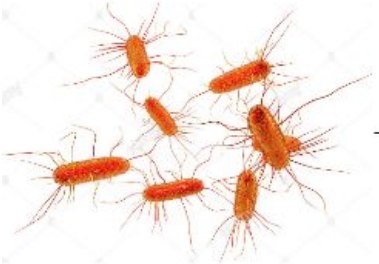
De Novo Genome Assembly

Etienne JEAN
M2 BI - Short Project Presentation
September 2018

Principle of de novo assembly

CELLS



SEQUENCER

READS

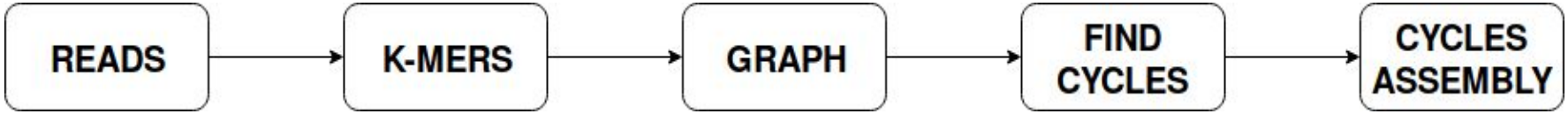
```
@IL31_4368:1:1:1001:7078/2
TGGGTGTGTGTGTTTGCGTGCGCGCTTCTGGGTGTAGTTATGTGTGTGTCTG
+
??>D7DD?AD?EDECEAB@<@,7@A9'>7><BC;=A31?A.;<>>B)A708>>4
@IL31_4368:1:1:1001:11670/2
AGGATGATGACAGCAACTTTCATATGGATTTCATCGTGGCTGCATCCAACCTCC
+
;EEE,FEAEFEFFFECEFCEFA>FCEEC@@ECEBCDFFAFD9E;9E@E?BD@?C
@IL31_4368:1:1:1001:15110/2
GGAGTACAAGAAGGAAGGCATTGAGTGGACGTTCATTGACTTTGGGATGGACCT
+
FCEFCGGEGFECEEGCGCAAFEC=EAFGFEE?8BAA<C=DA=4:EE>9@6;4B>
@IL31_4368:1:1:1002:6807/2
AGTCTTTGGACTTGCAAGAGAAGCCCAGCATTAAGAAAGAGACCCTCCTCAAAA
```

ASSEMBLER

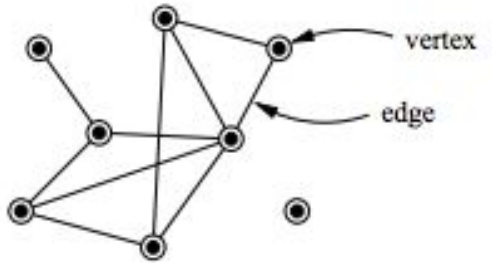
```
>fasta sequence 1
CCGcAagCGATtCGTAGCGCGCTGGaCGTCTACGGCGAGAACGTCcGCAAGATCGTGCCC
ATCGACAGCCCGcAGTTCGGCCGCGTCACCTGCGTCtGCATCGGCGCGATGATGGTCGGT
AGCATCCACATGACCGTCGAGGAGGGCCAGCAGGTGAAGCGCGGCCAGgAGTTTGGGTAC
TTCGCTTTCGGTGGCTCGACGATCGTCGTGCTCTTCGAGAAGGGCGTCGTGCAGTGGGAC
GAGGACTTGCTGATCAACAGCCGCGCGTGCCTCGAGACTCTCGTGCGCGTCGGCATGGGT
ATCGGgcAcGGCAAaCGcAAGCCGAACACGGACGGTAGTAGTGCACAGTAGACGTCGGAT
cGACcGCCTTtcTGGtGTGCATGTAGATACCTCGCGGAGTAGCCTACCcGGACCCTCCTA
ATAACGGACTTGTATGCAAACGTCCGCGCGGGTCCGCGACTACGTGCGCGGCGTGCCACG
```

GENOME ASSEMBLY

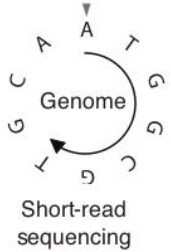
Description of an assembler



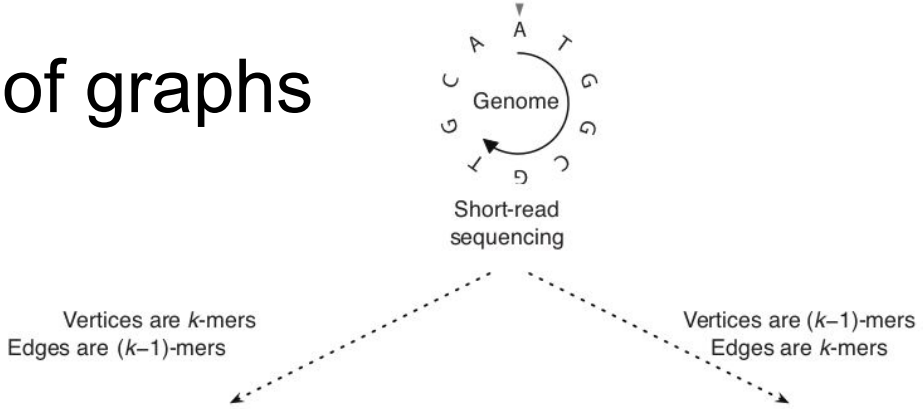
- **K-mers** strategy : substrings of length K
  - Remove redundancy of reads
  - Graph is no bigger than needed
  - Necessary step for De Bruijn graph
- **Graph** : set of **vertices** connected by **edges**



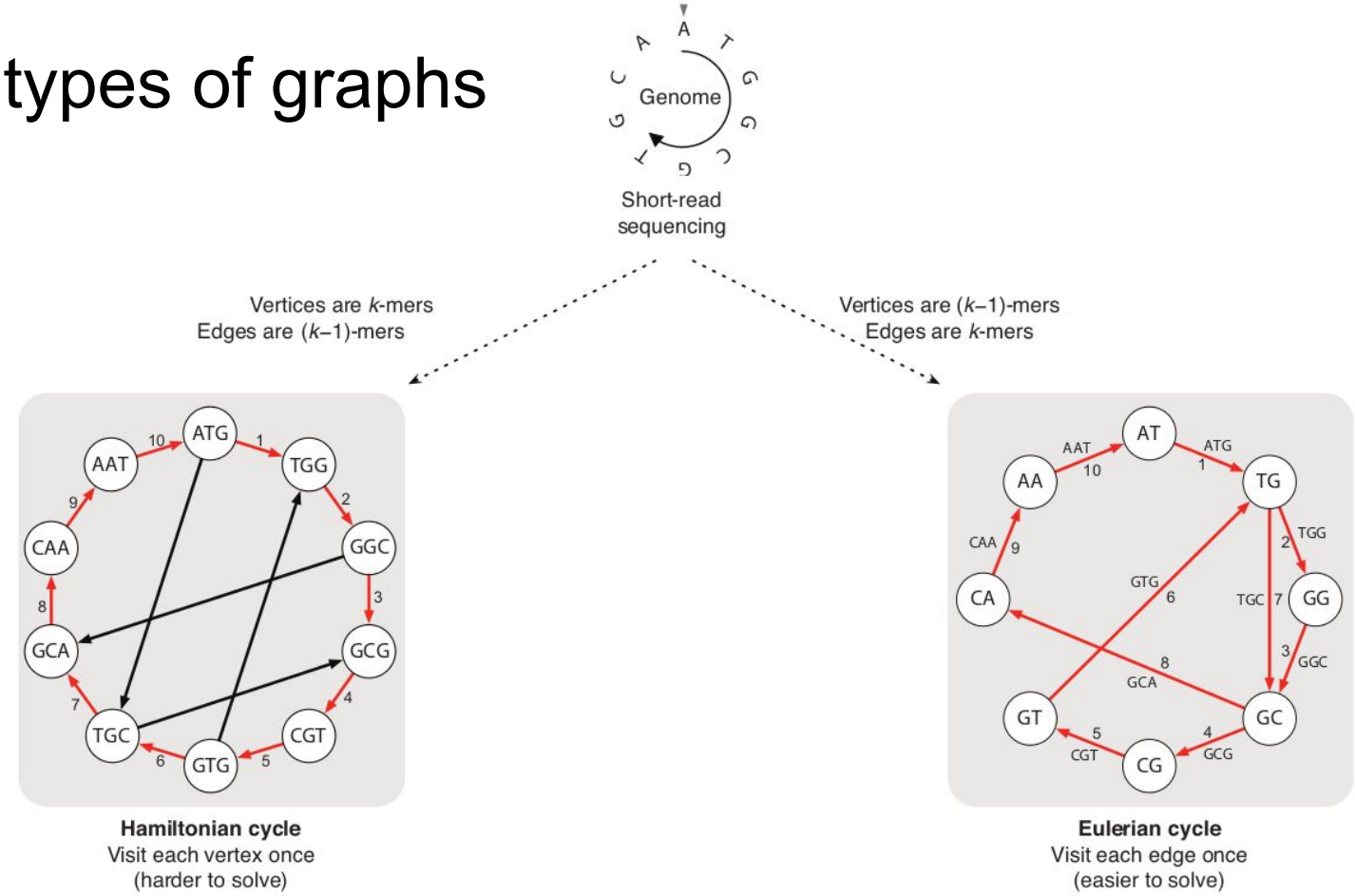
Two types of graphs



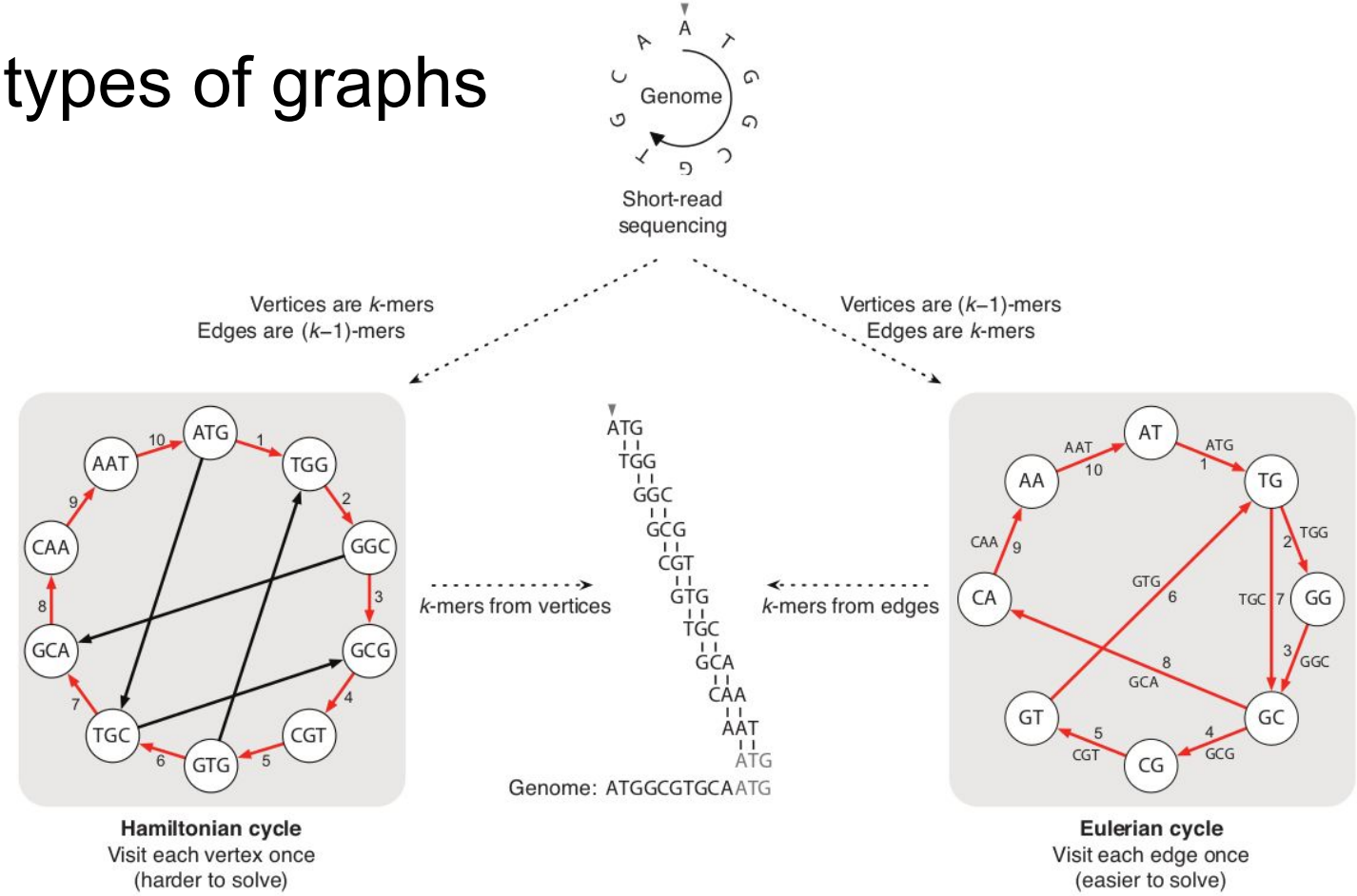
Two types of graphs



Two types of graphs

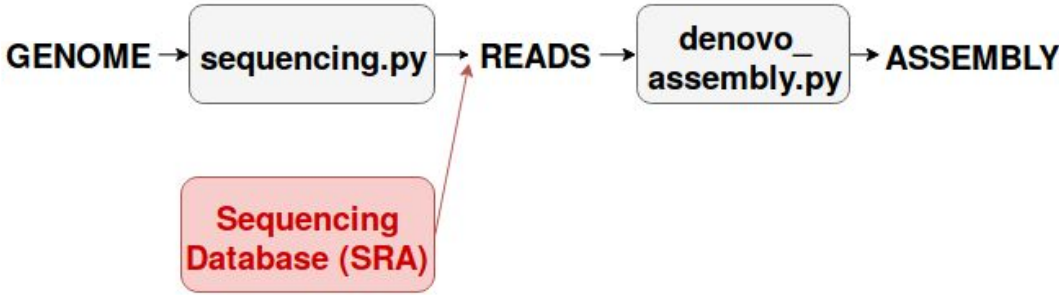


Two types of graphs

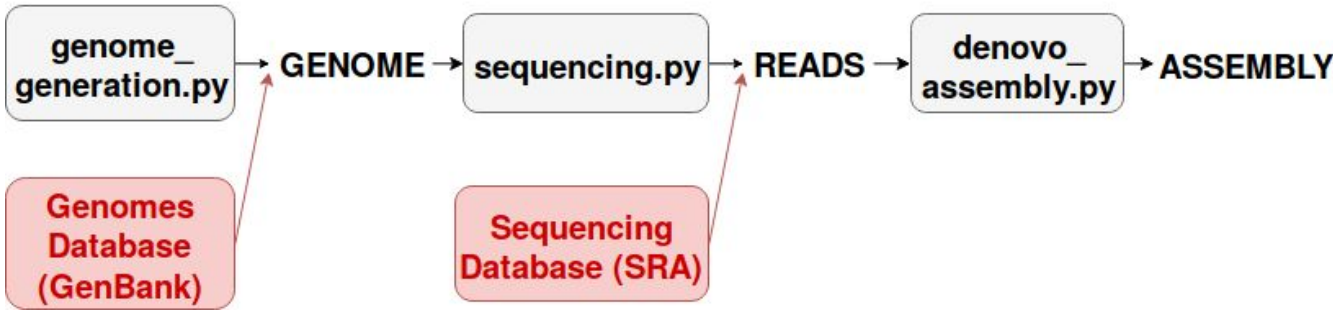


Program architecture : 4 different modules

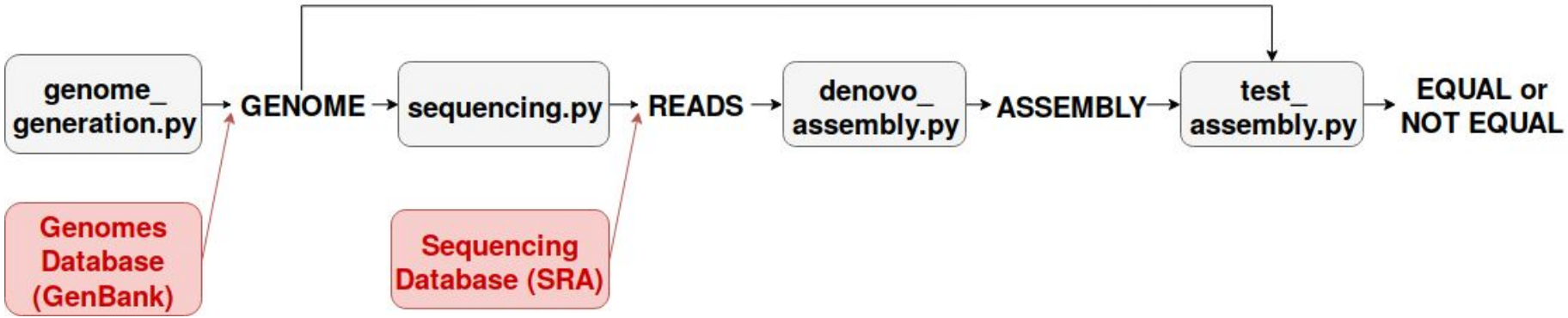
Program architecture : 4 different modules



Program architecture : 4 different modules

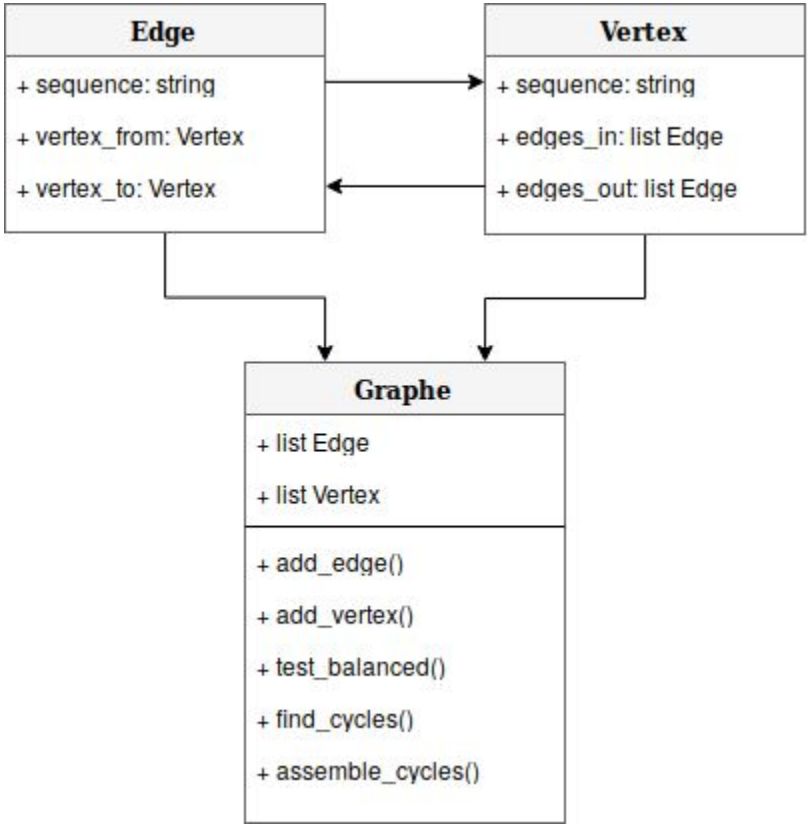


Program architecture : 4 different modules



Graph implementation

- Object oriented
- Vertices object point to Edges object
  - Reduces computation time
- Adjacency list
  - More adapted to sparse graphs than adjacency matrix
- Test for balanced De Bruijn graph
- Find Eulerian cycles
- Assemble cycles into one



Results

- Random genome, 1 Mb :
  - reads 100 bp
  - coverage 50x
  - K-mers 55 bp

  93 seconds

- Mycoplasma genitalium genome, 586 kb :
  - reads 400 bp
  - coverage 100x
  - K-mers 250 bp

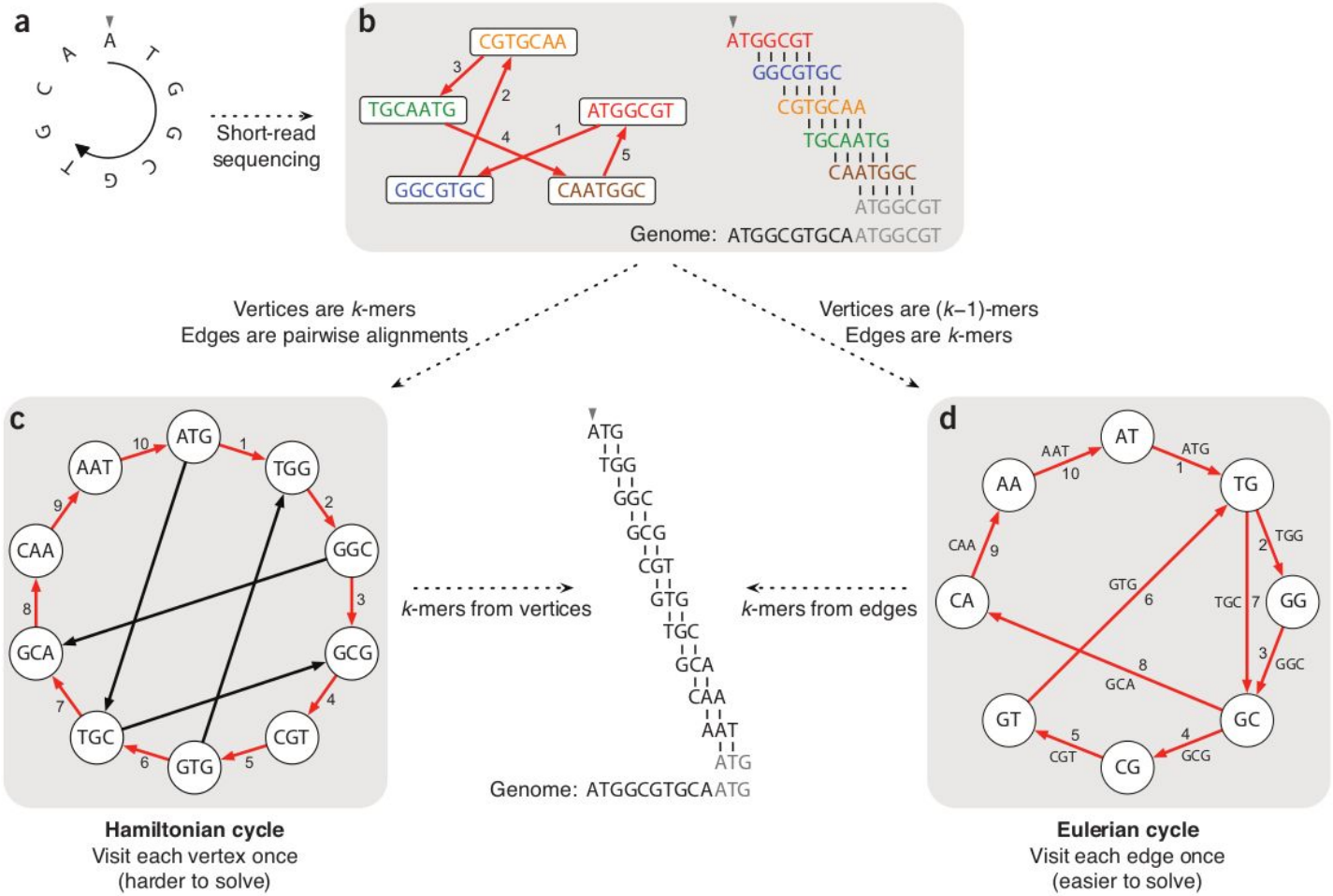
  82 seconds

Improvements

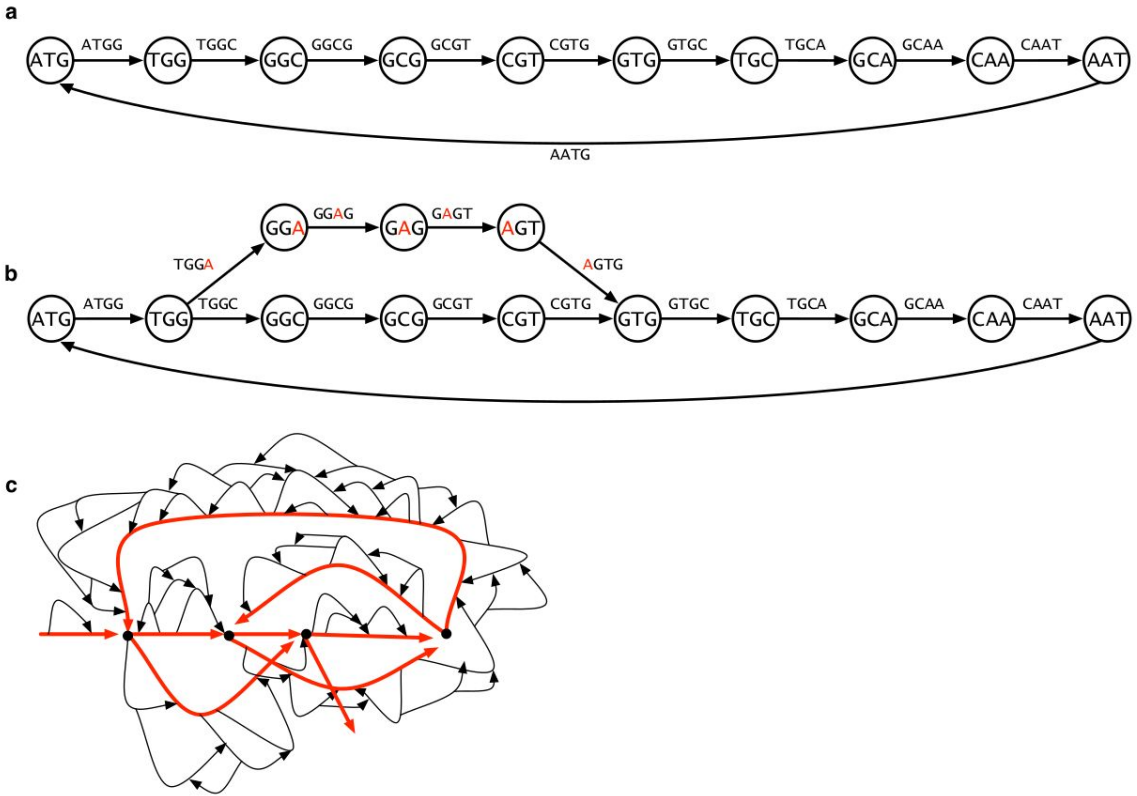
- Sequencing errors management
  - sequencing.py is currently not optional
- Multiple assemblies
  - 2 cycles connected by 2 different nodes
- Linear chromosome assembly
  - Eulerian path instead of Eulerian cycle
- Partial assemblies
  - When graph is not entirely balanced
  - Find set of contigs (subsequences of the genome)
- Multiple linear chromosomes assembly
  - Find different contigs, that correspond to different chromosomes
- Repeated sequences management
  - Introduce k-mers multiplicity

Thank you for your attention

Supplementary slides


a
Short-read sequencing
b
CGTGCAA
TGCAATG
ATGGCGT
GGCGTGC
CAATGGC
Genome: ATGGCGTGCAATGGCGT
Vertices are k-mers
Edges are pairwise alignments
Vertices are (k–1)-mers
Edges are k-mers
c
d
k-mers from vertices
k-mers from edges
Genome: ATGGCGTGCAATG
Hamiltonian cycle
Visit each vertex once
(harder to solve)
Eulerian cycle
Visit each edge once
(easier to solve)

Sequencing errors : creation of bulges in the graph



K-mers multiplicity for repeated sequences