

INTRODUCTION TO (GENOME) ASSEMBLY

DAVIDE BOLOGNINI

POSTDOCTORAL RESEARCHER

DEVELOPING @ [HTTPS://GITHUB.COM/DAVIDEBOLO1993](https://github.com/DAVIDEBOLO1993)

DAVIDE.BOLOGNINI@UNIFI.IT

1. Background

2. Assembly: 1st & 2nd Laws

3. A Bioinformatic Example

4. The Shortest Common Superstring Problem

4.1. Greedy Shortest Common Superstring

5. Assembly: 3rd Law

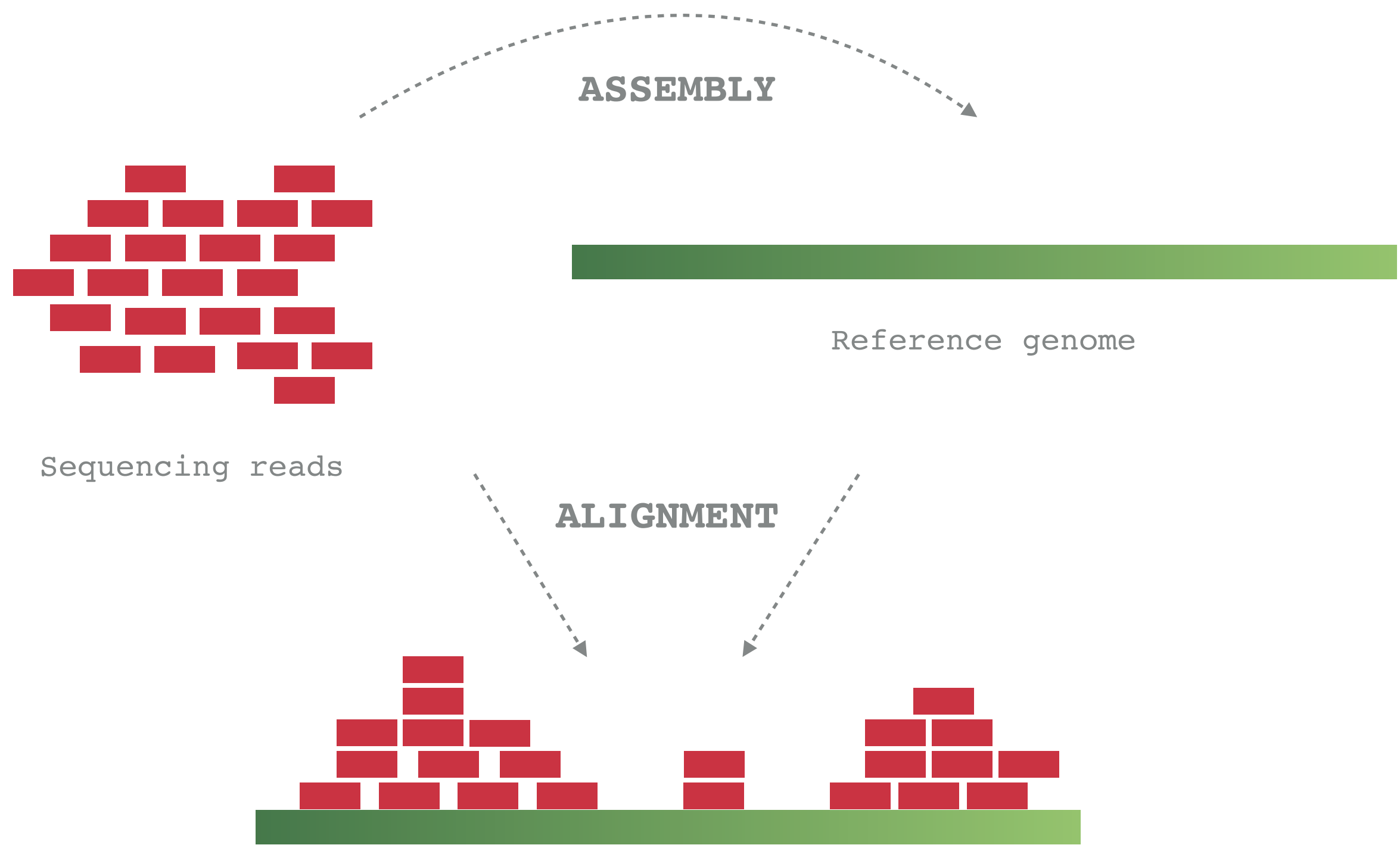
6. De Bruijn Graphs

6.1. Eulerian Walks

7. Assemblers In Practice

8. Teaching Material

1. BACKGROUND





The first human reference (2001)

Use these to ...
----->

CTCTAGGCCCTCAATTTTT CTAGGCCCTCAATTTTT
GGCTCTAGGCCCTCATTTTTT
CTCGGCTCTAGCCCCTCATTTT TATCTCGACTCTAGGCC
GGCGTCTATATCT TATCTCGACTCTAGGCCCTCA
TCTATATCTCGGCTCTAGG GGCGTCTATATCTCG GGCGTCGATATCT



... through this ...

Coverage=5 ←----- CTAGGCCCTCAATTTTT
CTCTAGGCCCTCAATTTTT
GGCTCTAGGCCCTCATTTTTT
CTCGGCTCTAGCCCCTCATTTT
TATCTCGACTCTAGGCCCTCA
TATCTCGACTCTAGGCC
TCTATATCTCGGCTCTAGG
GGCGTCTATATCTCG -----> Coverage=5
GGCGTCGATATCT
GGCGTCTATATCT



Overall coverage = 177 bp (reads) / 35 bp (reference) ~ 5 fold

??

... without knowing the final result
----->

... build this ...
----->

GGCGTCTATATCTCGGCTCTAGGCCCTCATTTTTT

2. ASSEMBLY: 1ST & 2ND LAWS

If a suffix of a read A is similar to a prefix of another read B, then A and B might overlap in the genome

A: T C T A T A T C T C G G C T C T A G G

B: T A T C T C G A C T C T A G G C C

Q: What happens here?

A: Sequencing errors or ploidy

A: T C T A T A T C T C G | G | C T C T A G G

... T C T A T A T C T C G : G : C T C T A G G C C ...

B: T A T C T C G A C T C T A G G C C

The more coverage we have, the more (and the longer) overlaps we get

TCTATATCTCGGCTCTAGG
TATCTCGACTCTAGGCC

TCTATATCTCG?CTCTAGGCC



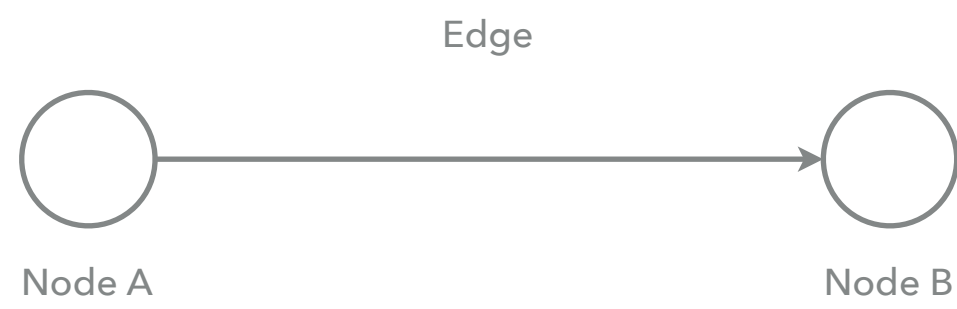
Low coverage
Few overlaps
Short (and inaccurate) reference

CTAGGCCCTCAATTTTT
CTCTAGGCCCTCAATTTTT
GGCTCTAGGCCCTCATTTTT
CTCGGCTCTAGCCCCTCATTTT
TATCTCGACTCTAGGCCCTCA
TATCTCGACTCTAGGCC
TCTATATCTCGGCTCTAGG
GGCGTCTATATCTCG
GGCGTCGATATCT
GGCGTCTATATCT
GGCGTCTATATCTCGGCTCTAGGCCCTCATTTTTT

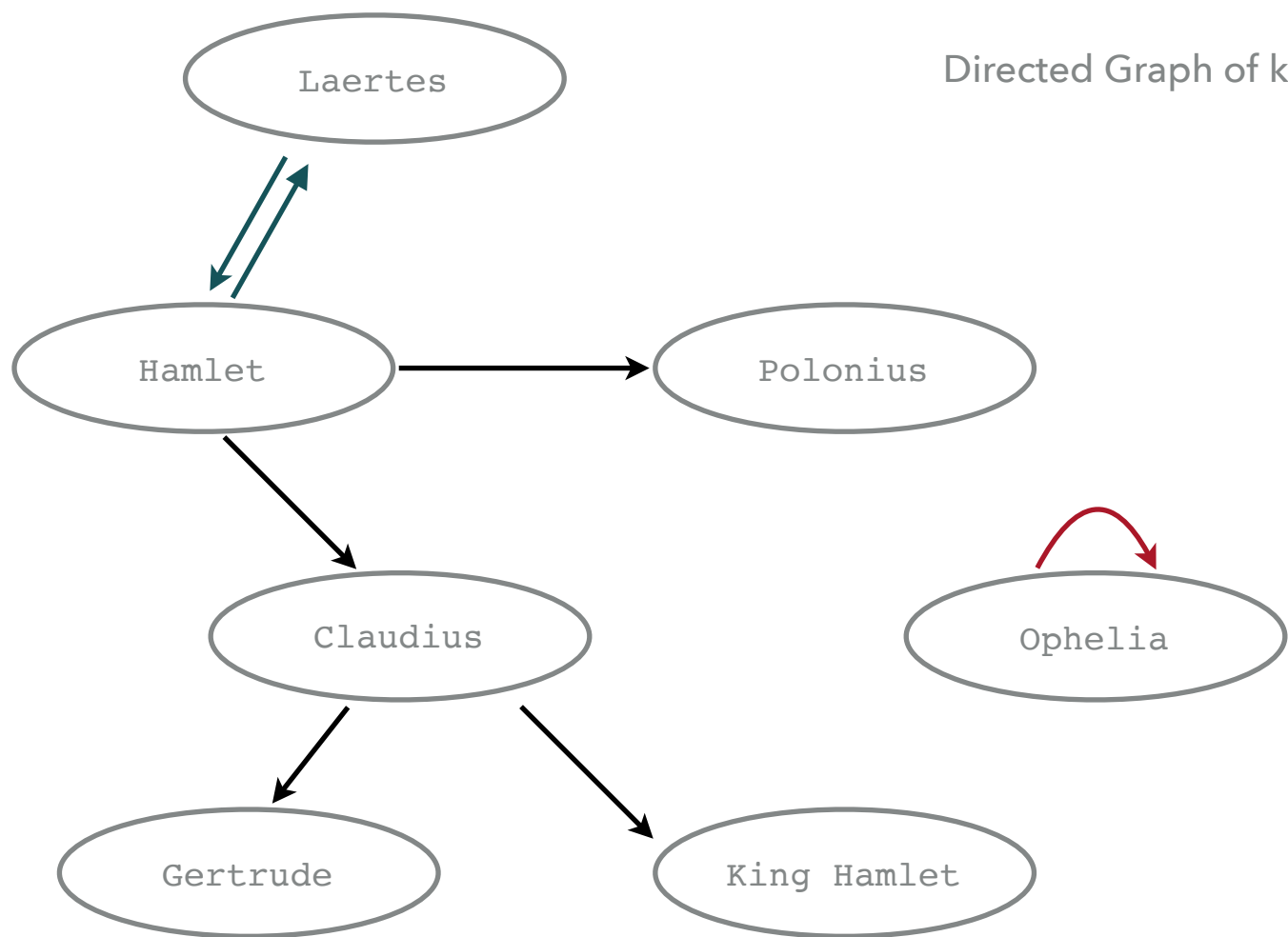


Higher coverage
More overlaps
Longer (and more accurate) reference

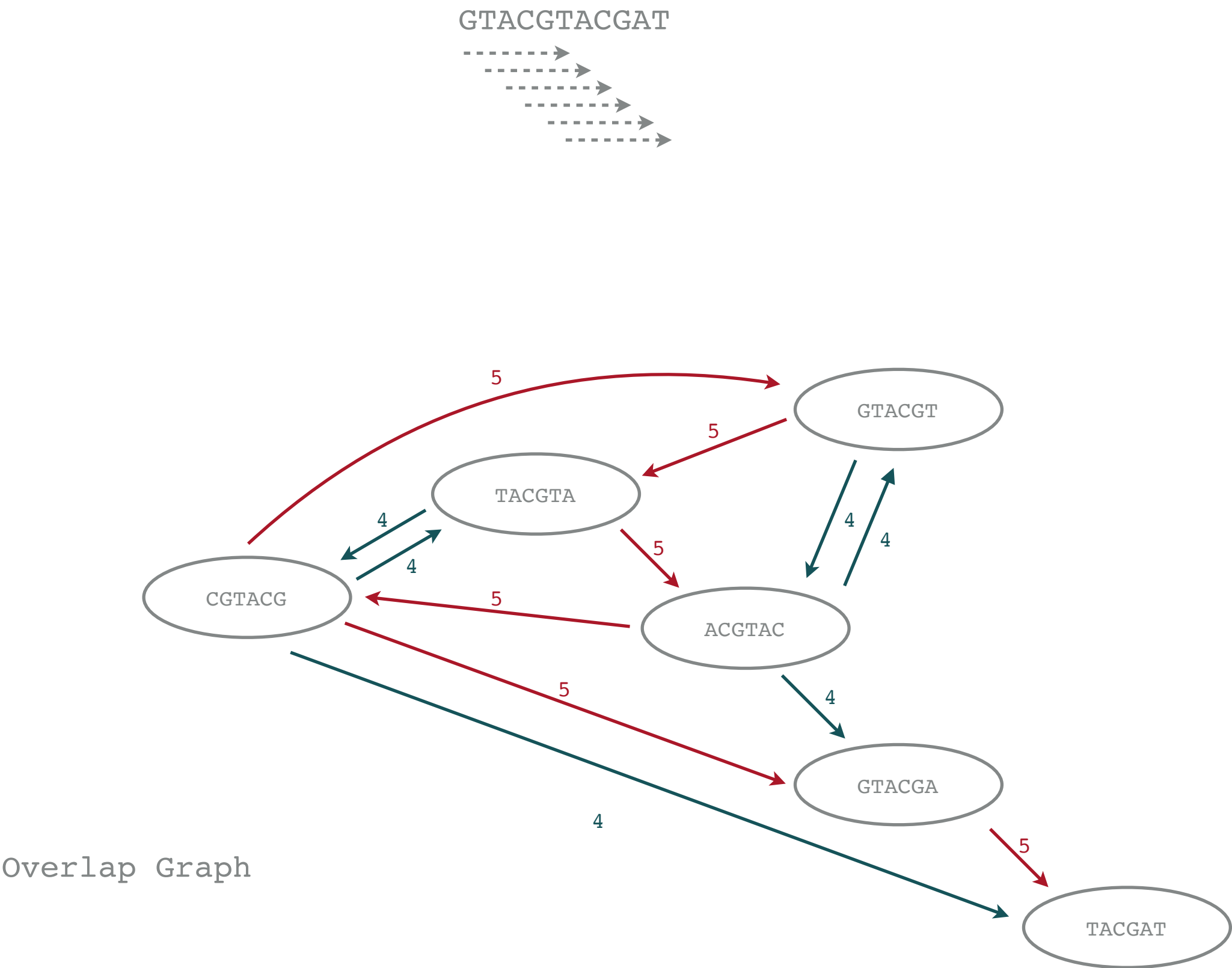
Q: How do we represent the overlaps?



Directed Graph



Directed Graph of kills in Hamlet



3. A BIOINFORMATIC EXAMPLE

```
In [18]: def Overlap(a,b,min_length=3):  
    '''  
    If a suffix of a overlaps a prefix of b, that is at least min_length characters long,  
    then return the length of the longest suffix. If not, return 0  
    '''  
  
    start=0  
    while True:  
        start=a.find(b[:min_length],start)  
        if start == -1:  
            return 0  
        if b.startswith(a[start:]):  
            return len(a)-start  
        start+=1
```

```
In [19]: from itertools import permutations  
  
def BruteOverlap(reads, k):  
    '''  
    Return overlaps between pair of reads  
    '''  
  
    olaps={}  
    for a,b in permutations(reads,2):  
        olen=Overlap(a,b,min_length=k)  
        if olen > 0:  
            olaps[(a,b)] = olen  
    return olaps
```

```
In [20]: reads=[ 'GTACGT', 'TACGTA', 'ACGTAC', 'CGTACG', 'GTACGA', 'TACGAT' ]  
          BruteOverlap(reads,4)
```

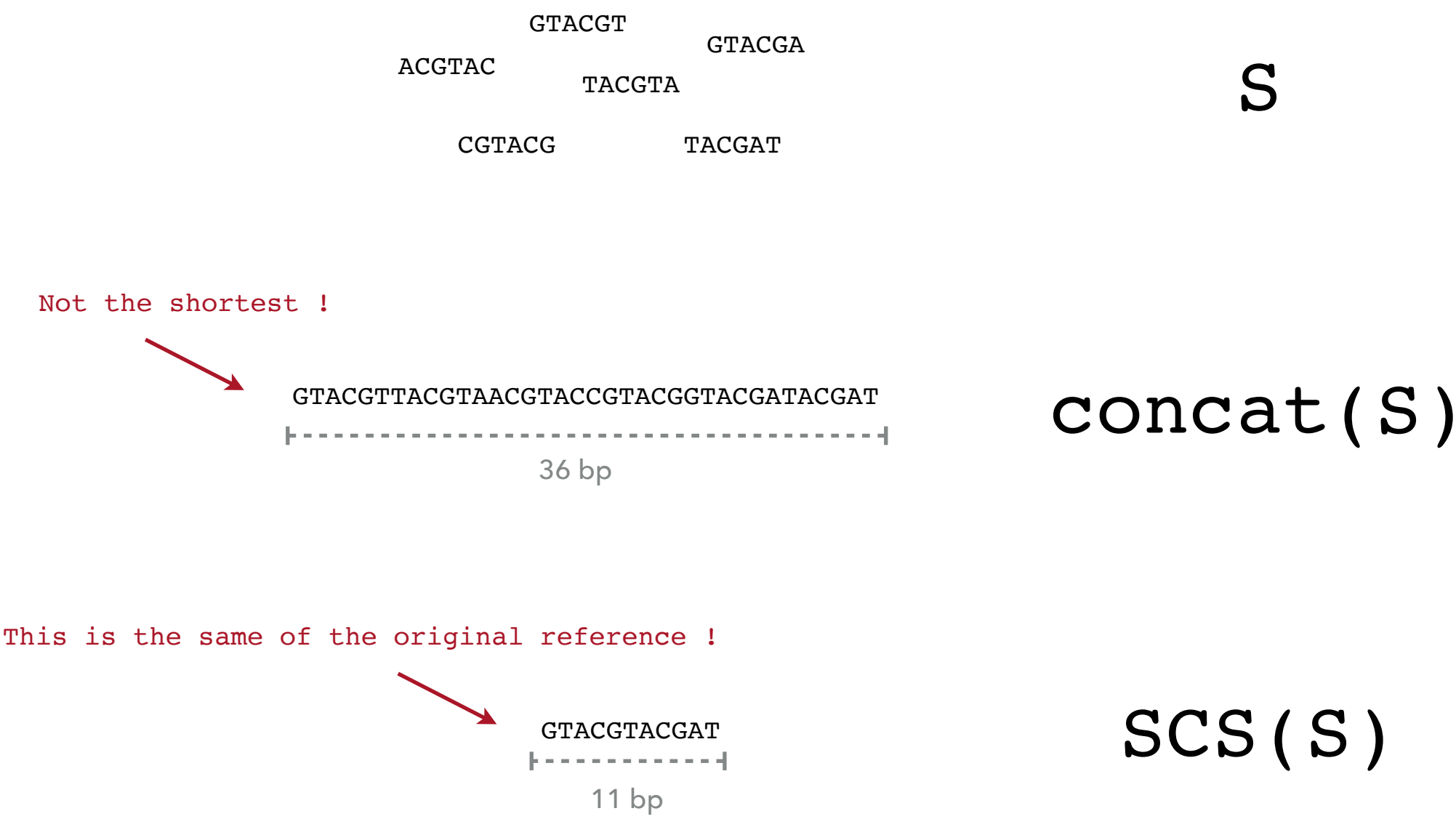
```
{('GTACGT', 'TACGTA'): 5,  
 ('GTACGT', 'ACGTAC'): 4,  
 ('TACGTA', 'ACGTAC'): 5,  
 ('TACGTA', 'CGTACG'): 4,  
 ('ACGTAC', 'GTACGT'): 4,  
 ('ACGTAC', 'CGTACG'): 5,  
 ('ACGTAC', 'GTACGA'): 4,  
 ('CGTACG', 'GTACGT'): 5,  
 ('CGTACG', 'TACGTA'): 4,  
 ('CGTACG', 'GTACGA'): 5,  
 ('CGTACG', 'TACGAT'): 4,  
 ('GTACGA', 'TACGAT'): 5}
```

Q: How to reconstruct the reference genome from these overlaps

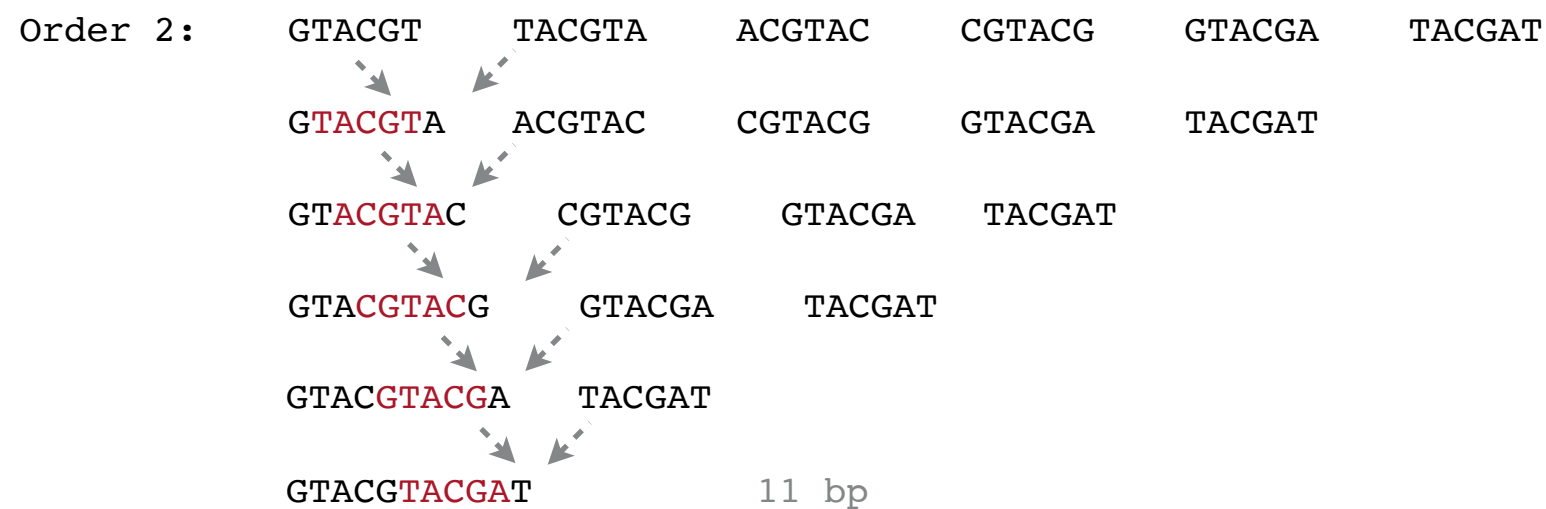
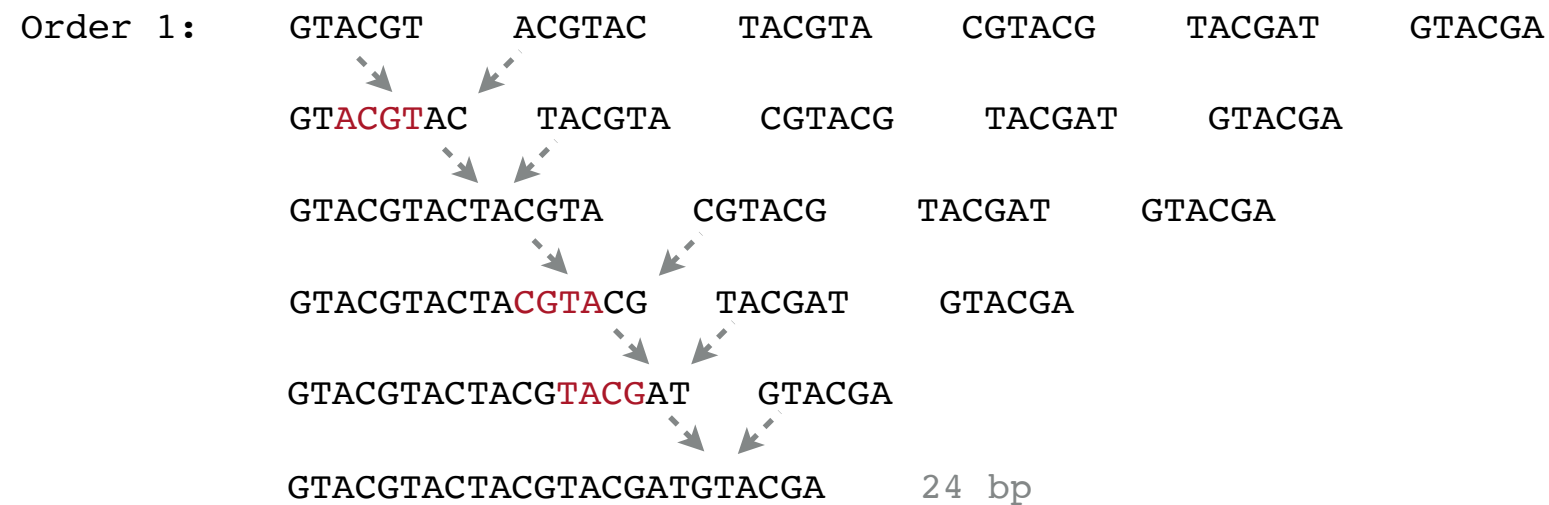
A: Find a superstring that contains all the overlapping strings

4. THE SHORTEST COMMON SUPERSTRING PROBLEM

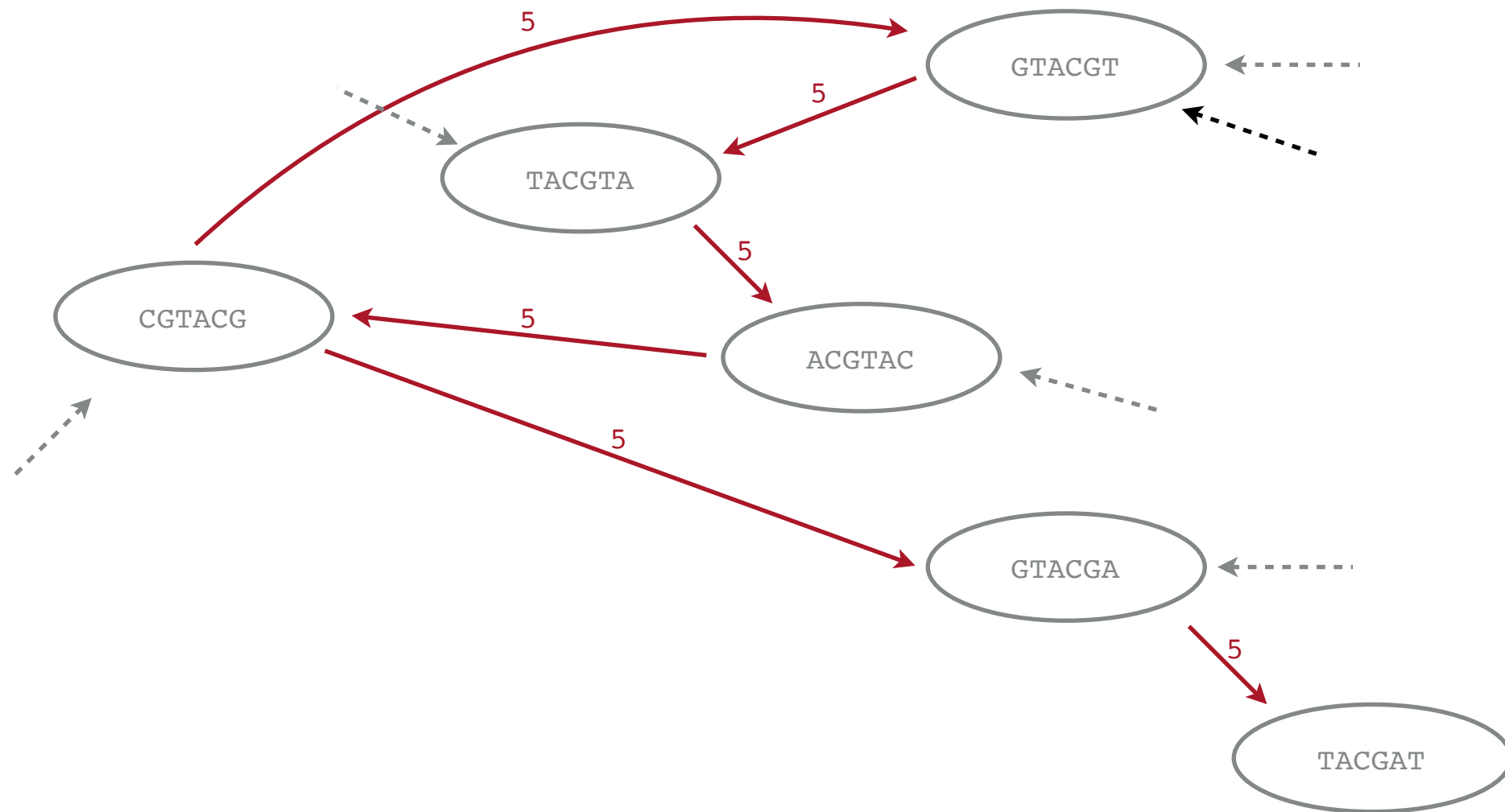
Given a set of string S , the Shortest Common Superstring (SCS) is the shortest string containing strings in S as substrings



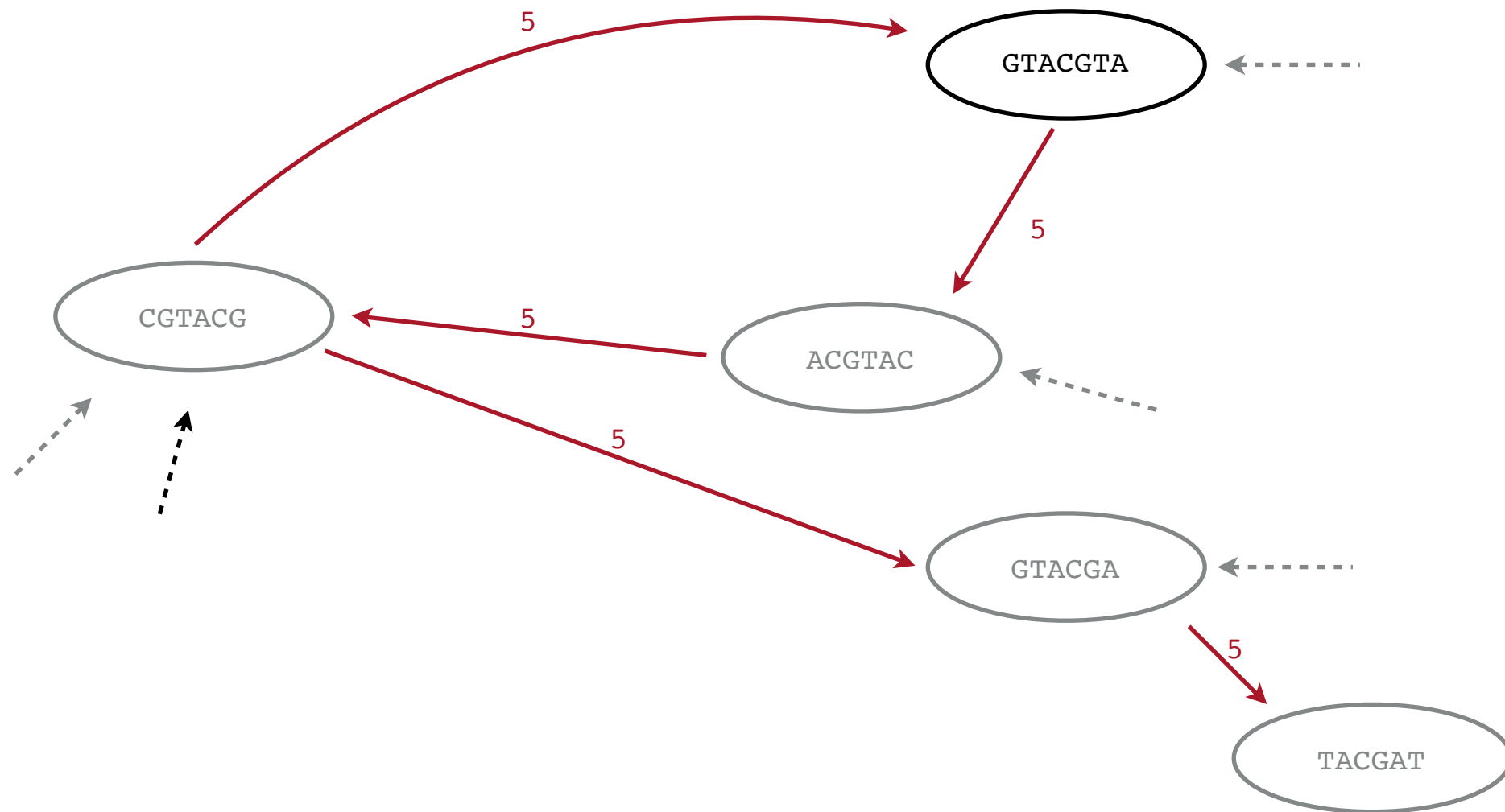
The SCS problem is *de facto* an assembly problem, but SCS is NP-complete



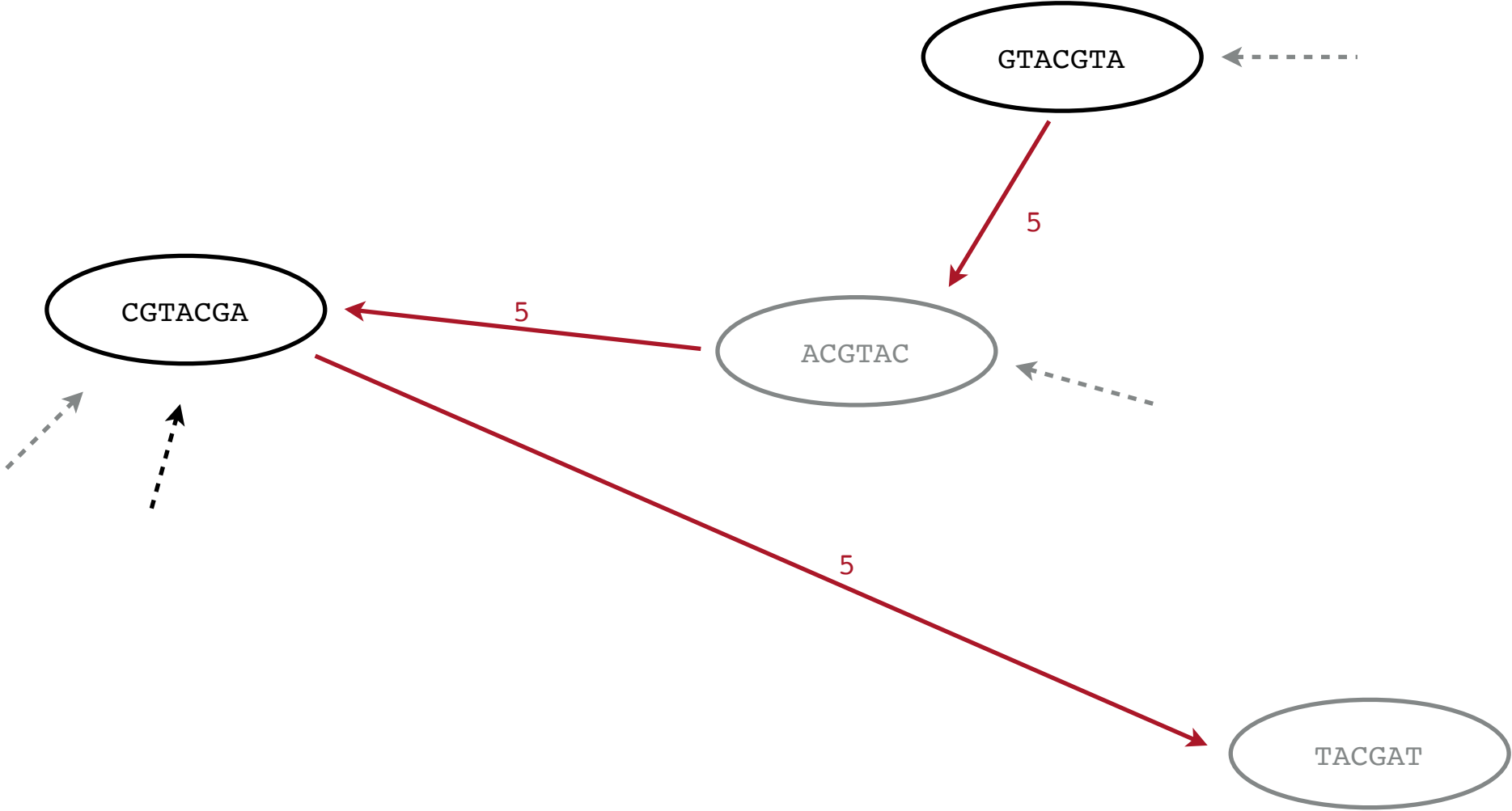
In order to find the SCS one has to try all the possible orderings.
With N strings, this equals to $N!$ orders

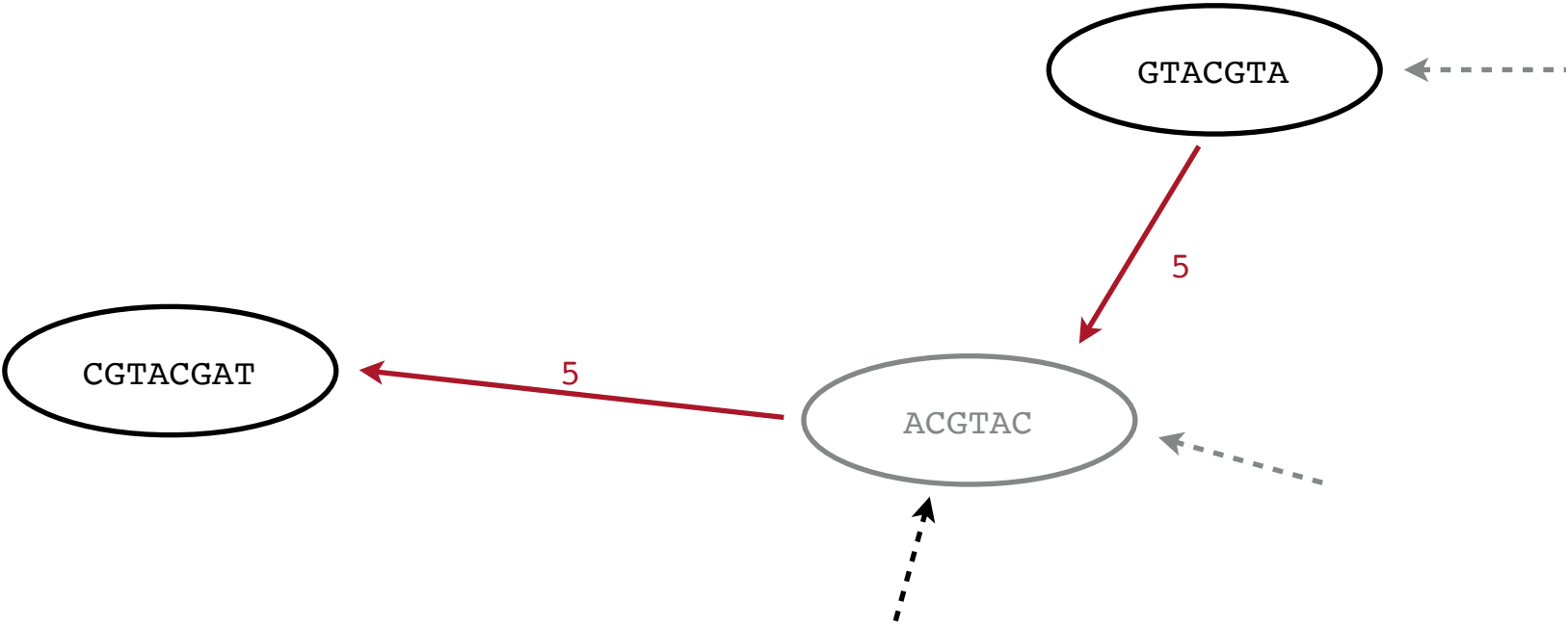


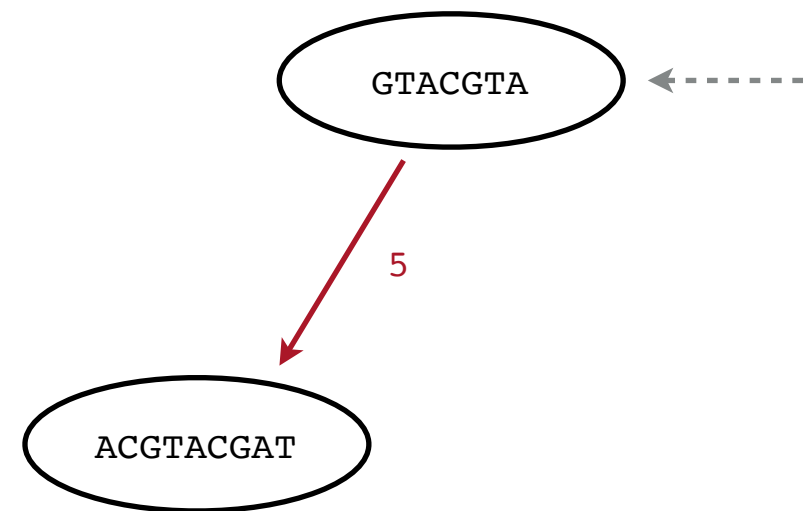
Start from the node which has the maximum overlap with another. If multiple maximum values, choose one node randomly. Merge the chosen nodes

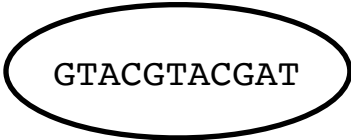


Repeat the previous step until all nodes have been processed









GTACGTACGAT

The last node corresponds to the expected SCS

This is not always true. It might happen that the SCS algorithm does not find the SCS but a slightly longer superstring. This is because nodes are chosen random when they have the same weight

5. ASSEMBLY: 3RD LAW

Repeated regions make assembly difficult

```

a_long_long_long_time
a_long
 _long_ ←-----
long_l  ←-----
ong_lo  ←-----
ng_lon  ←-----
g_long  ←-----
 _long_ ←-----
long_l  ←-----
ong_lo  ←-----
ng_lon  ←-----
g_long  ←-----
 _long_ ←-----
long_t
ong_ti
ng_tim
g_time

```

```
SCS(a_long_long_long_time) = a_long_long_time
```

Repetitive Elements May Comprise Over Two-Thirds of the Human Genome

A. P. Jason de Koning¹, Wanjun Gu^{1*}, Todd A. Castoe¹, Mark A. Batzer², David D. Pollock^{1*}

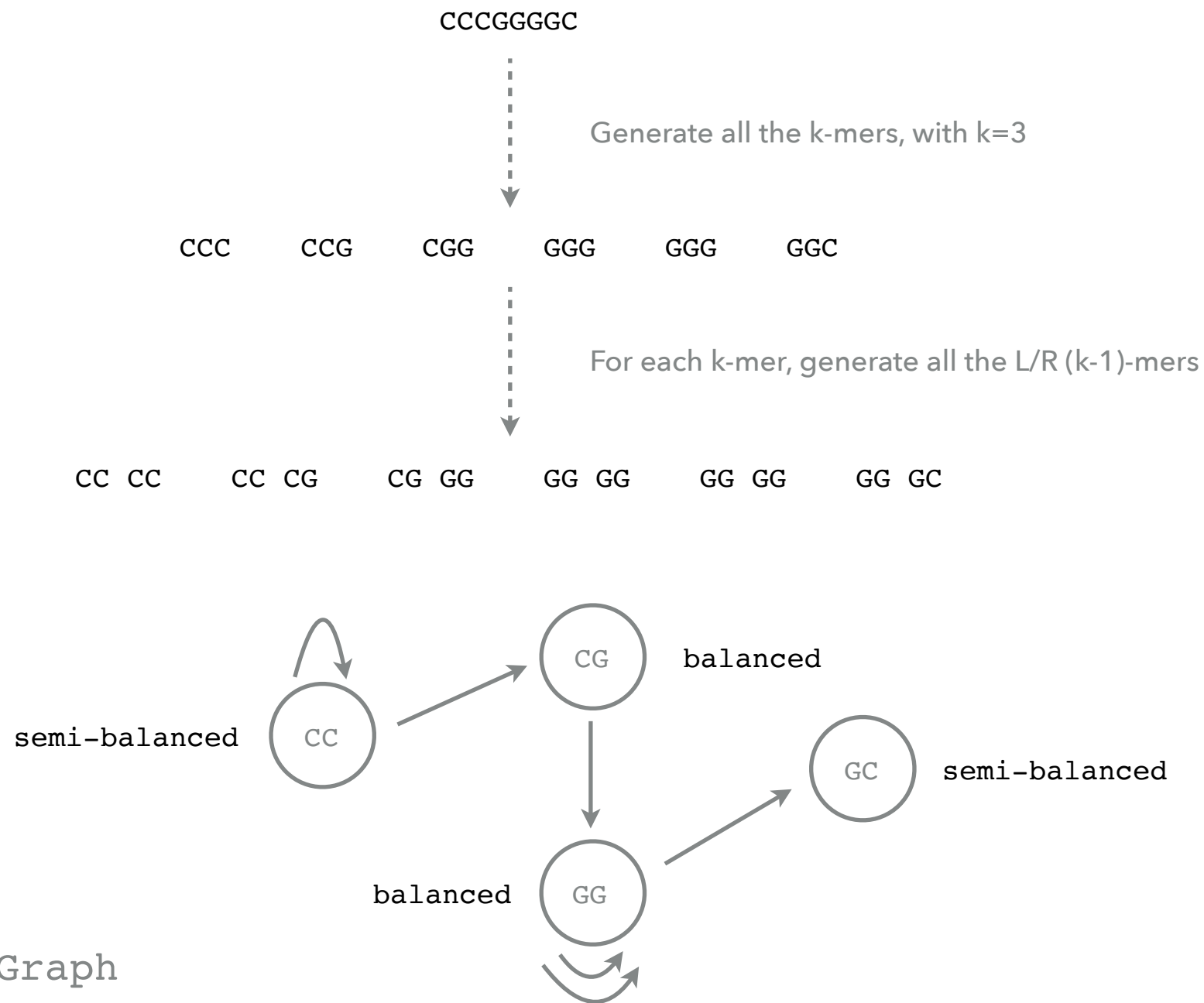
¹ Department of Biochemistry and Molecular Genetics, School of Medicine, University of Colorado, Aurora, Colorado, United States of America, ² Department of Biological Sciences, Louisiana State University, Baton Rouge, Louisiana, United States of America

Abstract

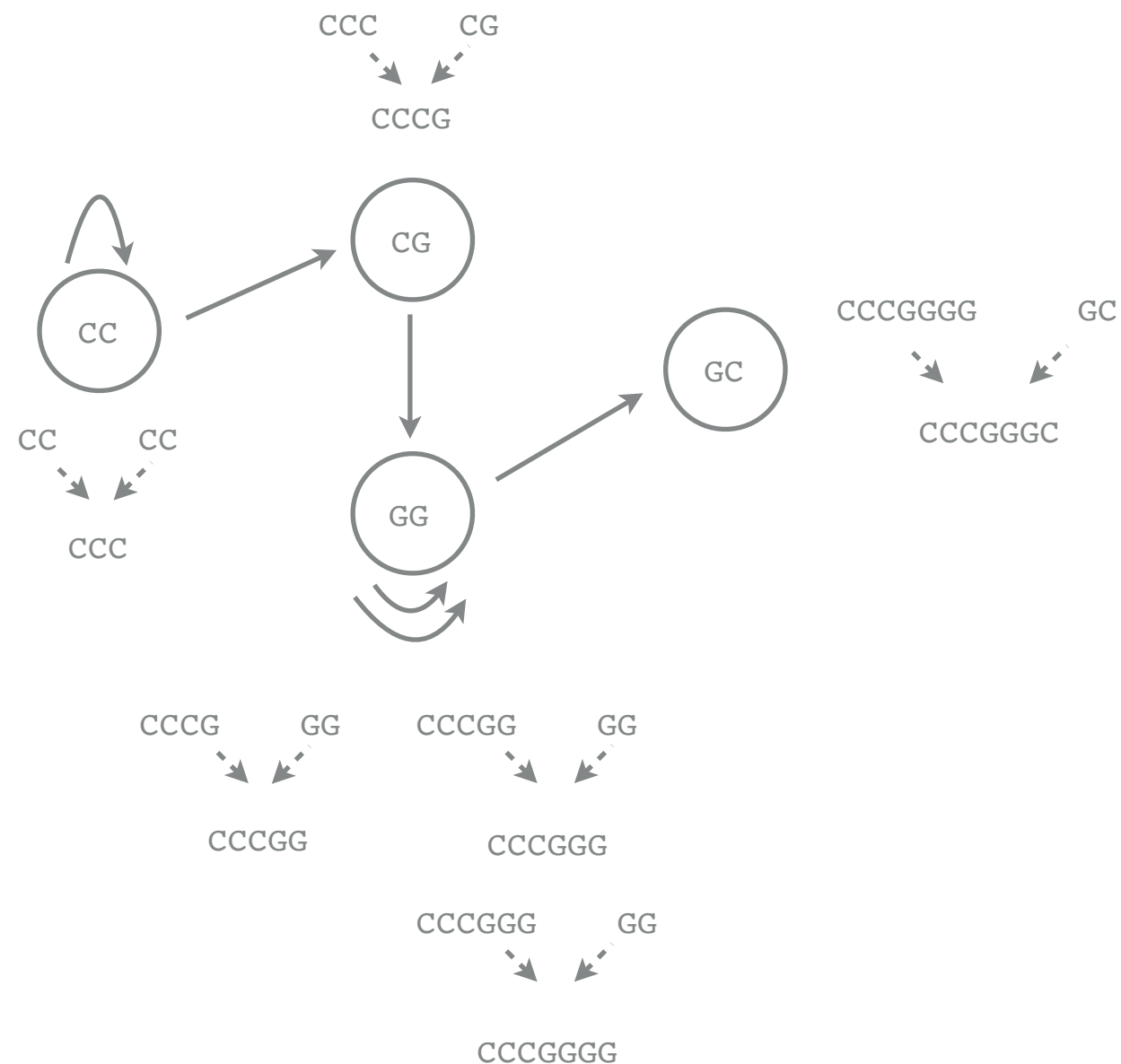
Transposable elements (TEs) are conventionally identified in eukaryotic genomes by alignment to consensus element sequences. Using this approach, about half of the human genome has been previously identified as TEs and low-complexity repeats. We recently developed a highly sensitive alternative *de novo* strategy, *P-clouds*, that instead searches for clusters of high-abundance oligonucleotides that are related in sequence space (oligo “clouds”). We show here that *P-clouds* predicts >840 Mbp of additional repetitive sequences in the human genome, thus suggesting that 66%–69% of the human genome is repetitive or repeat-derived. To investigate this remarkable difference, we conducted detailed analyses of the ability of both *P-clouds* and a commonly used conventional approach, *RepeatMasker* (RM), to detect different sized fragments of the highly abundant human Alu and MIR SINEs. RM can have surprisingly low sensitivity for even moderately long fragments, in contrast to *P-clouds*, which has good sensitivity down to small fragment sizes (~25 bp). Although short fragments have a high intrinsic probability of being false positives, we performed a probabilistic annotation that reflects this fact. We further developed “element-specific” *P-clouds* (ESPs) to identify novel Alu and MIR SINE elements, and using it we identified ~100 Mb of previously unannotated human elements. ESP estimates of new *MIR* sequences are in good agreement with RM-based predictions of the amount that RM missed. These results highlight the need for combined, probabilistic genome annotation approaches and suggest that the human genome consists of substantially more repetitive sequence than previously believed.

SCS is likely to collapse repetitive regions, which are abundant in genomes.
We need a method that avoids the over-collapsing problem

6. DE BRUIJN GRAPHS

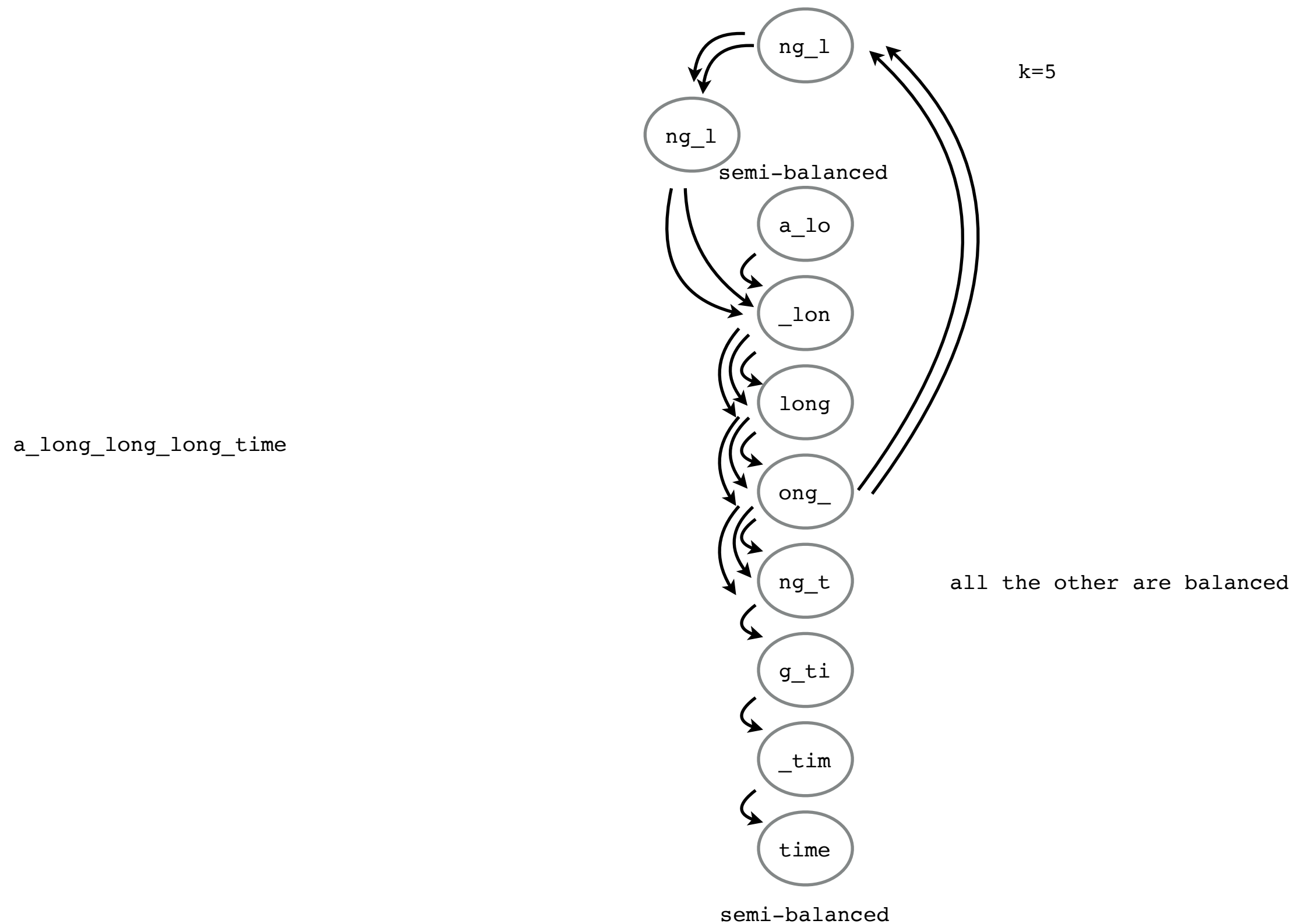


Nodes in De Bruijn Graphs correspond to unique L/R $(k-1)$ -mers. There is an edge for each original k -mer. A node is balanced if $\text{indegree} == \text{outdegree}$, semi-balanced if $|\text{outdegree} - \text{indegree}| == 1$



Find the path that crosses each edge exactly once

Not all the graphs have Eulerian walks. A graph that does is called Eulerian (graph). A De Bruijn Graph will always be Eulerian if it has at most 2 semi-balanced nodes

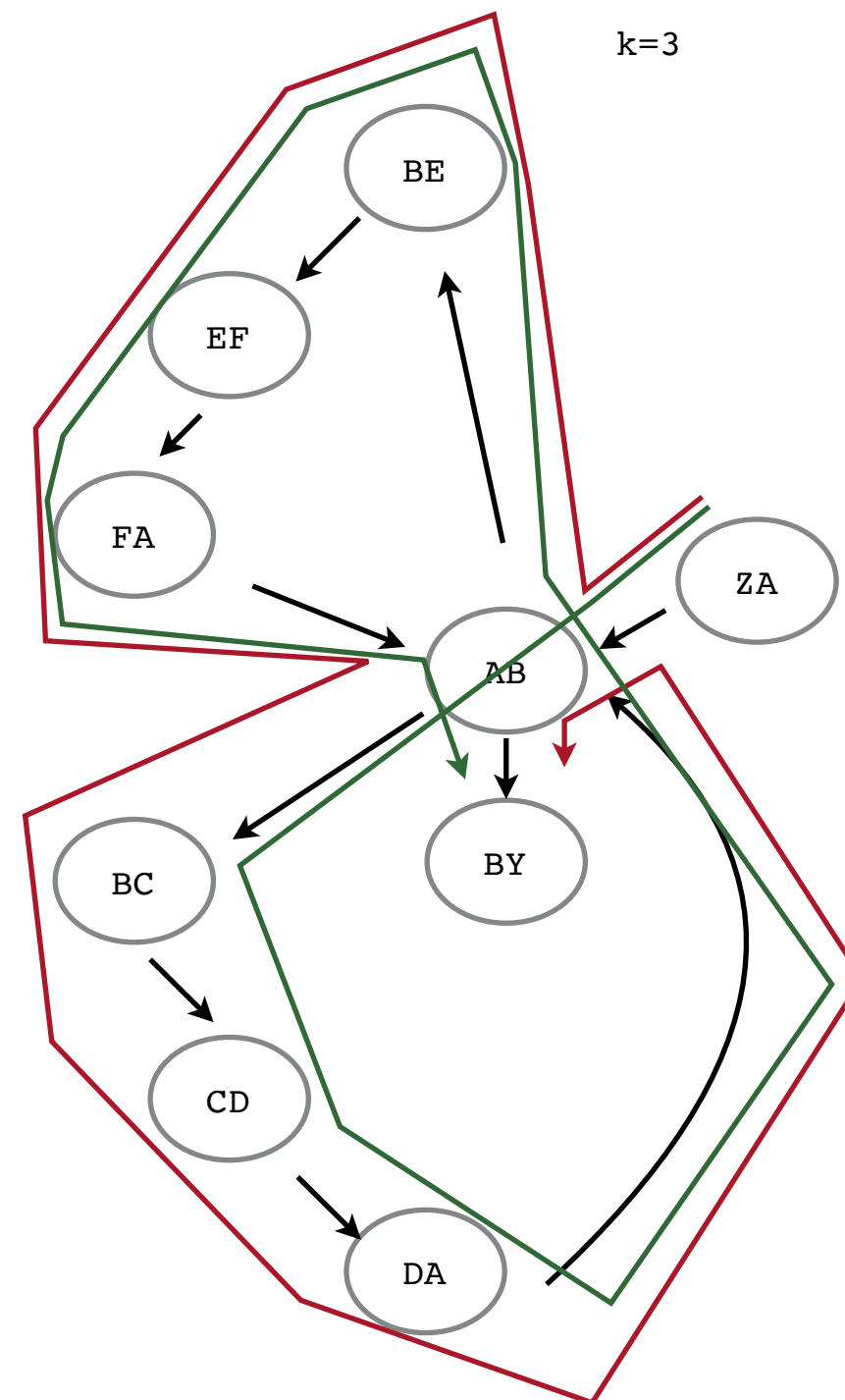


The Eulerian walk for this graph correctly reconstructs the input sequence. Is this always true?

ZABCDABEFABY

ZABCDABEFABY

ZABEFABCDABY



Repeats still complicate the reconstruction of the original sequence through the Eulerian walk. Larger k-mers may help in this sense. Solving issues with uneven depth of coverage and sequencing errors remains challenging

7. ASSEMBLERS IN PRACTICE

Comparison of the two major classes of assembly algorithms: overlap–layout–consensus and de-bruijn–graph FREE

Zhenyu Li, Yanxiang Chen, Desheng Mu, Jianying Yuan, Yujian Shi, Hao Zhang, Jun Gan, Nan Li, Xuesong Hu, Binghang Liu ... [Show more](#)

[Author Notes](#)

Briefings in Functional Genomics, Volume 11, Issue 1, January 2012, Pages 25–37,
<https://doi.org/10.1093/bfgp/elr035>

Published: 19 December 2011



PDF

Split View

Cite



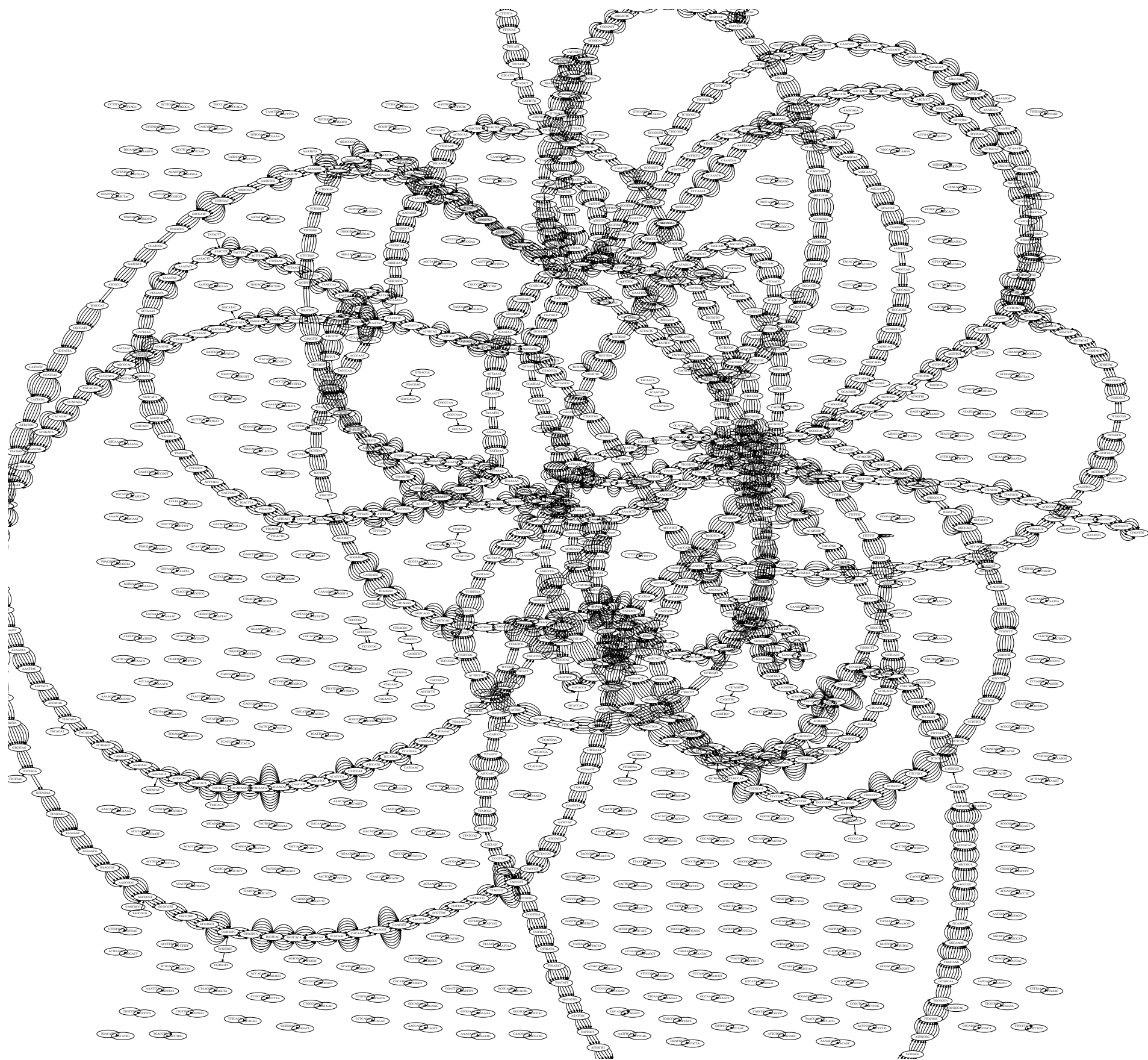
Permissions

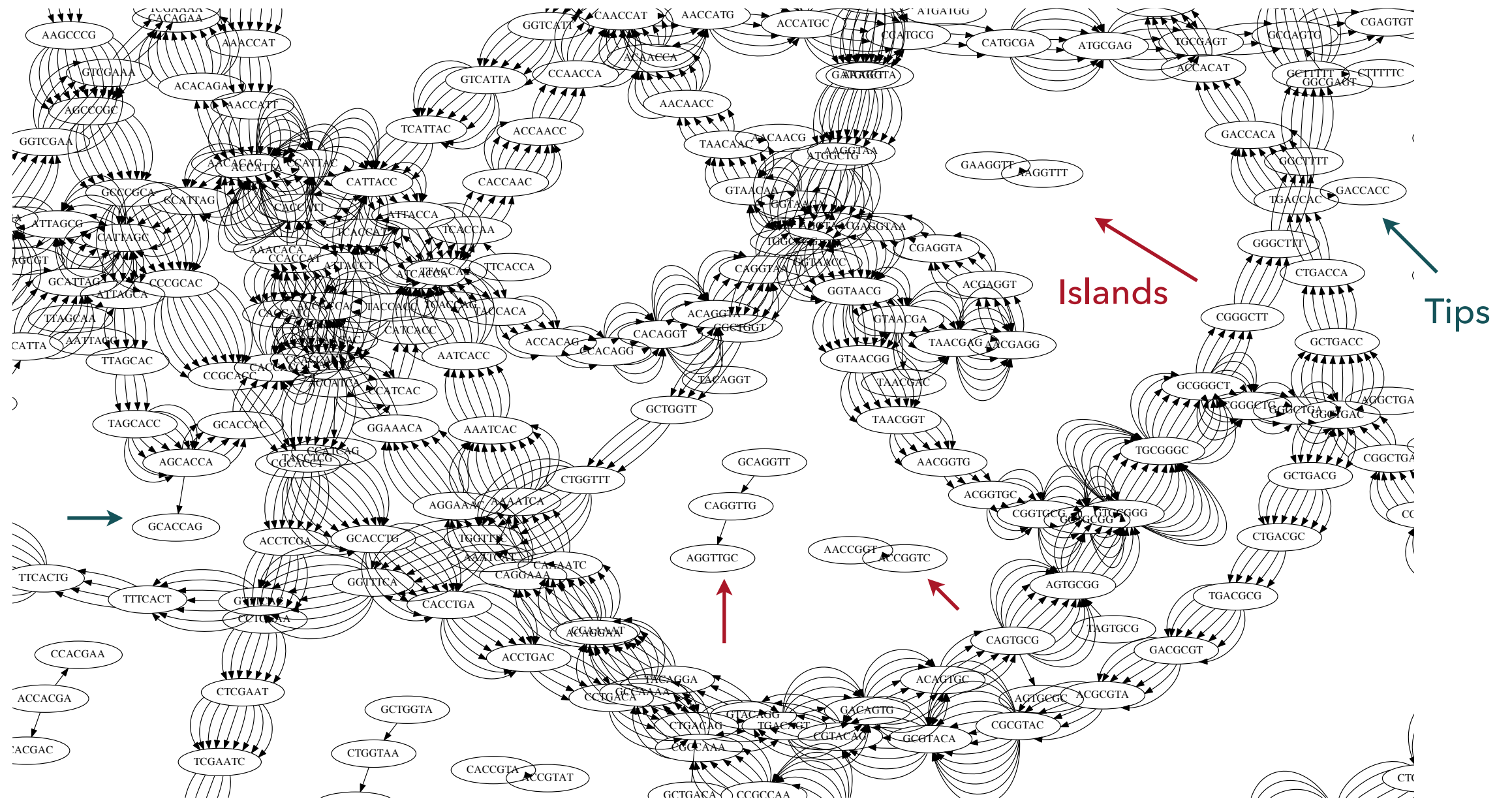


Share ▼

Abstract

Since the completion of the cucumber and panda genome projects using Illumina sequencing in 2009, the global scientific community has had to pay much more attention to this new cost-effective approach to generate the draft sequence of large genomes. To allow new users to more easily understand the assembly algorithms and the optimum software packages for their projects, we make a detailed comparison of the two major classes of assembly algorithms: **overlap–layout–consensus** and **de-bruijn–graph** from how they match the Lander–Waterman model, to the required sequencing depth and reads length. We also discuss the computational efficiency of each class of algorithm, the influence of repeats and heterozygosity and points of note in the subsequent scaffold linkage and gap closure steps. We hope this review can help further promote the application of second-generation *de novo* sequencing, as well as aid the future development of assembly algorithms.





Sequencing errors turn frequent k-mers into infrequent k-mers

GCGTATTACGCGTCTGGCCT

- GCGTATTA, 8
- CGTATTAC, 8
- GTATTACG, 10
- TATTACGC, 10
- ATTACGCG, 10
- TTACGCGT, 9
- TACGCGTC, 10
- ACGCGTCT, 8
- CGCGTCTG, 11
- GCGTCTGG, 8
- CGTCTGGC, 9
- GTCTGGCC, 11
- TCTGGCCT, 10

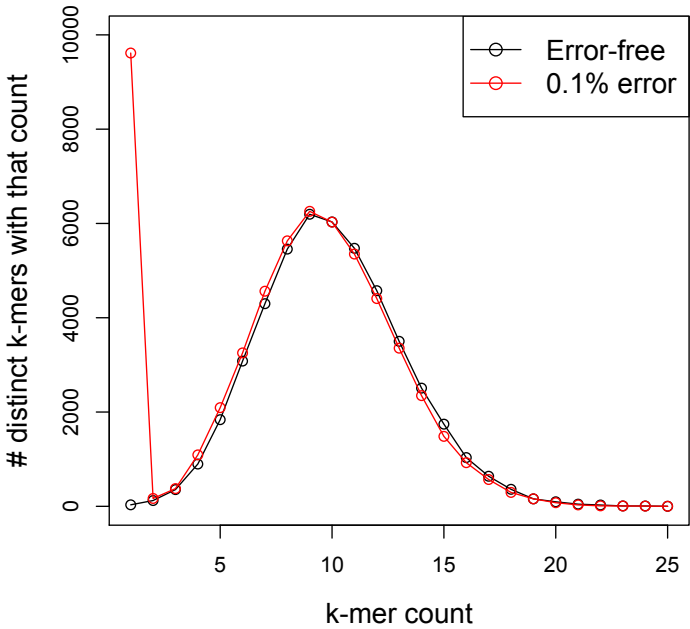
Error-free

GCGTACTACGCGTCTGGCCT

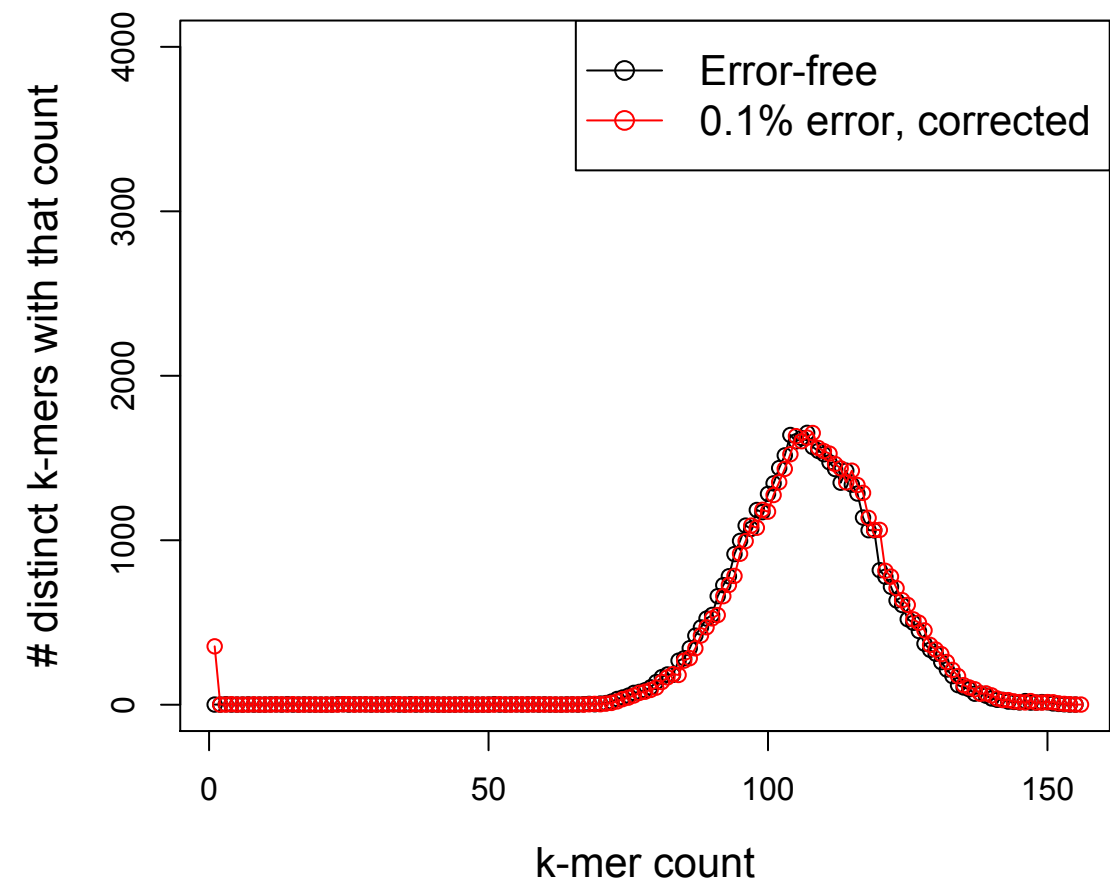
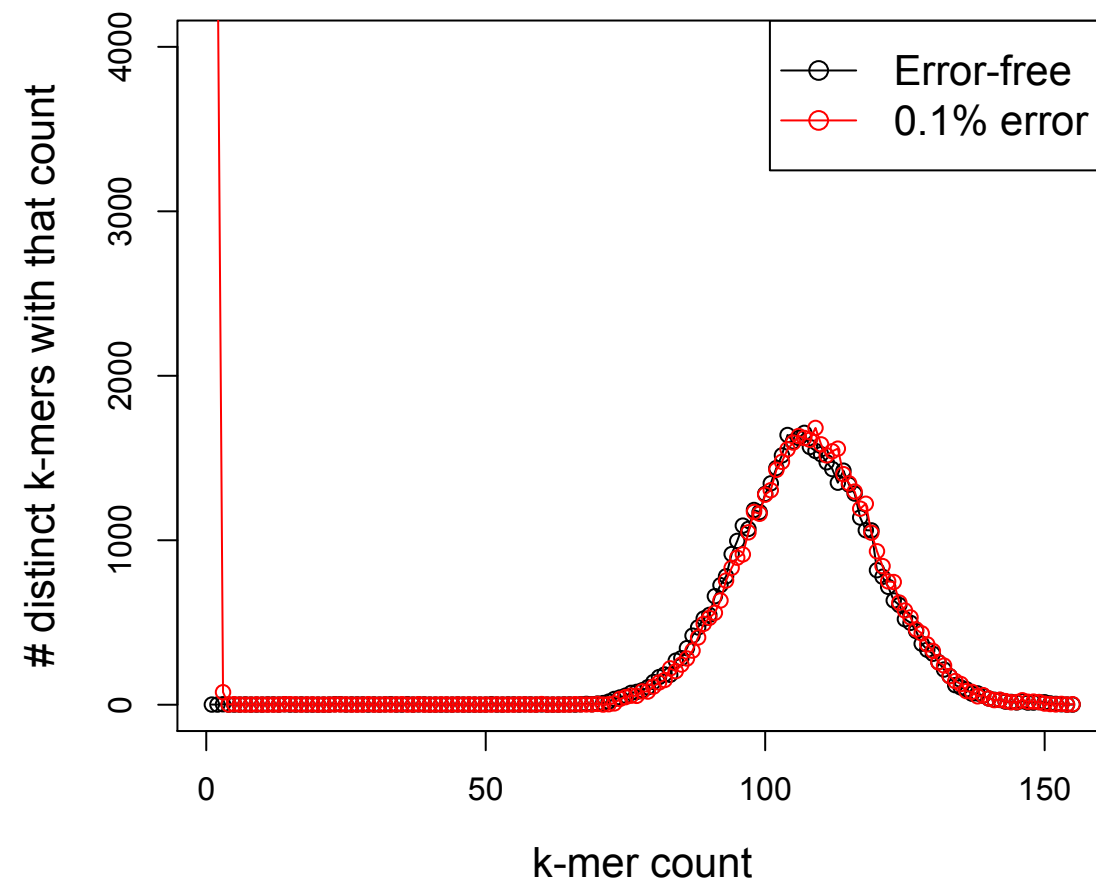
- GCGTACTA, 1
- CGTACTAC, 2
- GTACTACG, 1
- TACTACGC, 1
- ACTACGCG, 2
- CTACGCGT, 1
- TACGCGTC, 10
- ACGCGTCT, 8
- CGCGTCTG, 11
- GCGTCTGG, 8
- CGTCTGGC, 9
- GTCTGGCC, 11
- TCTGGCCT, 10

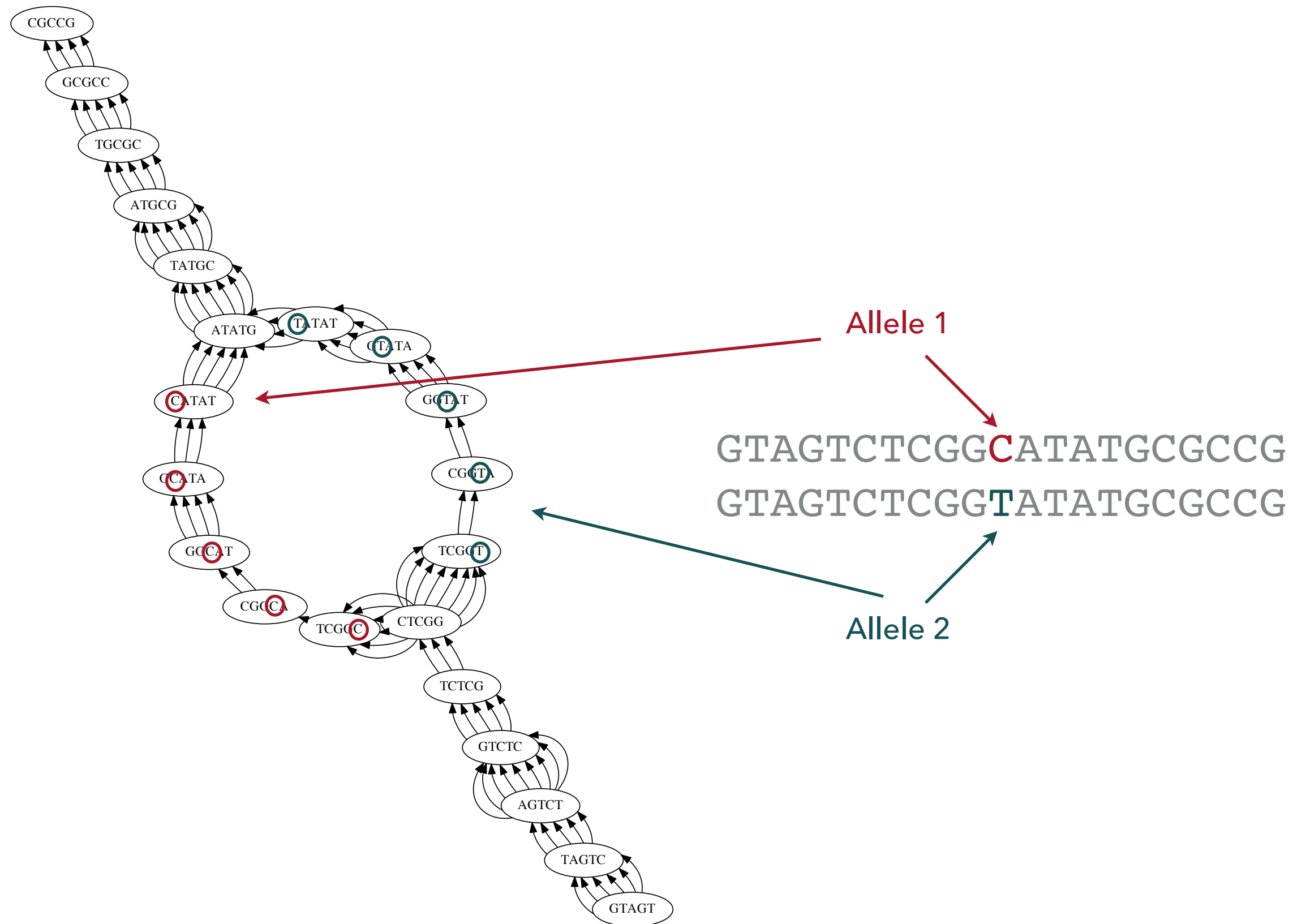
Error

Assume ~10 fold coverage



For each read and each k-mer, if a k-mer occurs less than t times, replace this k-mer with a more frequent hamming/edit-distance neighbor





nature


View all Nature Research journals

Search  Login Explore our content Journal information 

nature > letters > article

Published: 10 November 2014

Resolving the complexity of the human genome using single-molecule sequencing

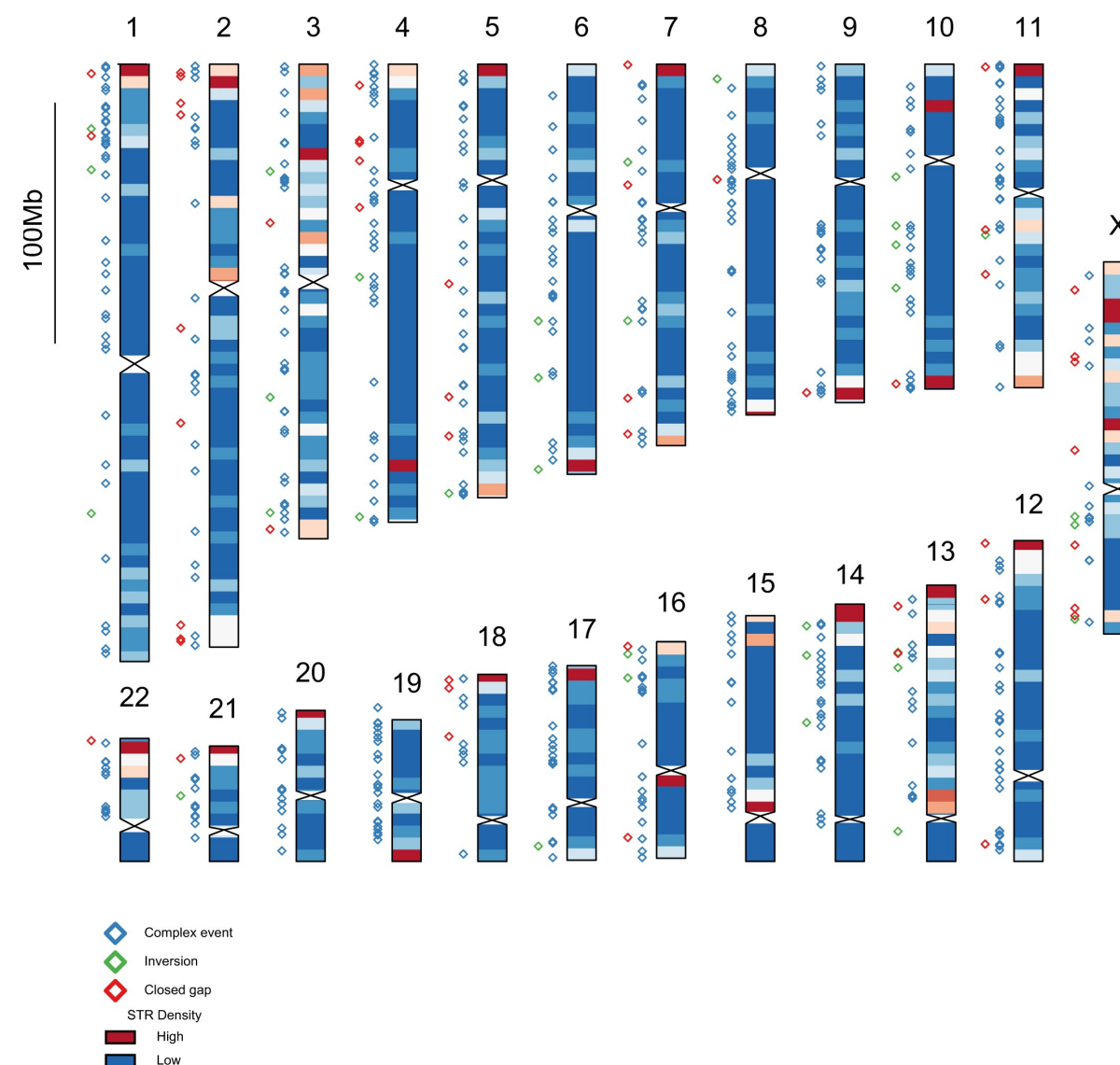
Mark J. P. Chaisson, John Huddleston, Megan Y. Dennis, Peter H. Sudmant, Maika Malig, Fereydoun Hormozdiari, Francesca Antonacci, Urvashi Surti, Richard Sandstrom, Matthew Boitano, Jane M. Landolin, John A. Stamatoyannopoulos, Michael W. Hunkapiller, Jonas Koriach & Evan E. Eichler 

Nature **517**, 608–611(2015) | [Cite this article](#)

2958 Accesses | **355** Citations | **344** Altmetric | [Metrics](#)

Abstract

The human genome is arguably the most complete mammalian reference assembly^{1,2,3}, yet more than 160 euchromatic gaps remain^{4,5,6} and aspects of its structural variation remain poorly understood ten years after its completion^{7,8,9}. To identify missing sequence and genetic variation, here we sequence and analyse a haploid human genome (CHM1) using single-molecule, real-time DNA sequencing¹⁰. We close or extend 55% of the remaining interstitial gaps in the human GRCh37 reference genome—78% of which carried long runs of degenerate short tandem repeats, often several kilobases in length, embedded within (G+C)-rich genomic regions. We resolve the complete sequence of 26,079 euchromatic structural variants at the base-pair level, including inversions, complex insertions and long tracts of tandem repeats. Most have not been previously reported, with the greatest increases in sensitivity occurring for events less than 5 kilobases in size. Compared to the human reference, we find a significant insertional bias (3:1) in regions corresponding to complex insertions and long short tandem repeats. Our results suggest a greater complexity of the human genome in the form of variation of longer and more complex repetitive DNA that can now be largely resolved with the application of this longer-read sequencing technology.



8. TEACHING MATERIAL

1. Email me @:
davide.bolognini@unifi.it
2. Get slides (.key + .pdf) from GitHub:
<https://github.com/davidebolo1993/Classes>

THAT ' ALL , FOLKS !
