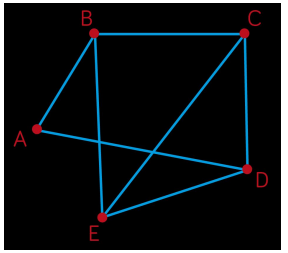
De novo Assembly of a little circular genome

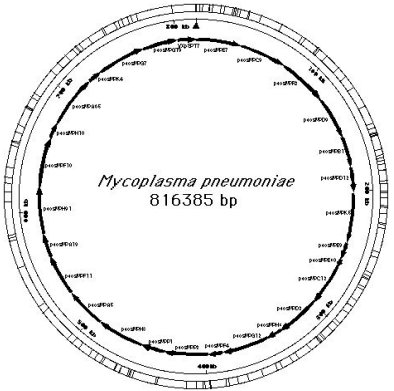
Adam Bellaïche, University of Paris 7, M2BI
adam.bellaiche@gmail.com

Introduction

- subject come from: *Compeau, P. E., Pevzner, P. A., & Tesler, G. (2011). How to apply de Bruijn graphs to genome assembly. Nature biotechnology,29(11), 987.*



- goal: write a program which permit to assemble de novo a little circular genome by using a De Bruijn graph and one of strategy describe in this article.



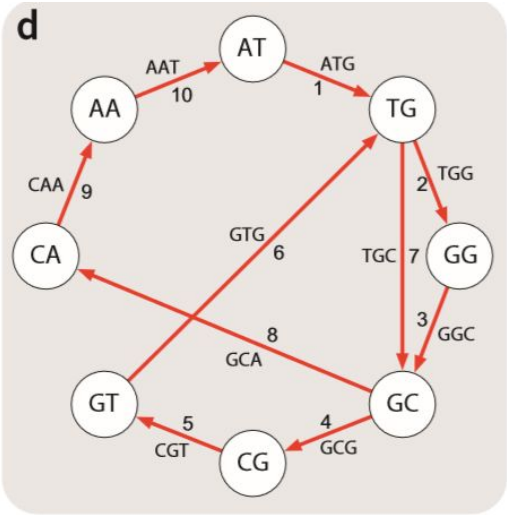
- the two strategy: use an hamiltonian cycle or **an Euler cycle**.

- Euler cycle: a cycle which start and end at the same vertex and visits each edge only once.

Adam Bellaïche, University of Paris 7, M2BI
adam.bellaiche@gmail.com

Some Explanations on the De Bruijn Graph for genome assembly

- De Bruijn Graph is an oriented graph which permit to **represent words of length *n*** with **overlap of words of length *n-1*** construct with a **given alphabet**.
- genome ⇨reads ⇨k-mers ⇨suffix and prefix
- node can’t appear twice.



Drawing which represent a graph of De Bruijn apply for genome assembly. (from *Compeau, P. E., Pevzner, P. A., & Tesler, G. (2011). How to apply de Bruijn graphs to genome assembly)*

Adam Bellaïche, University of Paris 7, M2BI
adam.bellaiche@gmail.com

Materials and Methods

Materials:

- python 3:
  - three files: main.py, Genome.py, Graph.py
  - four extern library: copy, random, time and sys
- fasta sequence of Mycoplasma genitalium taken on GenBank

Methods

- Initialize variables and use index, NameOfTheFonction, name_of_the_variable
- Generate the data:
  - random genome or given genome
  - random read with given length
  - k-mers with given length
  - suffix and prefix
- Generate the graph
  - create edge and node from distincts suffix or prefix
- Find a Euler Cycle
- Test the found sequence
  - Consider the circularity and compare initial sequence with founded sequence.

Adam Bellaïche, University of Paris 7, M2BI
adam.bellaiche@gmail.com

Find an Euler Cycle

```
EulerianCycle(BalancedGraph)
  form a Cycle by randomly walking in BalancedGraph (avoiding already visited edges)
    while Cycle is not Eulerian
      select a node newStart  in Cycle with still unexplored outgoing edges
      form a Cycle' by traversing Cycle from newStart and randomly walking afterwards
      Cycle ← Cycle'
  return Cycle
```

Euler algorithm taken from youtube channel: Bioinformatics Algorithms: An Active Learning Approach.

Results

Génome en pb	nucléotides disponible	nombre de read	taille des reads	tailles des k-mers	temps d'exécution en seconde	séquence trouvée?
10000	“ATCG”	5000	100	20	0.5	oui
100000	“ACTG”	50000	100	55	20	oui
500000	“ATCG”	20000	100	55	437	oui
mycoplasma genitalium (environ 600000)	“ACTG”	20000	400	300	538	oui

Conclusion

- Parameters are very importants
- program works but it could be faster
- Need to be throw again
  - if the built graph is not eulerian
  - if the found sequence is not similar to initial sequence

Génome en pb	nucléotides disponible	nombre de read	taille des reads	tailles des k-mers	temps d'exécution en seconde	séquence trouvée?
10000	“ATCG”	5000	100	20	0.5	oui
100000	“ACTG”	50000	100	55	20	oui
500000	“ATCG”	20000	100	55	437	oui
mycoplasma genitalium (environ 600000)	“ACTG”	20000	400	300	538	oui

Available on github

https://github.com/toontun/AssemblyDenovo

AssemblyDenovo UML