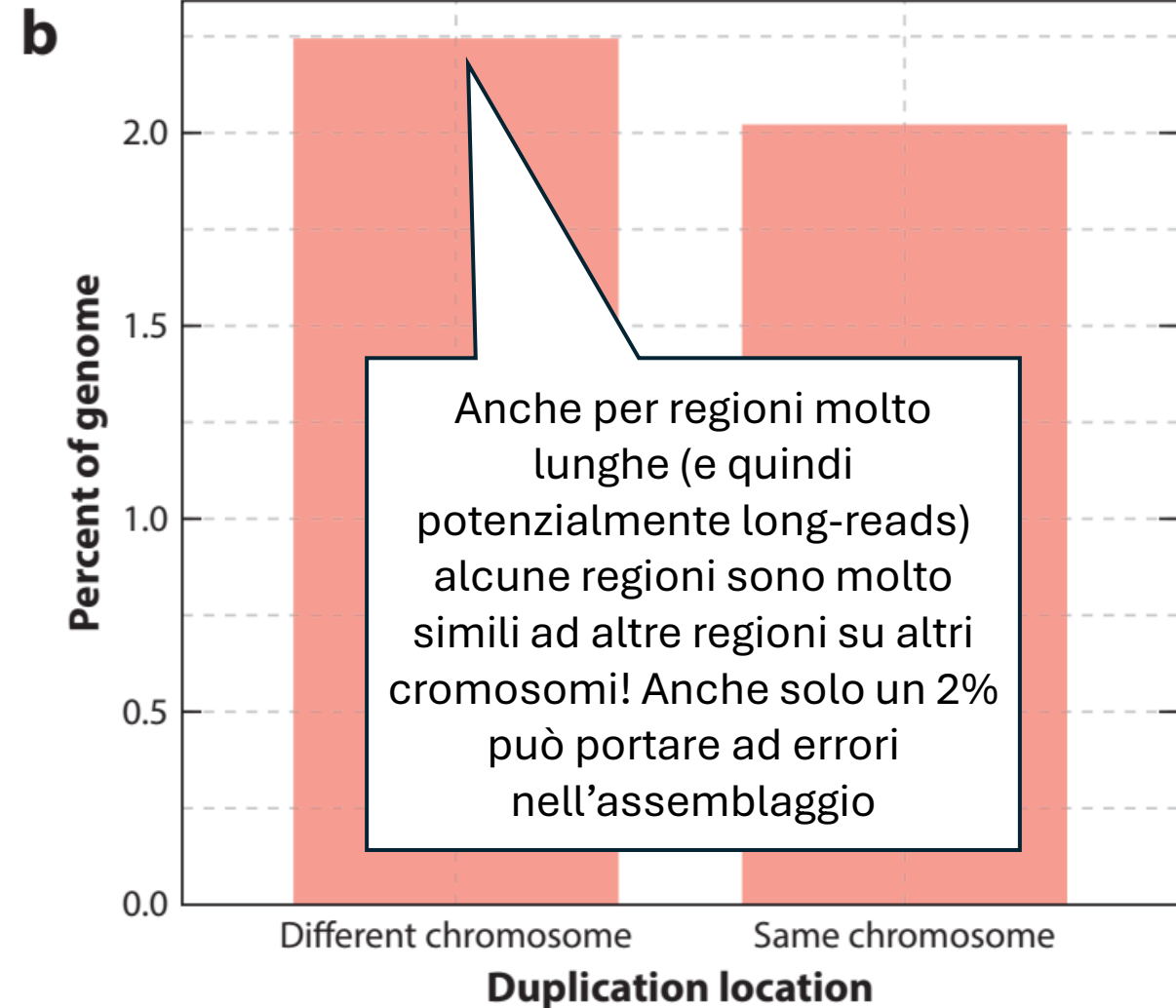
- Piu' in generale, diventa impossibile ricostruire un genoma in modo ragionevole a causa dei molti «percorsi» errati nell'overlap graph



Tuttavia alcune regioni ripetute sono >5kb e possono essere risolte solo con sequenze lunghissime e accurate (o con lo scaffolding)

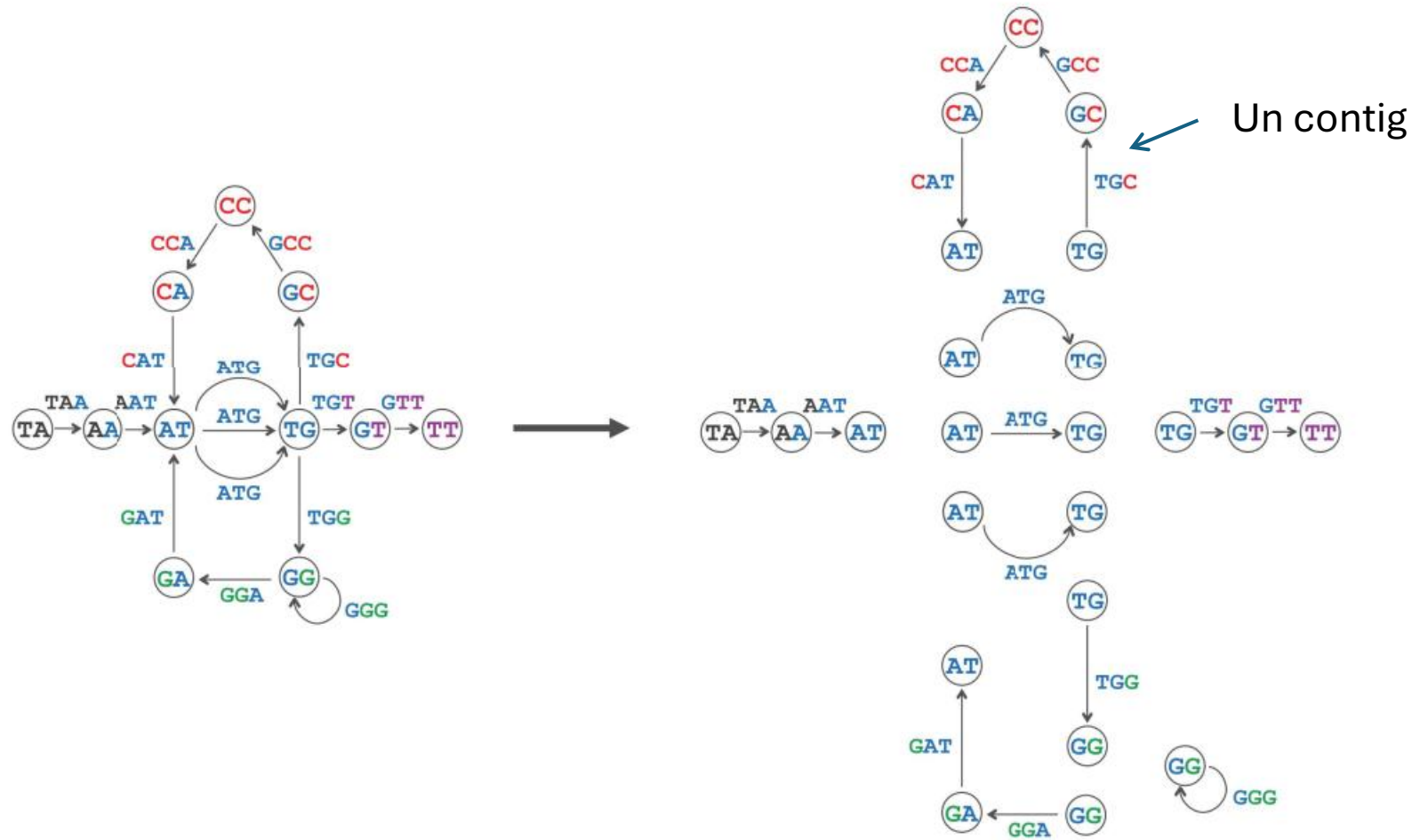


Percentuale di genoma unico per k-mers di varie lunghezze



Percentuale di genoma duplicato >5kb con similarità >95%

In pratica, gli assemblaggi vengono ridotti a contigs non ambigui



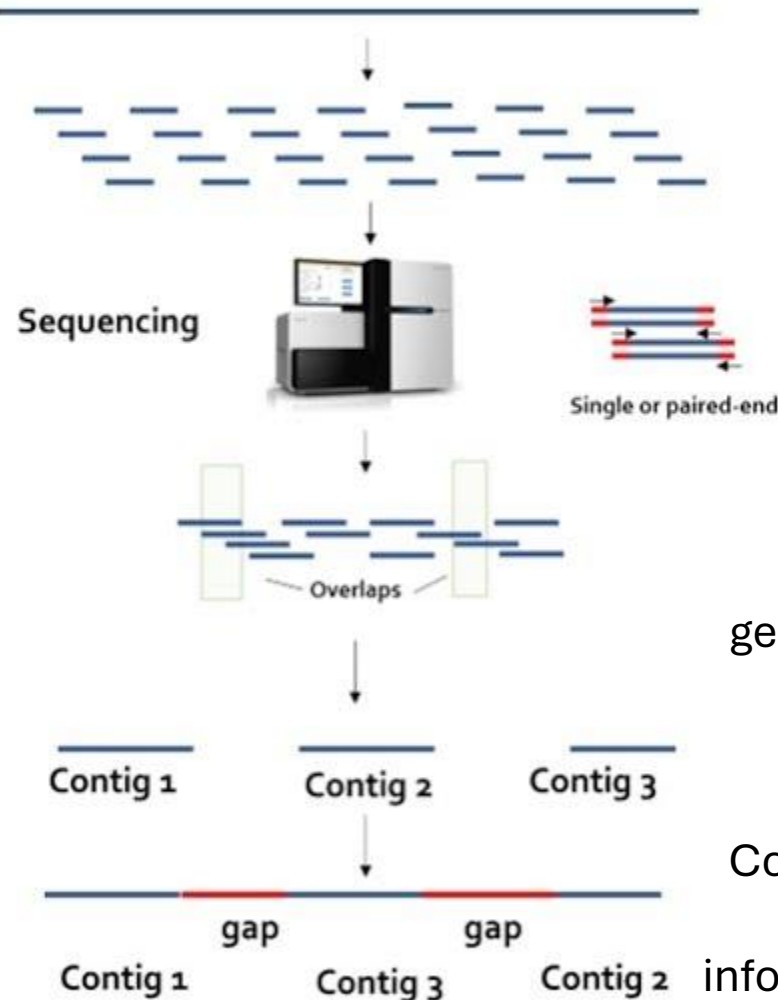
Reads, contigs and scaffolds (in genomic context)

Genome

Reads

Contigs

Scaffold



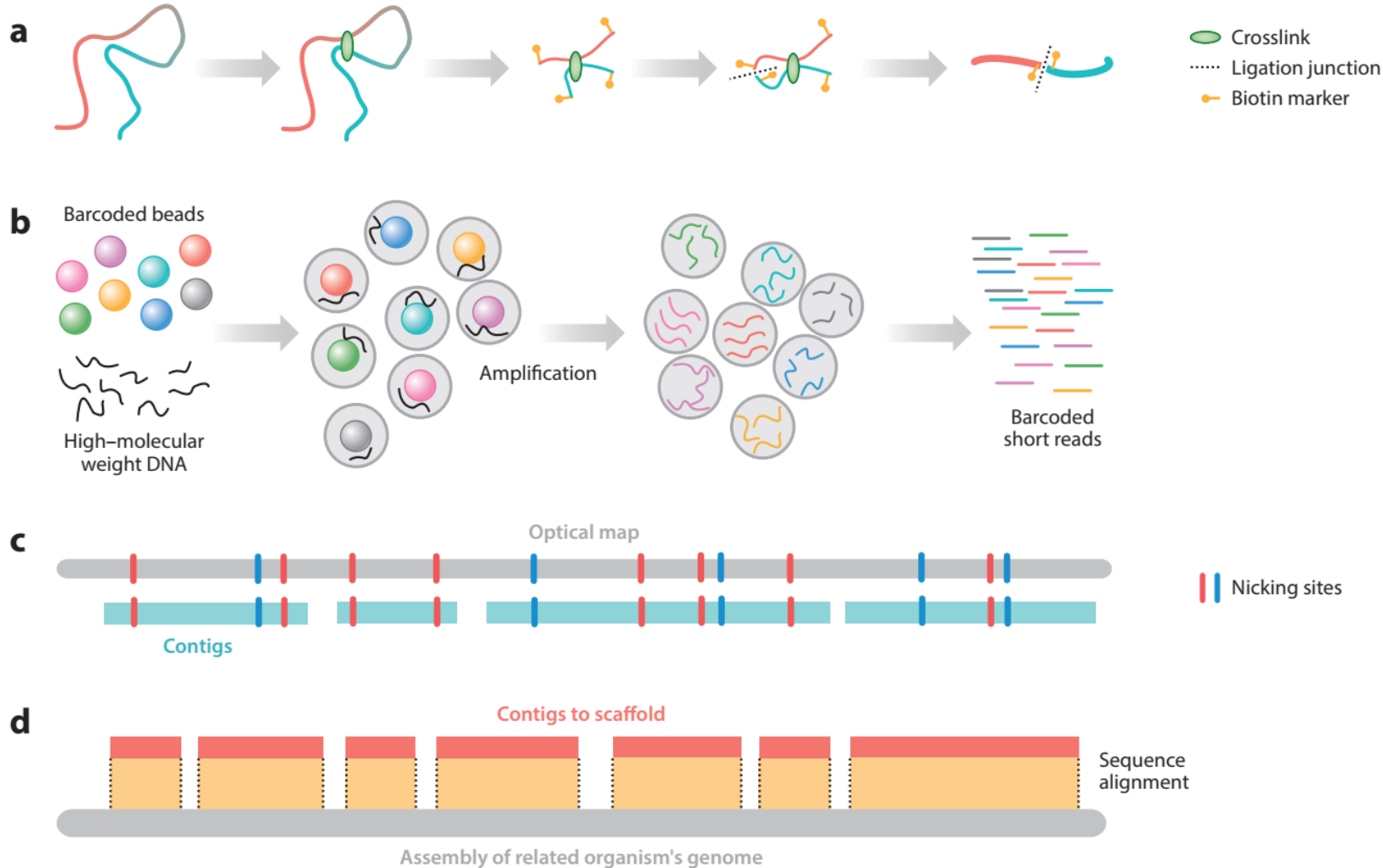
Assemblaggio

generalmente con grafi
di de Bruijn

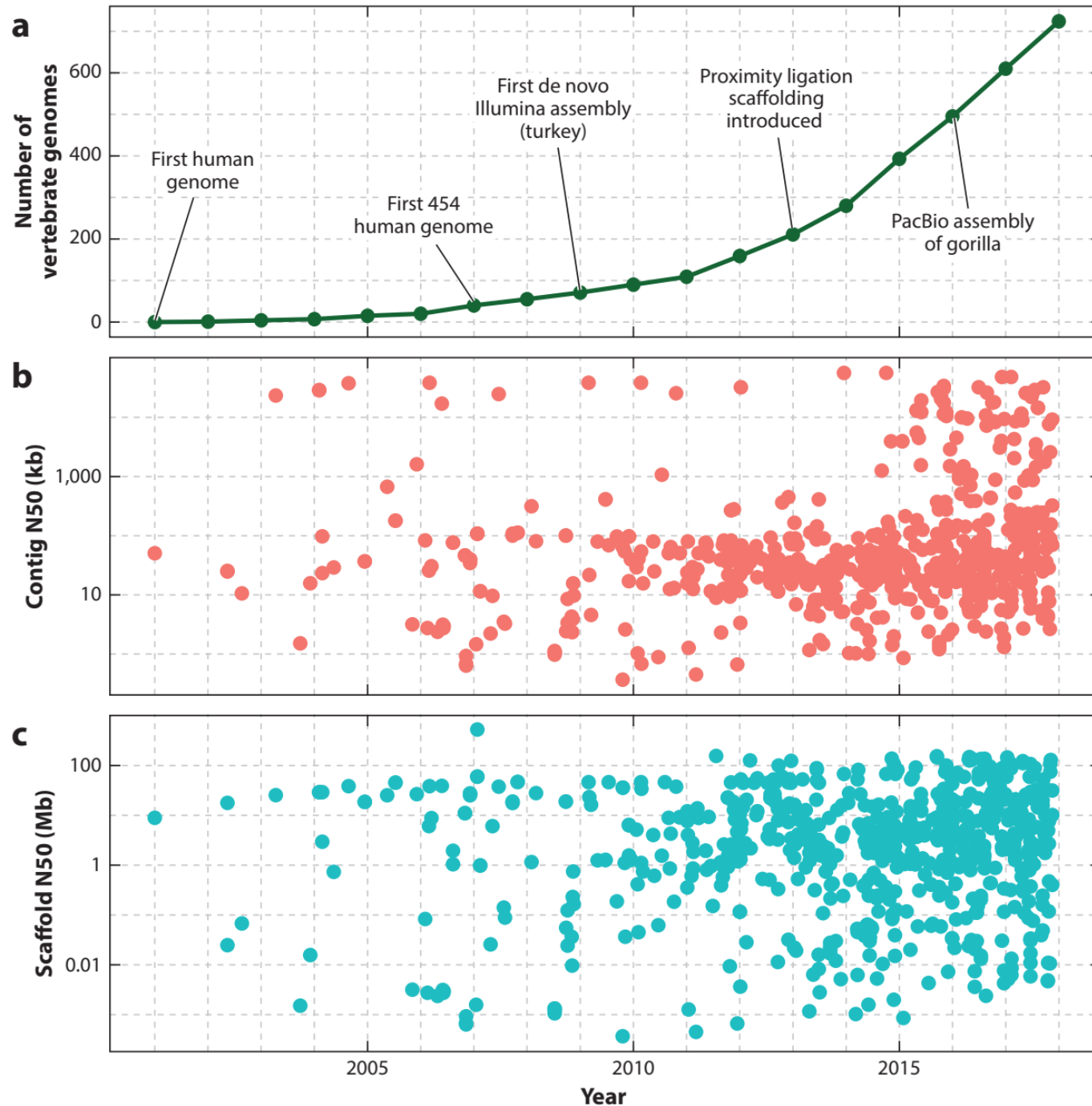
Scaffolding

Contig separate vengono
“uniti” con fonti di
informazione a lungo raggio

Lo scaffolding permette di «unire» piu' contigs

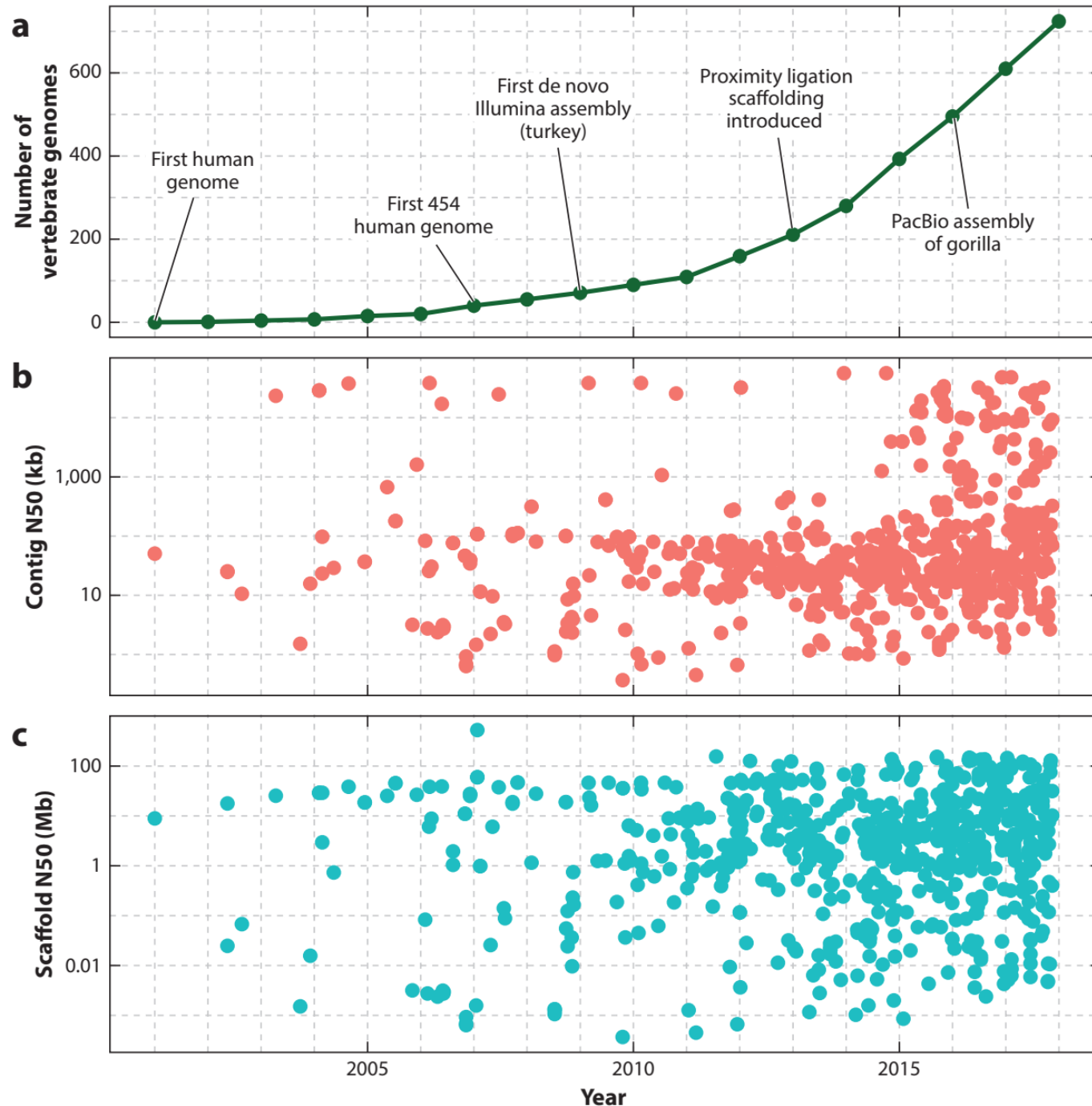


- Proximity ligation (Hi-C)
- Linked-read sequencing (10x), amplificazione delle sequenze dentro droplets
- Enzimi che attaccano marcatori fluorescenti (BioNano)
- Approcci basati su sintenia che usano altri genomi come referenza



L'arrivo di alcune tecnologie (Hi-C) ha portato a miglioramenti nella contiguità, ma il grosso dei genomi è stato finora limitato (dalle lunghe sequenze ripetute).

Contiguità dei genomi



L'arrivo di alcune tecnologie (Hi-C) ha portato a miglioramenti nella contiguità, ma il grosso dei genomi è stato finora limitato (dalle lunghe sequenze ripetute).

Solo dal 2019 circa le sequenze accurate ultralunghe hanno permesso i primi genomi T2T con overlap-graphs

Contiguità dei genomi

Misurare la contiguità di un assemblaggio: N50

Definizione: la lunghezza del contig piu piccolo nel piu' piccolo insieme di contig che contiene il 50% del
genoma

O

Cercando di selezionare solo i contigs piu' lunghi, il valore del contig piu' piccolo con il quale riesco ancora a rappresentare almeno il 50% del genoma. Il numero di quei contigs viene definito **L50**

Esempio con 1Mb:



N50 size = 30 kbp

$(300k + 100k + 45k + 45k + 30k = 520k \geq 500kbp)$

Misurare la contiguità di un assemblaggio: N50

Definizione: la lunghezza del contig piu piccolo nel piu' piccolo insieme di contig che contiene il 50% del
genoma

O

Cercando di selezionare solo i contigs piu' lunghi, il valore del contig piu' piccolo con il quale riesco ancora a rappresentare almeno il 50% del genoma. Il numero di quei contigs viene definito **L50**

Esempio con 1Mb:



Alto N50, basso L50: pochi contigs molto lunghi

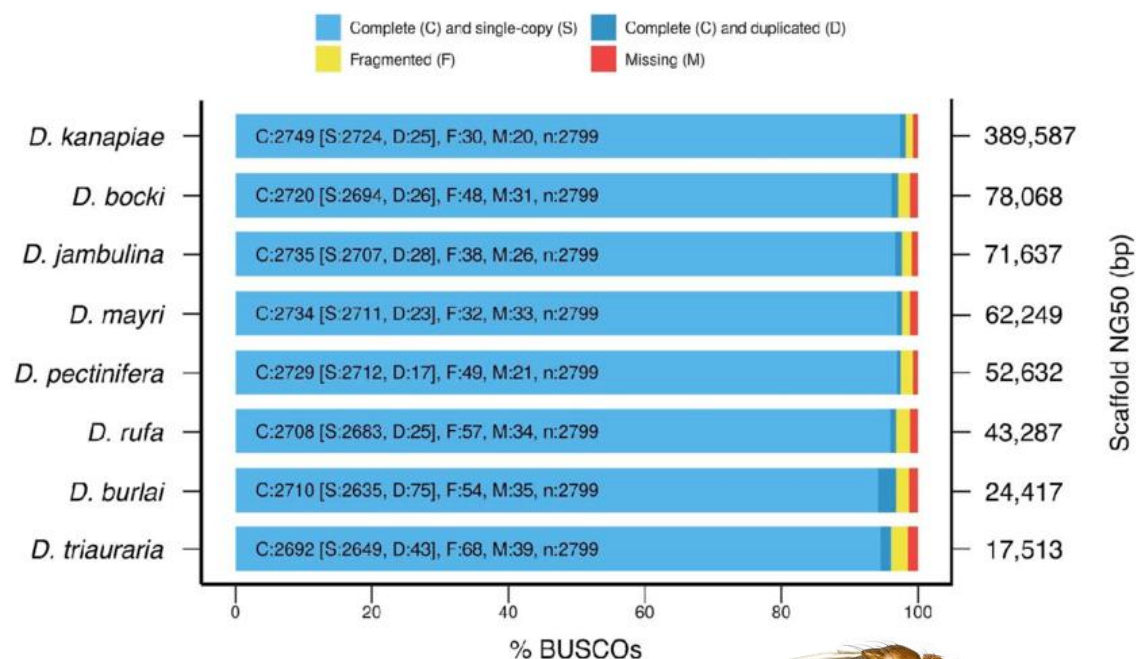


Basso N50, Alto L50: molti contigs corti

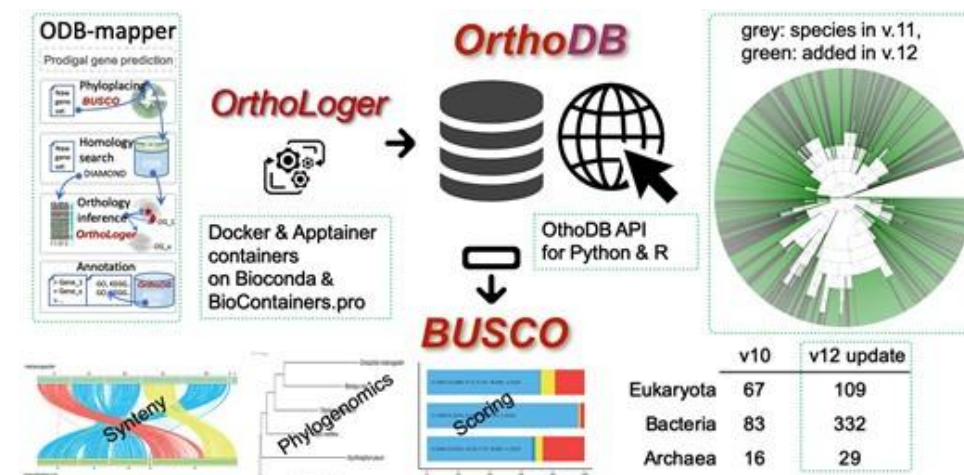


Misurare la completezza dei genomi assemblati

- Una collezione di geni che ci si attende siano presente in quasi ogni organismo.
- Permette di misurare la completezza di un genoma semplicemente in termini di percentuali di geni BUSCO rappresentati (e se frammentati o duplicati) in un genoma assemblato.



BUSCO



Nucleic Acids Research, 2025, 53, D516–D522
<https://doi.org/10.1093/nar/gkaf987>
 Advance access publication date: 13 November 2024
 Database issue



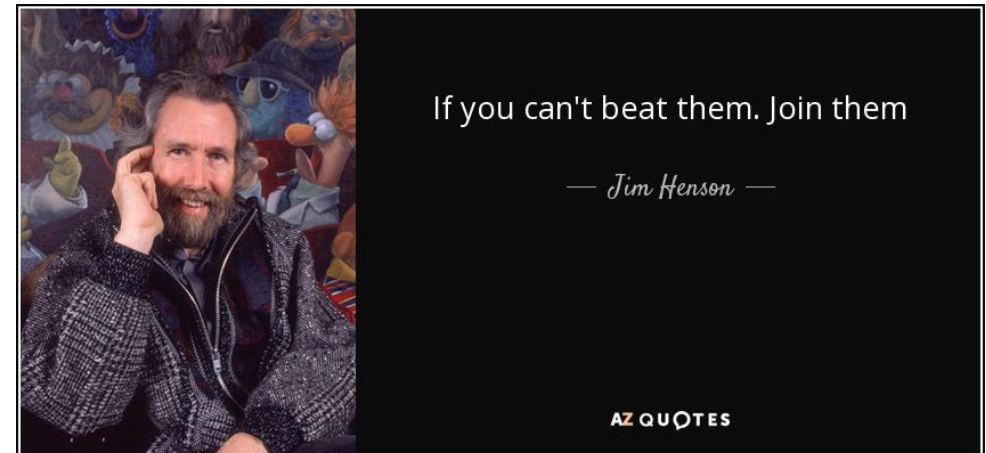
OrthoDB and BUSCO update: annotation of orthologs with wider sampling of genomes

Fredrik Tegenfeldt[†], Dmitry Kuznetsov[†], Mosè Manni[†], Matthew Berkeley[†], Evgeny M. Zdobnov[✉] and Evgenia V. Kriventseva[✉]

Problema 2: Ripetitività del genoma

Soluzioni
«sperimentali»

- Sequenziare reads **piu' lunghe** ed accurate:
Hi-fi, ultra-long read sequencing
- Scaffolding: unire contigs e bypassare ripetizioni con Hi-C, Bionano, genetic maps e altre fonti di informazione/tecnologie



Problema 3: Ripetitività DEL genoma (intero)

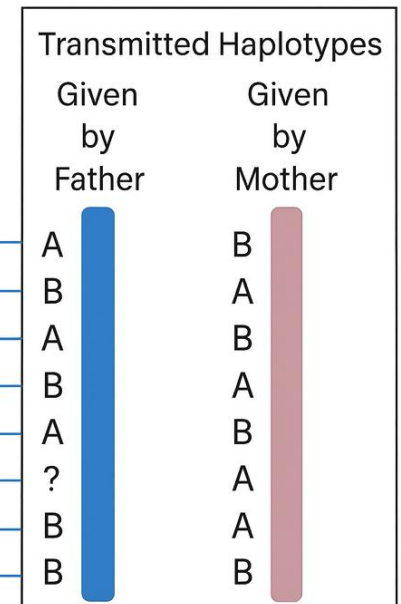
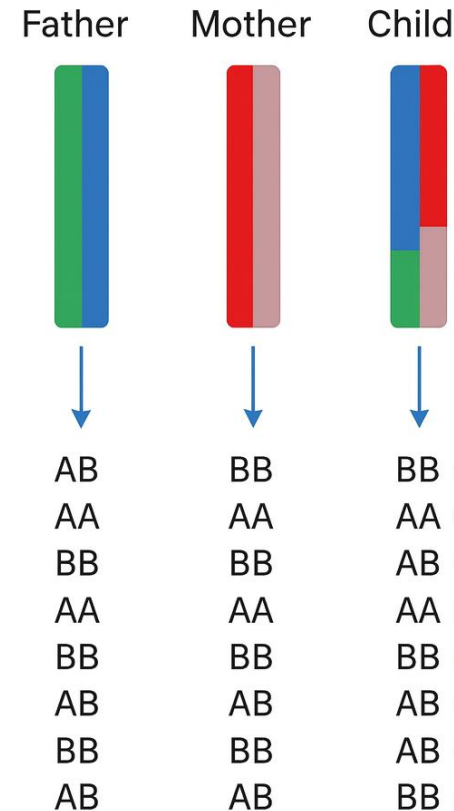
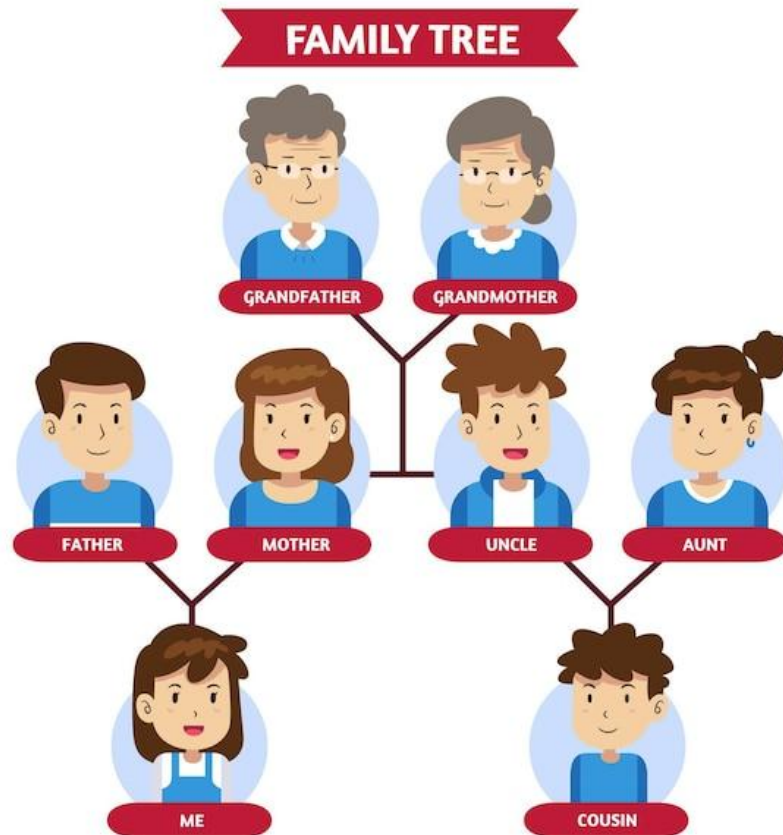
Problema 3: Ripetitività DEL genoma (intero)

Il nostro intero genoma è ripetuto
almeno una volta!!!!



Un tipo di ripetizioni «quasi» inevitabile: i genomi diploidi!!

- Il nostro intero genoma è ripetuto almeno una volta!!!!



Un tipo di ripetizioni «quasi» inevitabile: i genomi diploidi!!

Una prima semplice soluzione sperimentale: non usare diploidi!
Usare individui «inbred» od omozigoti



Generazione 1



Generazione 2



Generazione 3

Un tipo di ripetizioni «quasi» inevitabile: i genomi diploidi!!

- Una prima semplice soluzione sperimentale: non usare diploidi! Usare individui «inbred» od omozigoti



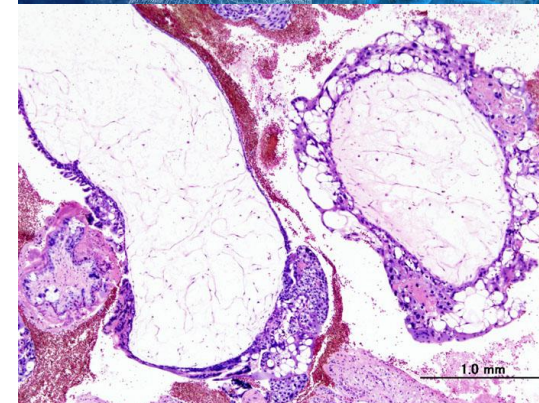
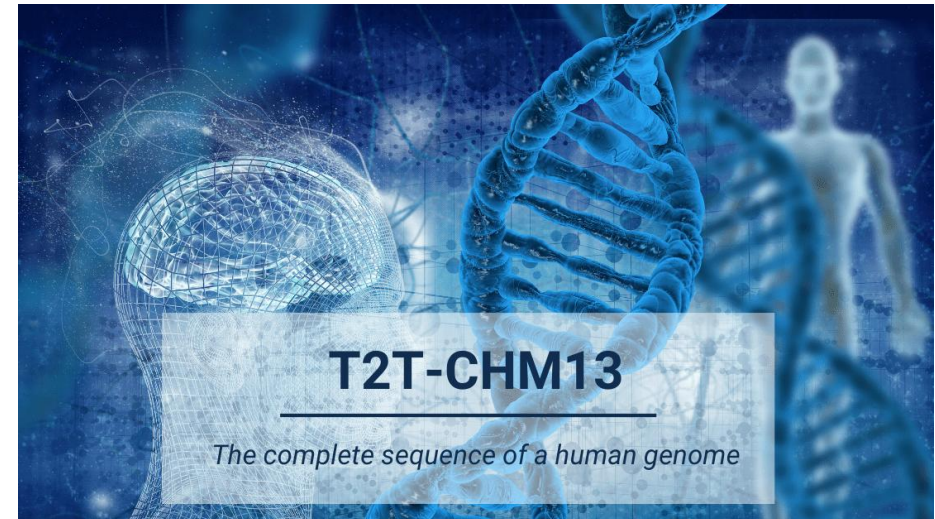
Generazione 1



Generazione 2



Generazione 3



Ottenuto da una **mole idatiforme*** interamente **omozigote e con cariotipo 46XX**.

*un embrione anomalo che cresce come massa nell'utero in cui spesso uno dei due «genomi» parentali viene espulso, e che non sviluppa in un feto

Un tipo di ripetizioni «quasi» inevitabile: i genomi diploidi!!

- Una prima semplice soluzione sperimentale: non usare diploidi!
Usare individui «inbred» od omozigoti



Generazione 1



Generazione 2



Generazione 3

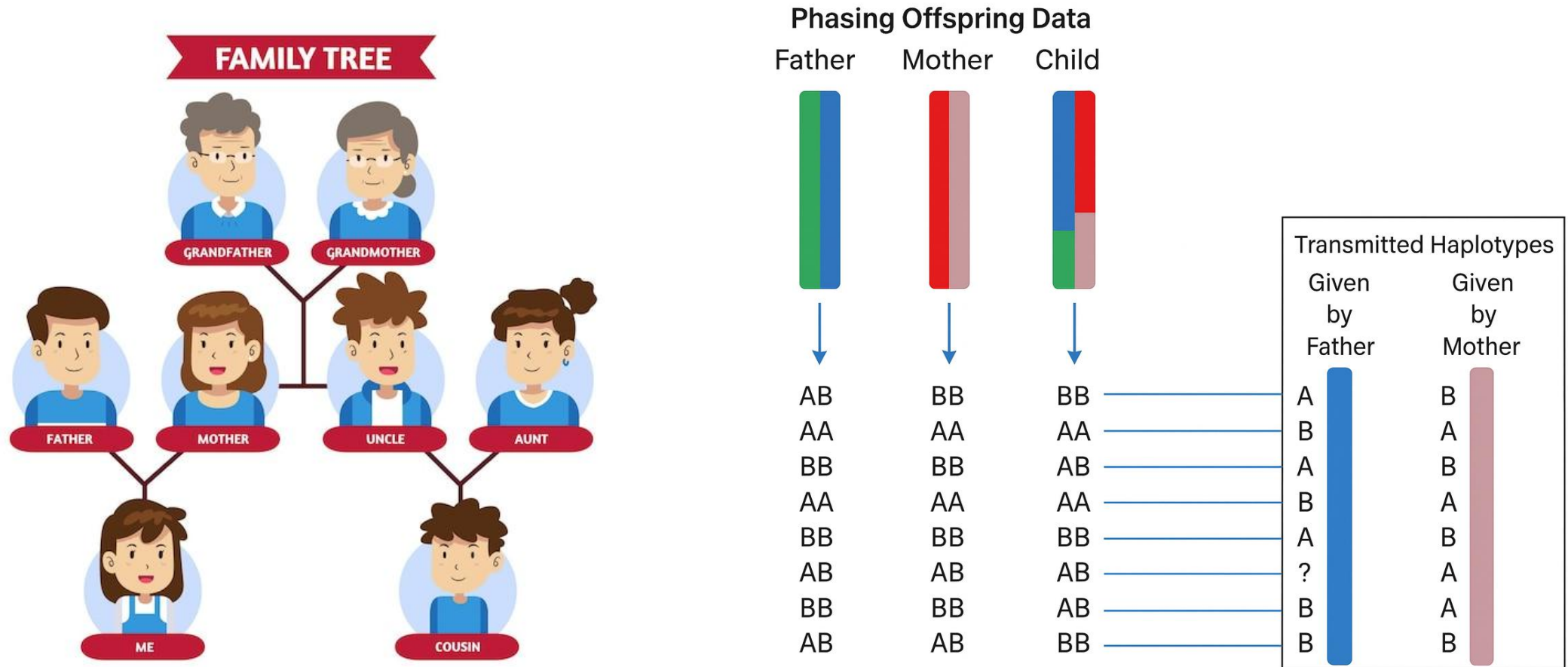


Quasi ubiquitario
nelle piante, dove
molte si auto-
fertilizzano (grano,
arabidopsis) o è
facile creare linee
«inbred»



Un tipo di ripetizioni «quasi» inevitabile: i genomi diploidi!!

- Una seconda strategia: utilizzo di alberi genealogici (o trio). Sequenziare anche i genitori per inferire la “fase” dei cromosomi!



Un tipo di ripetizioni «quasi» inevitabile: i genomi diploidi!!

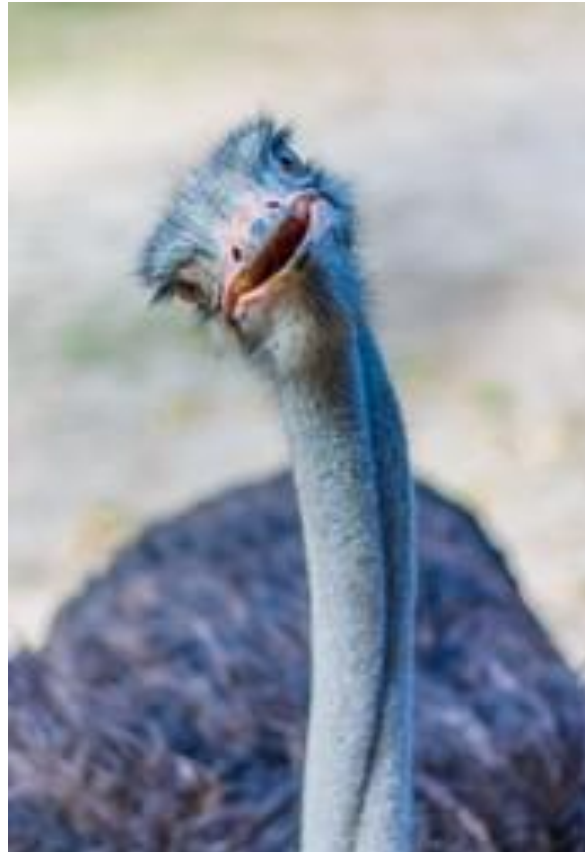
Un'altra strategia molto usata: ignorare il problema.



Un tipo di ripetizioni «quasi» inevitabile: i genomi diploidi!!

Un'altra strategia molto usata: ignorare il problema.

Perché è un problema?



Cosa succede ad ignorare la diploidia

Il genoma di Bart Simpson



Una possibile risoluzione di un assemblaggio che ignora la diploidia del genoma:

- errori di assemblaggio
- sovrastima della lunghezza



Oggi le recenti tecnologie di sequenziamento ultra-lungo permettono e gli algoritmi di assemblaggio permettono di affrontare il problema (siamo agli inizi) risolvendo i due aplotipi «materno» e «paterno»

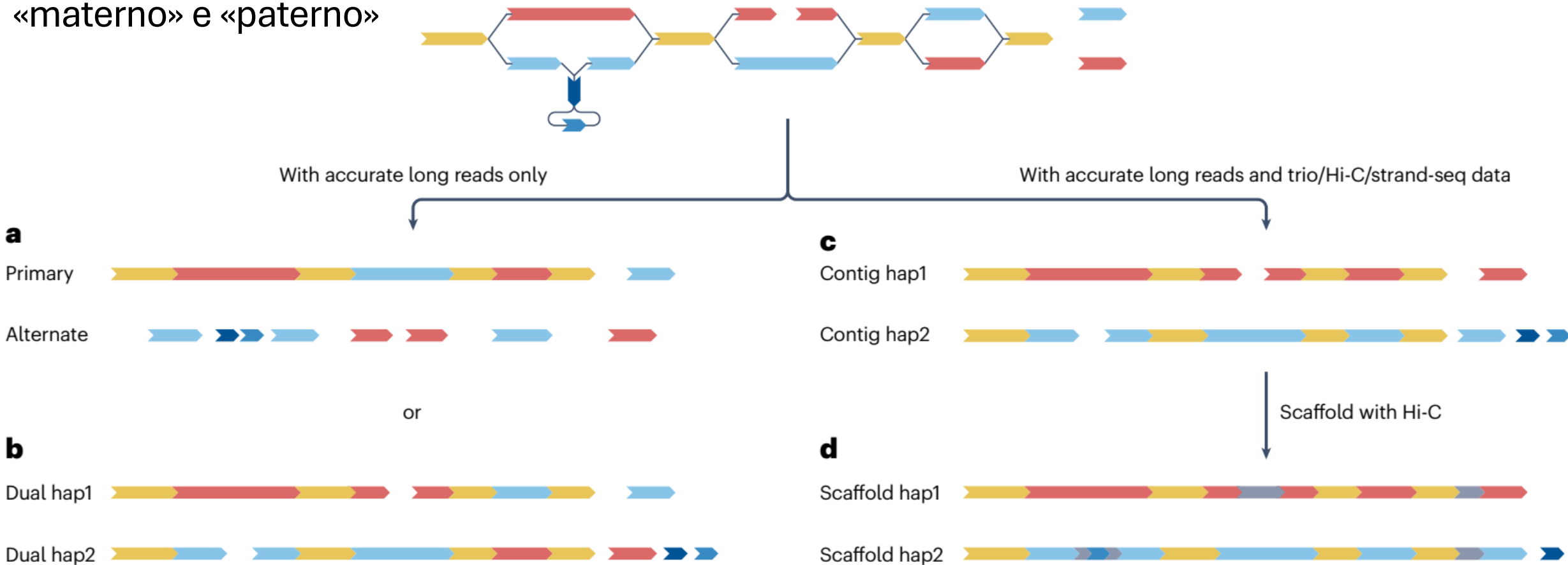


Fig.2 | Types of phased assembly of diploid samples. The initial assembly graph can be further processed into different types of assemblies. If only accurate long-read sequence data are available, a primary–alternate assembly pair or a dual assembly pair can be generated. If long-range data (such as trio, Hi-C or strand-seq data) are available, chromosome-phased assemblies can be generated. **a**, In a primary–alternate assembly pair, the primary assembly represents a complete haploid genome with occasional phase switches whereas the alternate assembly

is fragmented and contains no homozygous regions longer than the read length. **b**, In a pair of dual assemblies, each assembly is similar to a primary assembly. **c**, In a pair of chromosome-phased assemblies, long-range data are used to partition contigs from the same haploid chromosome to the same assembly. **d**, In a pair of chromosome-phased assemblies with scaffolding, Hi-C data are used to join contigs into chromosomes across assembly gaps (marked by thick grey arrows). hap, haplotype; trio data, sequencing data on an individual and both its parents.

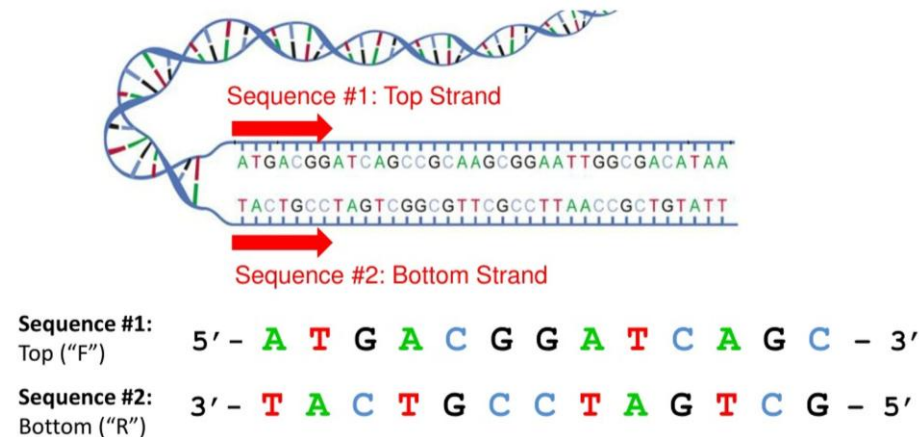
Un tipo di «ripetizione» totalmente inevitabile

Il DNA è a doppio filamento!!!

Di conseguenza per ogni filamento **GATGTCATA** avremo anche **TATGACATC**

Per risolvere questo problema, vengono usate euristiche:

- 1) Creazione per ogni sequenza del suo reverse complement
- 2) Risoluzione del «doppio grafo» a posteriori

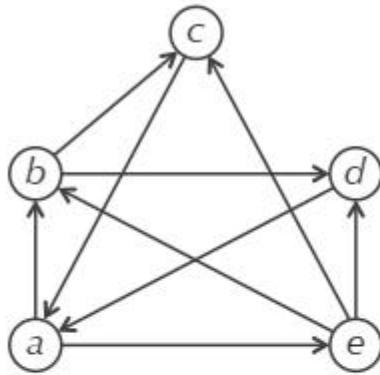


Come rappresentare tutti i nostri grafi in un computer?



I grafi si possono rappresentare come matrici o liste

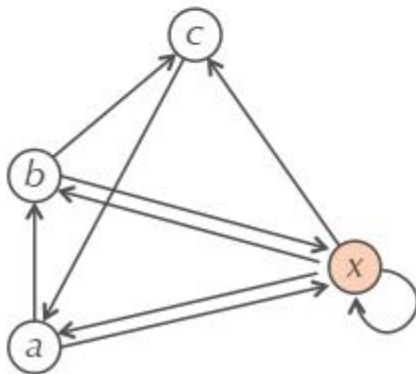
Graph



Adjacency List

a is adjacent to *b* and *e*
b is adjacent to *c* and *d*
c is adjacent to *a*
d is adjacent to *a*
e is adjacent to *b*, *c*, and *d*

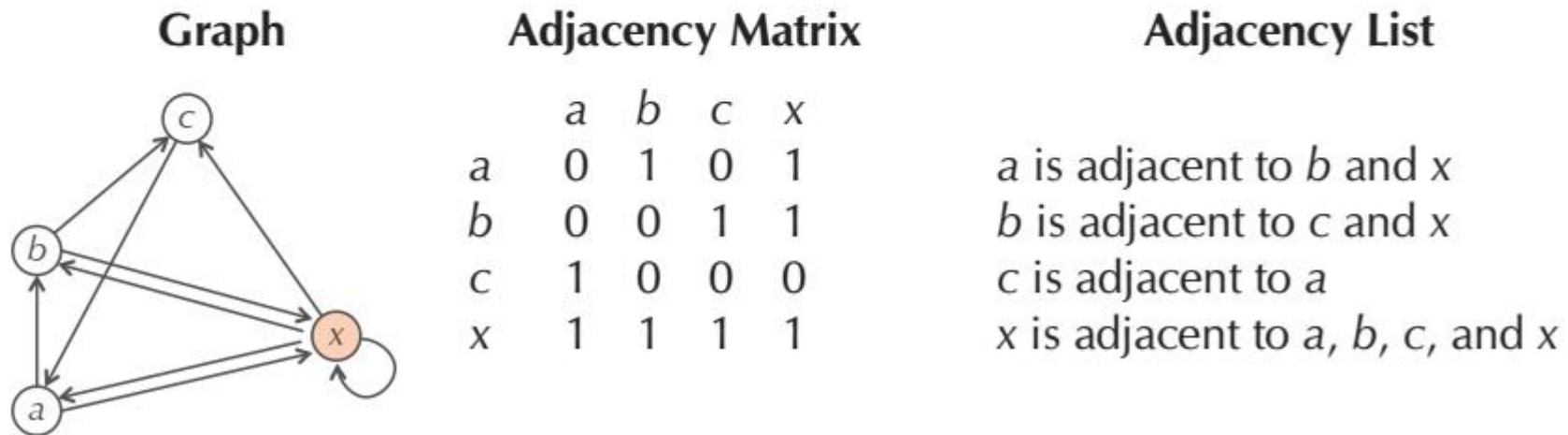
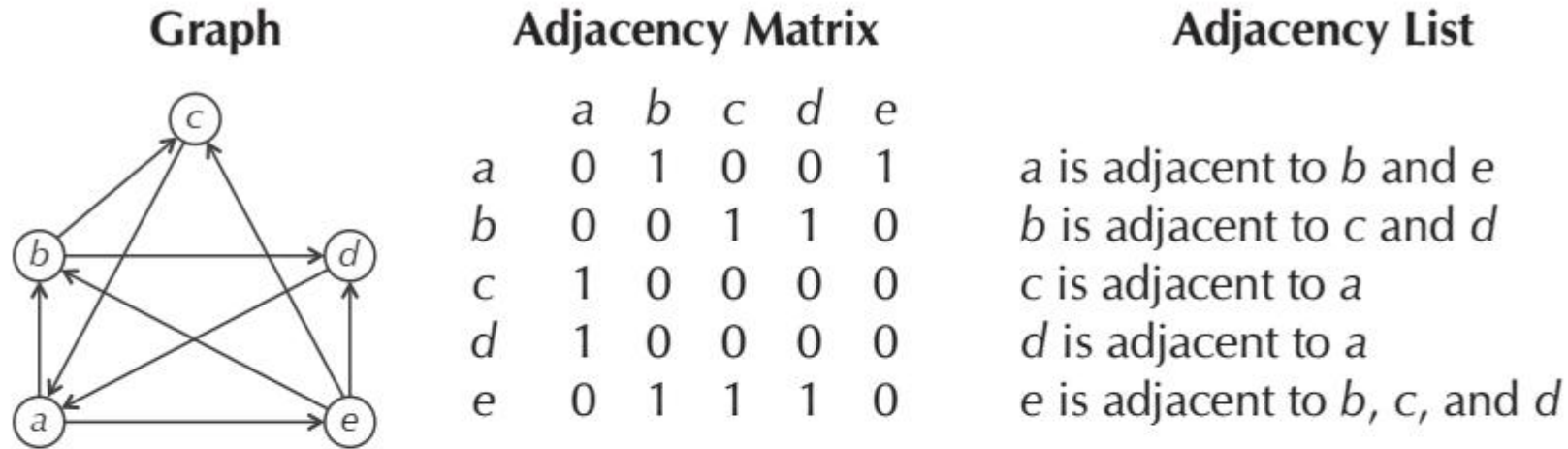
Graph



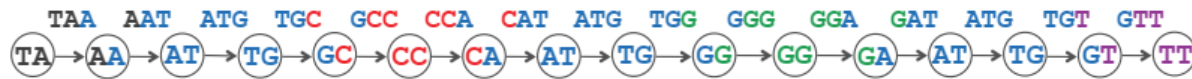
Adjacency List

a is adjacent to *b* and *x*
b is adjacent to *c* and *x*
c is adjacent to *a*
x is adjacent to *a*, *b*, *c*, and *x*

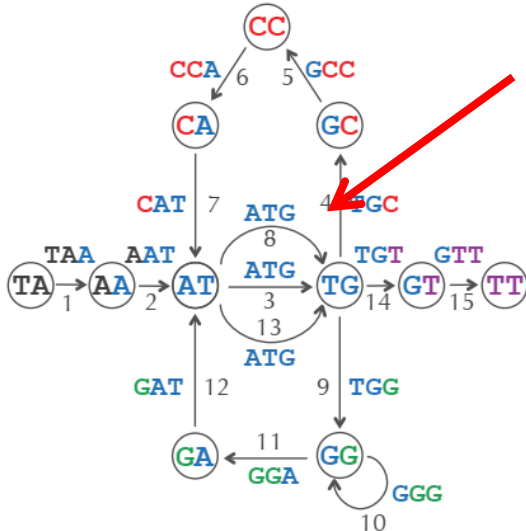
I grafi si possono rappresentare come matrici o liste



Le rappresentazioni matriciali possono anche rappresentare operazioni come «l'incollamento» dei grafi di De Bruijn

[illegible]

Le rappresentazioni matriciali possono anche rappresentare operazioni come «l'incollamento» dei grafi di De Bruijn

[illegible][illegible]

Cosa abbiamo la volta scorsa

- Perché è difficile assemblare
- Algoritmi per assemblare:
 - Overlap-graphs
 - De Bruijn graphs
- Concetti generali delle scienze computazionali (“Teoria dei grafi”, algoritmi polinomiali ed intrattabili)

Questa lezione

- Applicazioni specifiche a tipologie di dati di sequenziamento e problemi dei “genomi veri”
 - Il problema delle ripetizioni
 - Scaffolding
 - Assemblare con sequenze lunghe e corte e perché Overlap-graphs oggi tornano “di moda”
 - Valutazione degli assemblaggi
- Pratica di assemblaggio di genomi con strumenti bioinformatici (e cartoncini)