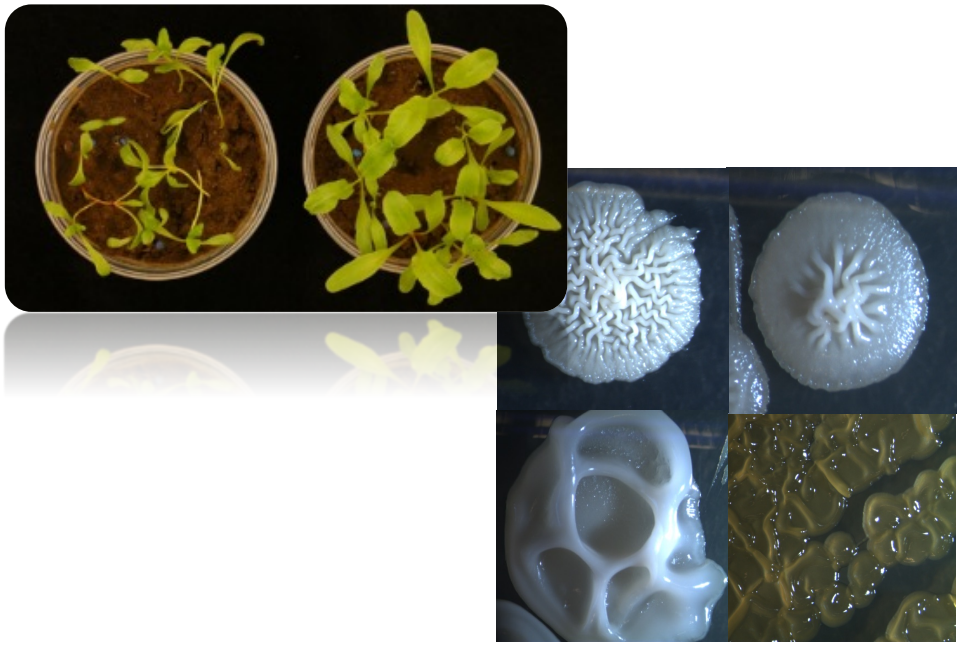
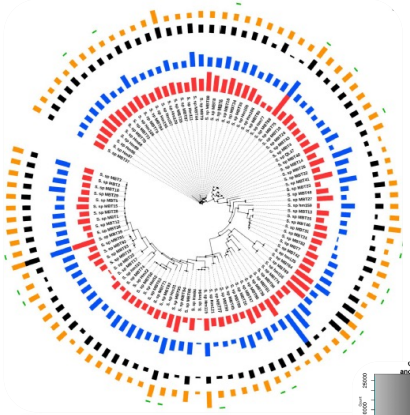
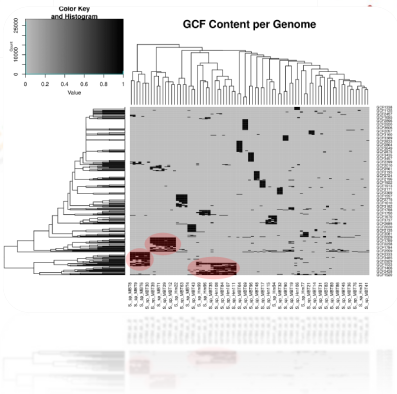
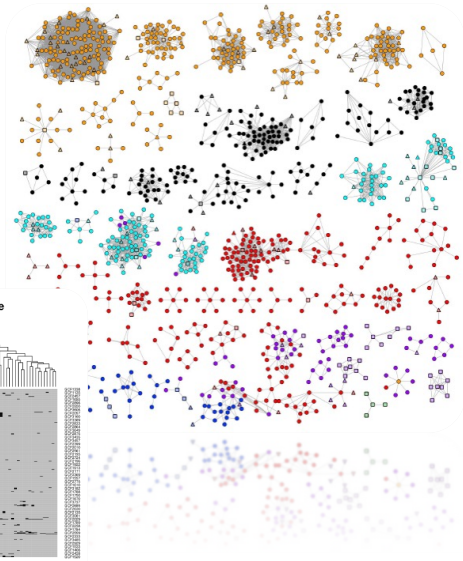
# Técnicas de análisis de los microbiomas II: ensamblaje



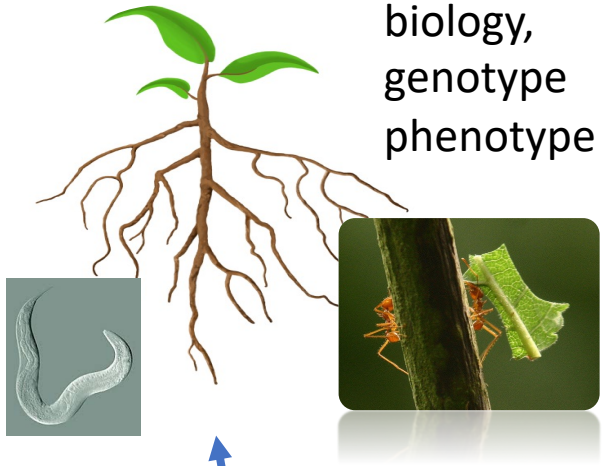




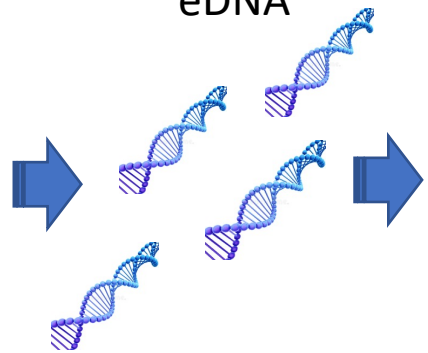
Victor J **Carrión**
vcarrion@uma.es

@VCarryOn1



biology,
genotype
phenotype

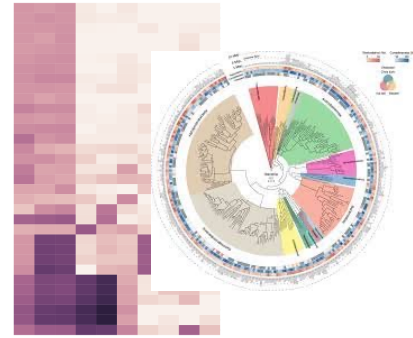
eDNA



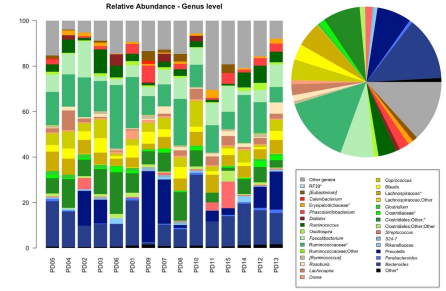
NGS sequencing

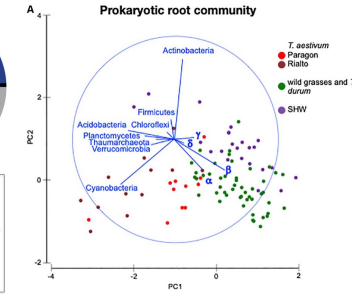
culturing
phenotyping

functions, meta'omics



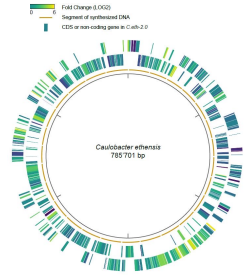
taxonomy, phylogeny



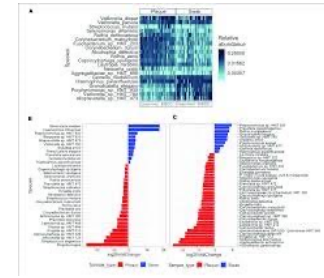



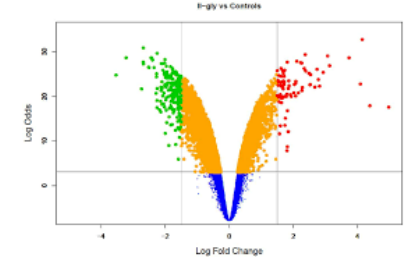

differential abundance (function & taxonomy)

genome assembly
comparative genomics

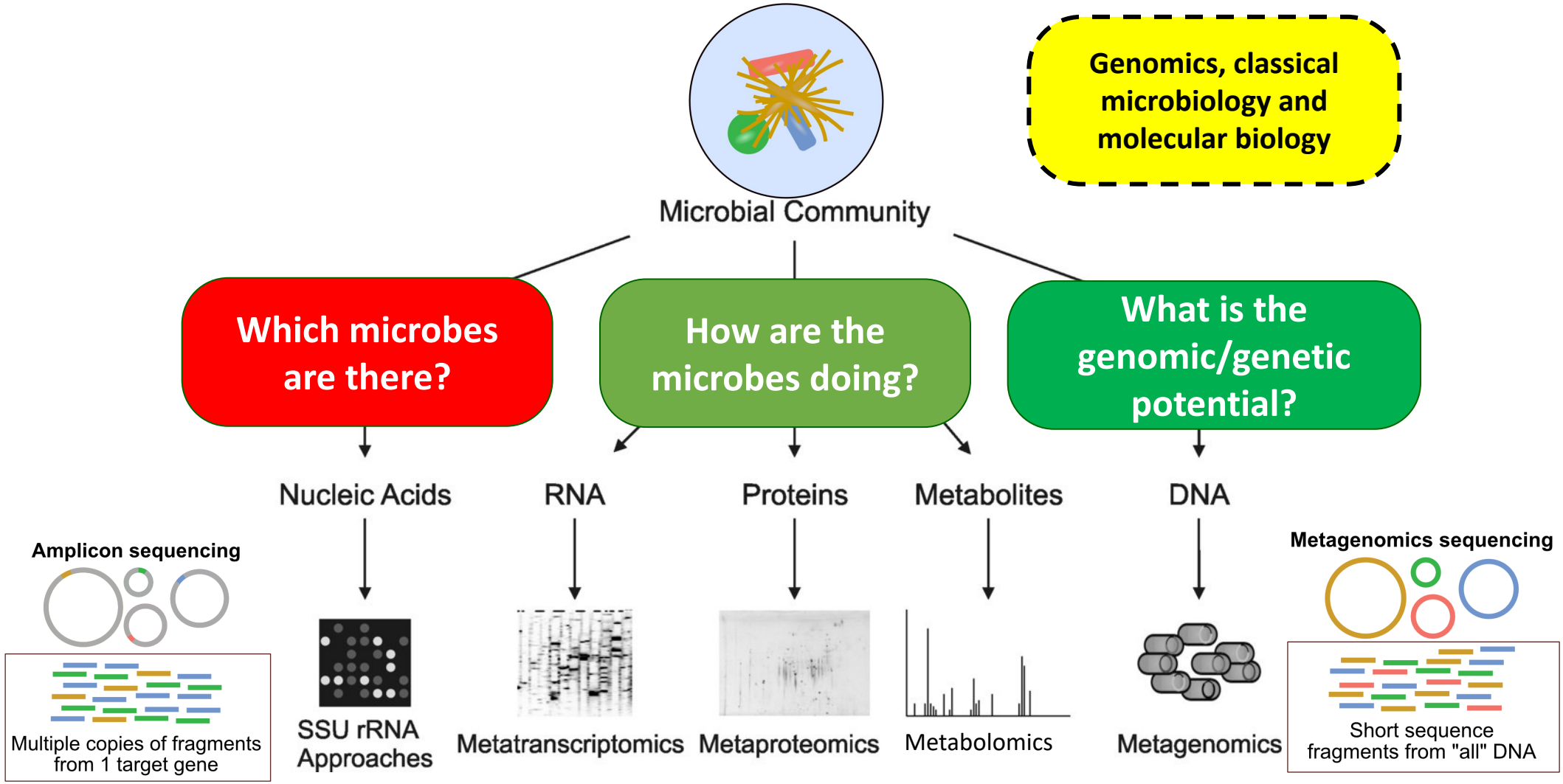


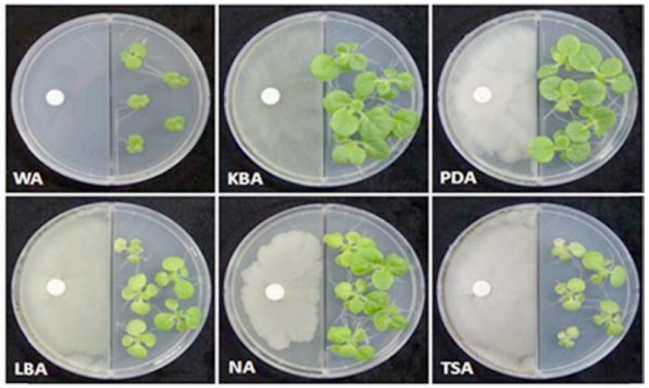

cross-link

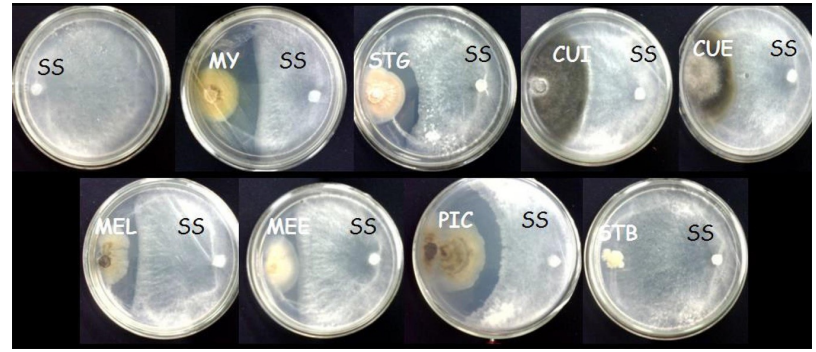




# THE TOOLBOX …


Genomics, classical microbiology and molecular biology
Microbial Community
Which microbes are there?
How are the microbes doing?
What is the genomic/genetic potential?
Nucleic Acids
RNA
Proteins
Metabolites
DNA
Amplicon sequencing
Multiple copies of fragments from 1 target gene
SSU rRNA Approaches
Metatranscriptomics
Metaproteomics
Metabolomics
Metagenomics
Metagenomics sequencing
Short sequence fragments from "all" DNA




DNA reads

Quality control

FastQC

Filtered DNA reads

Genome assembly

Velvet

SPAdes

MEGAHIT

Genome

Quast

Quality Assessment Tool for Genome Assemblies

CheckM

```
spades.py -k 21,33,55,77,99,127 --careful --only-assembler <your reads> -o spades_output
```

# LEARNING GOALS

1. (meta)Genome Assembly
2. Alignments
3. Phylogeny
4. Annotation
5. Comparative genomics

# OUTLINE

MICROBIOME TECHNOLOGIES & ANALYSIS II

1. (meta)Genome assembly -> (Li et al., 2011)
    a. Overlap-layout-consensus
    b. String graphs (de Bruijn graphs)
2. Whole genome alignments -> (Ahmed et al., 2019)
    a. Global
    b. Local
    c. Glocal
3. Phylogeny -> (Aniscar et al, 2020)

# 1. (META)GENOME ASSEMBLY


PHASE TWO : INTERPRETATION
SHERMAN Miller Ledger
I THINK I FOUND A CORNER PIECE.
3 BILLION PIECES
GENOME

# 1. GENOME ASSEMBLY DEFINITIONS

***Read***
a 500-900 long word that comes out of sequencer

***Mate pair***
a pair of reads from two ends of the same insert fragment

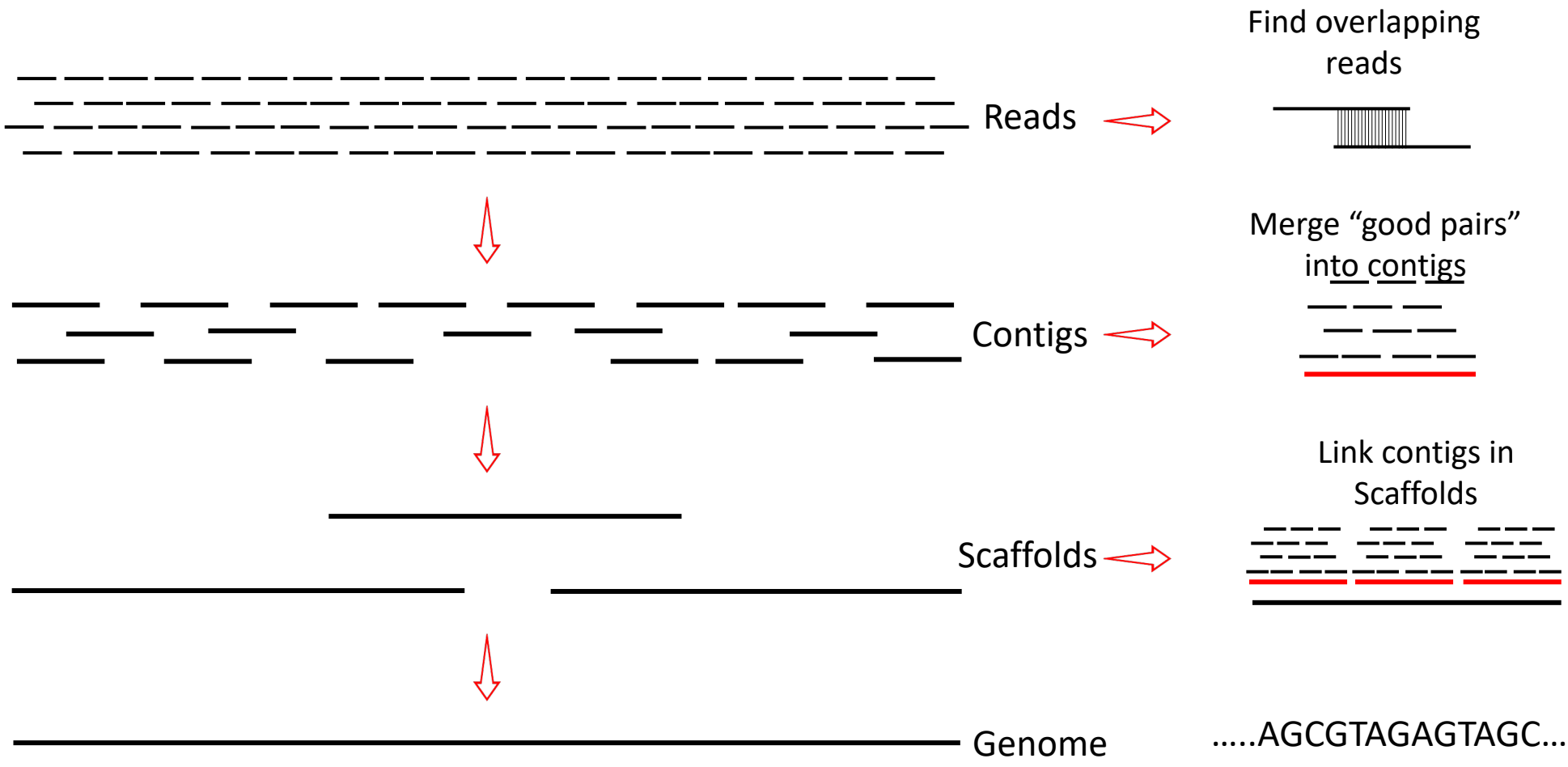
***Contig***
a contiguous sequence formed by several overlapping reads with no gaps

***Supercontig/scaffold***
an ordered and oriented set of contigs, usually by mate pairs

***Consensus sequence***
sequence derived from multiple alignment of reads in a contig

# 1. WHOLE GENOME SHOTGUN SEQUENCING

**>> steps to assemble a genome**



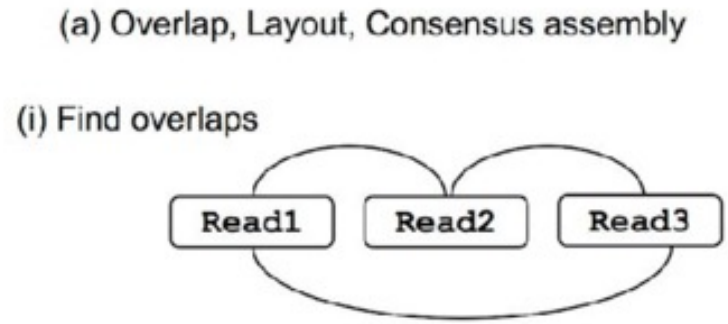
# 1A. GENOME ASSEMBLY I: OVERLAP-LAYOUT-CONSENSUS

**OLC** es un método de ensamblaje de genomas basado en identificar solapamientos directos entre lecturas, construir un ordenamiento óptimo y generar una secuencia de consenso, especialmente adecuado para lecturas largas y genomas complejos.

# 1A. OVERLAP-LAYOUT-CONSENSUS (OLC)

**OLC** generally works in three steps:

(i) first overlaps **(O)** among all the reads are found



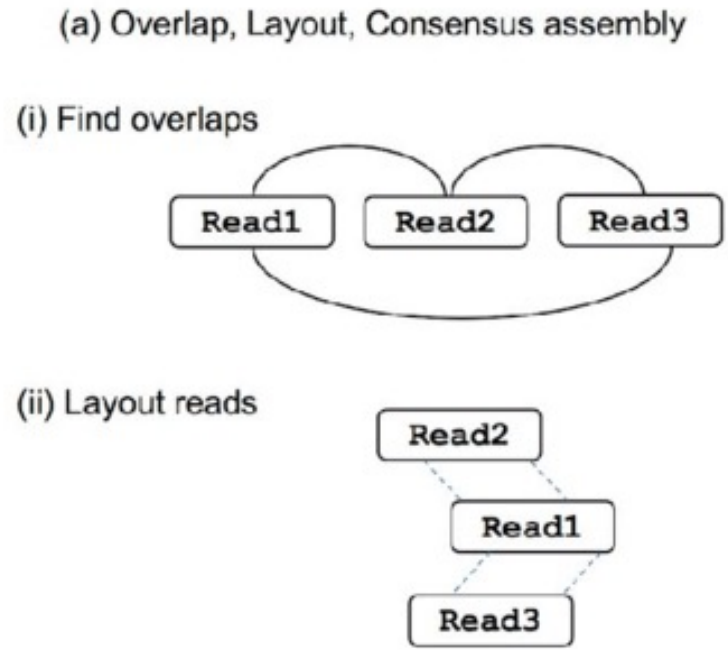

Ayling et al., 2018

# 1A. OVERLAP-LAYOUT-CONSENSUS (OLC)

**OLC** generally works in three steps:

(i) first overlaps **(O)** among all the reads are found

(ii) then it carries out a layout **(L)** of all the reads




Ayling et al., 2018

# 1A. OVERLAP-LAYOUT-CONSENSUS (OLC)

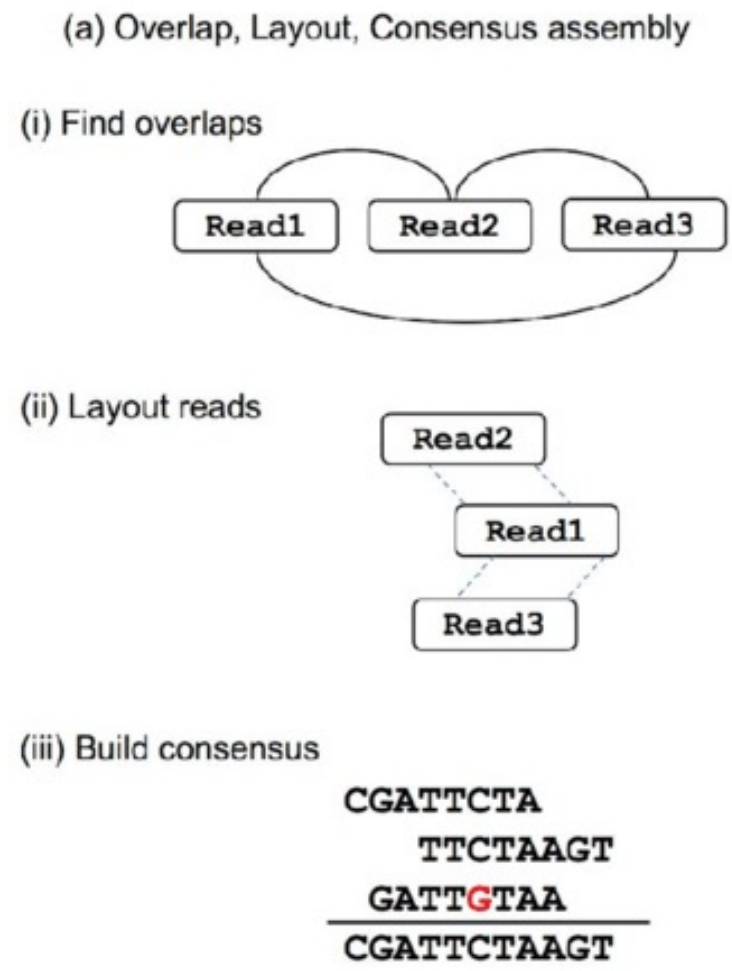
**OLC** generally works in three steps:

(i) first overlaps **(O)** among all the reads are found

(ii) then it carries out a layout **(L)** of all the reads

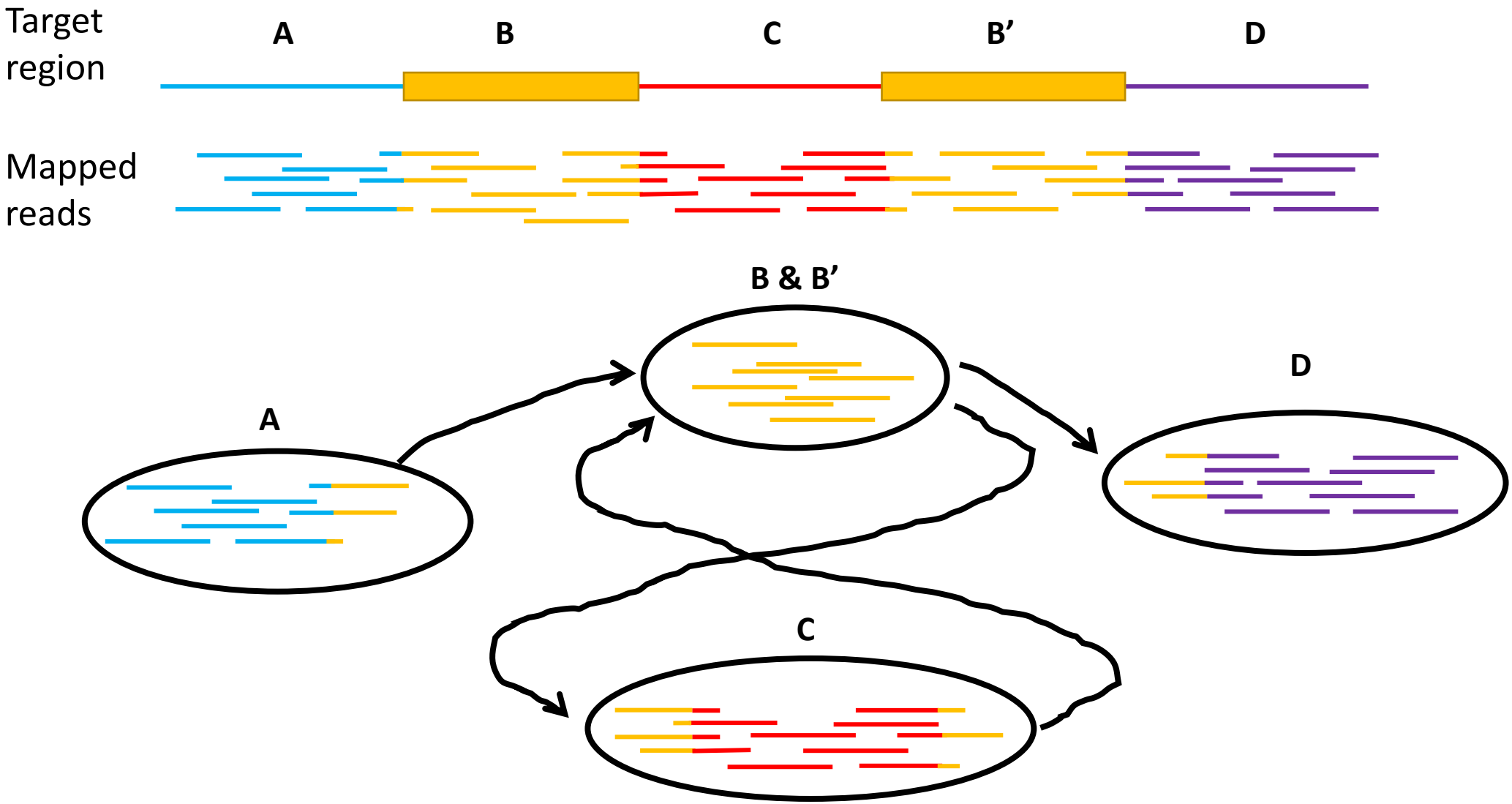
(iii) and overlaps information on a graph and finally the consensus **(C)** sequence is inferred.

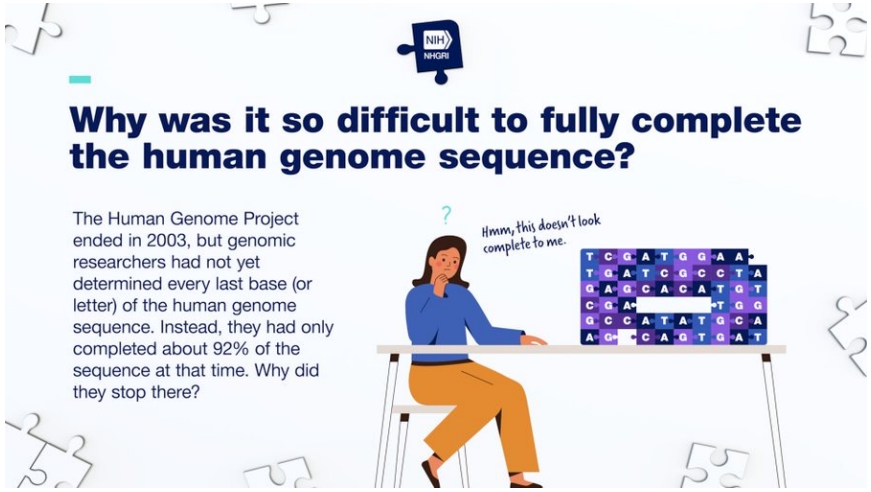
- developed by Staden (1980)

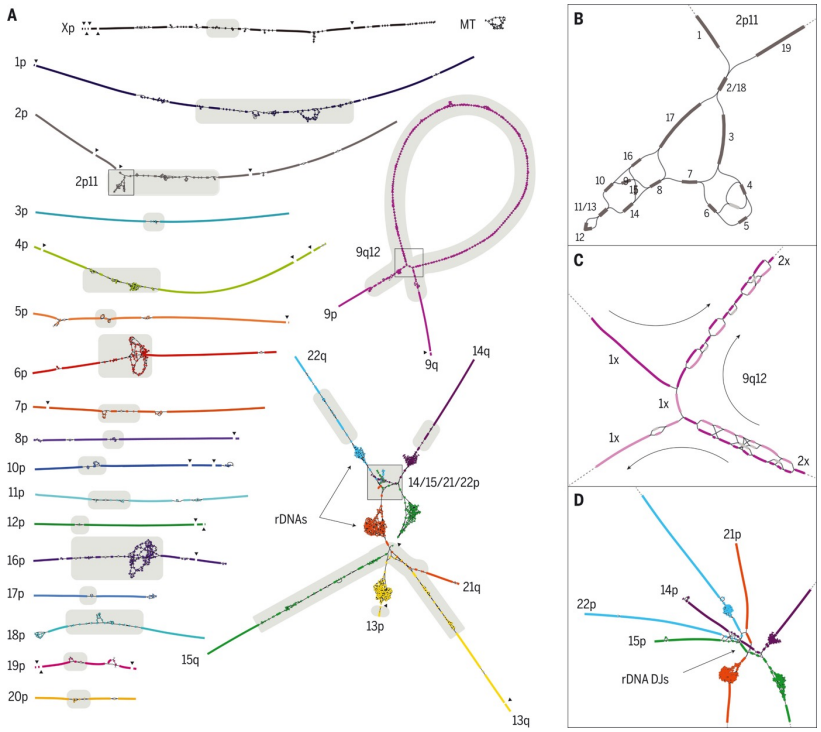



Ayling et al., 2018

# 1A. THE PROBLEM OF REPETITIVE SEQUENCES







Nurk et al., 2022

# OUTLINE

MICROBIOME TECHNOLOGIES & ANALYSIS I

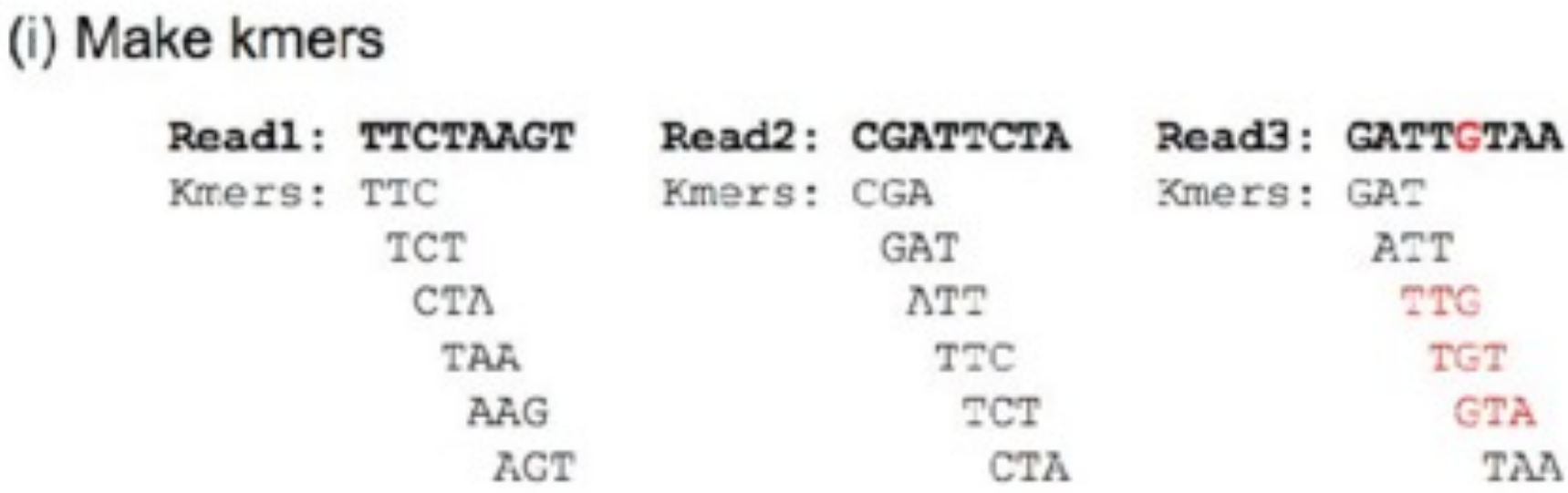
1. (meta)Genome assembly -> (Li et al., 2011)
    a. Overlap-layout-consensus
    b. String graphs (de Bruijn graphs)
2. Whole genome alignments -> (Ahmed et al., 2019)
    a. Global
    b. Local
    c. Glocal
3. Phylogeny -> (Aniscar et al, 2020)

# 1B. GENOME ASSEMBLY II: STRING GRAPH >>DE BRUIJN GRAPH

**De Bruijn Graph (DBG)** es un método de ensamblaje de genomas que representa las lecturas como nodos conectados por k−mers, permitiendo reconstruir el genoma mediante **un recorrido euleriano**. Es ideal para lecturas cortas y escalable a grandes conjuntos de datos.

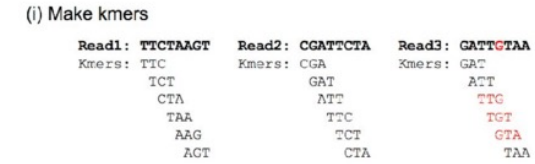
# 1B. DE BRUIJN GRAPH

first chopping reads into much shorter k-mers

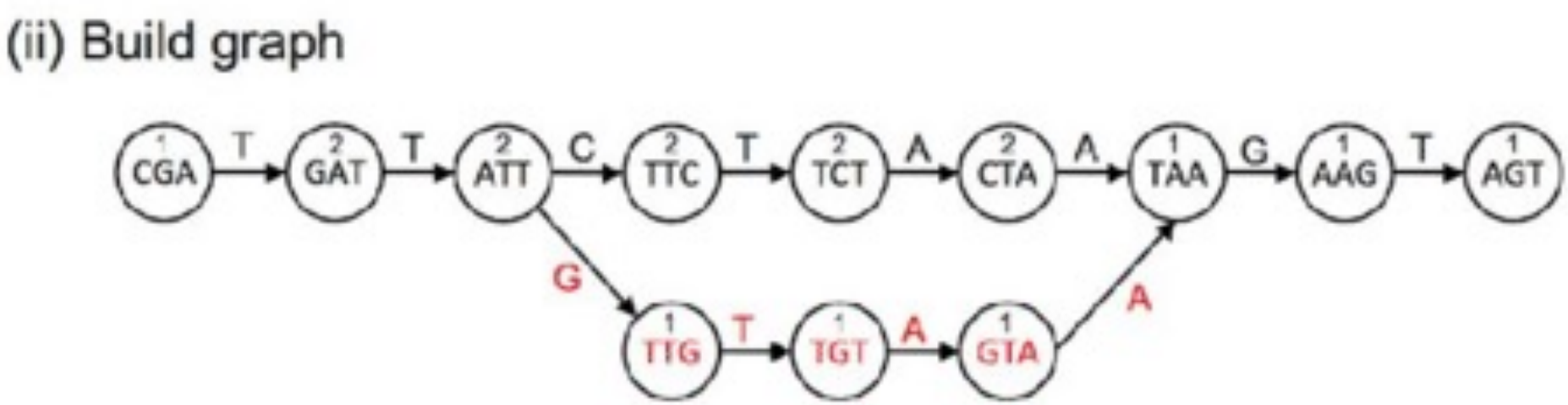



Ayling et al., 2018

# 1B. DE BRUIJN GRAPH



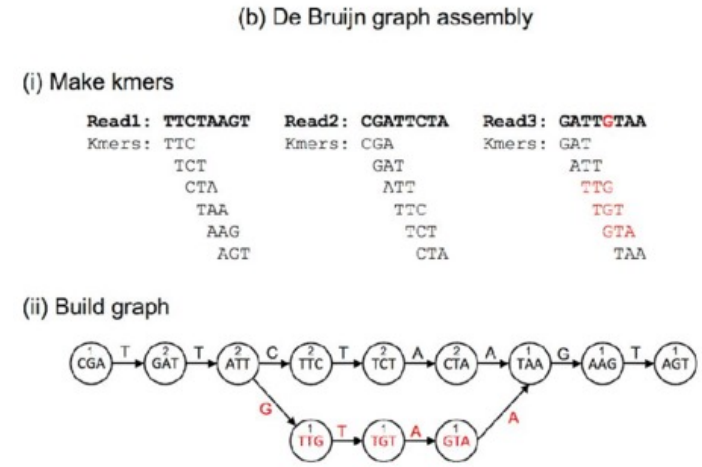

using all the k-mers to form a DBG

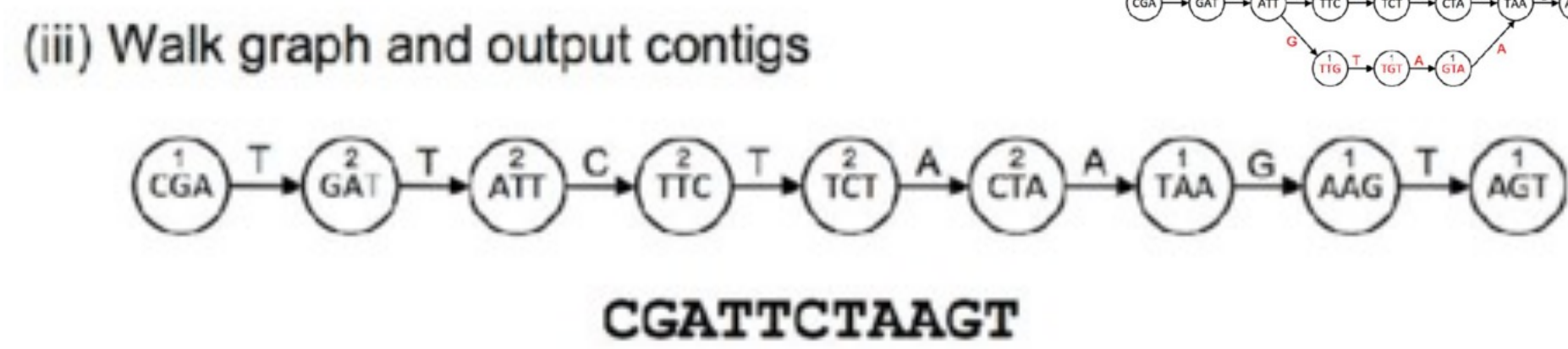



Ayling et al., 2018

# 1B. DE BRUIJN GRAPH

finally inferring the genome sequence on the DBG.




The first DBG assembler EULER was published in 2001 by Pevzner and Waterman.

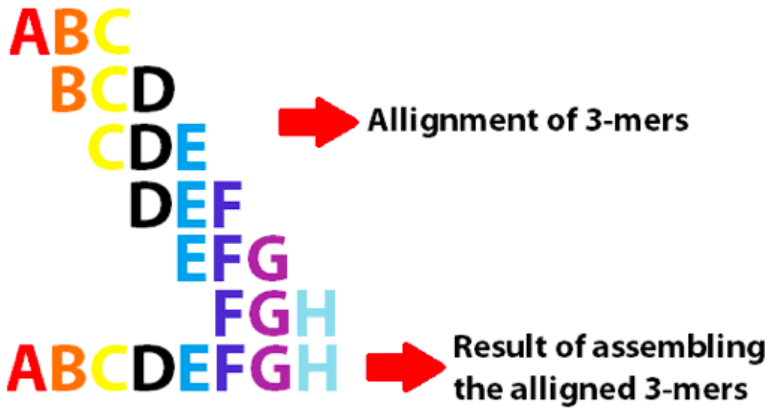
**Rather than overlapping “reads” you are overlapping k-mers**

Ayling et al., 2018

# 1B. THE PROBLEM OF REPETITIVE SEQUENCES

## *k*-mer

Short, **unique element of DNA sequence of length k**, used by many assembly algorithms




- Sort all k-mers in reads (k=24)
- Find pairs of reads sharing a k-mer
- Extend to full alignment - throw away if not >97% similar

AAGAGT (AA, AG, GA, AG, GT)

It contains the following 2-mers:

AA
AG (twice)
GA
GT
AAGA?

And the following 3-mers:

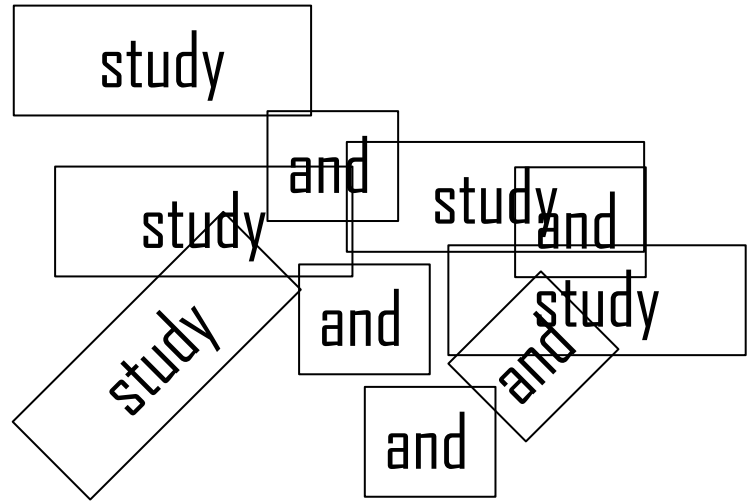
AAG
AGA
GAG
AGT
AAGAGT

Ayling et al., 2018

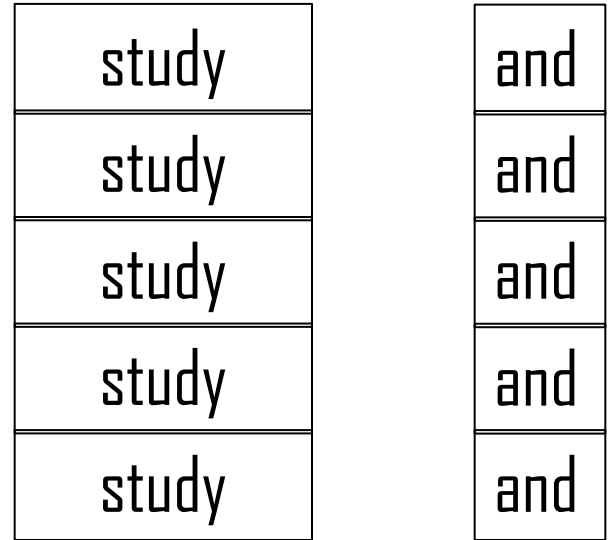
# 1B. DE BRUIJN GRAPH

"study and study and study"

Sequencing

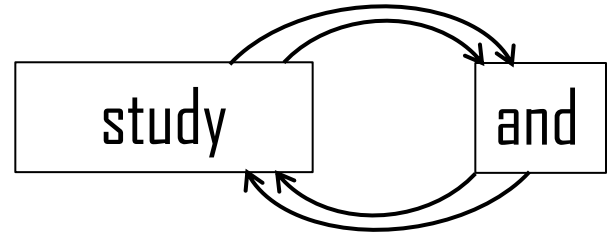



Assembly based on OCL



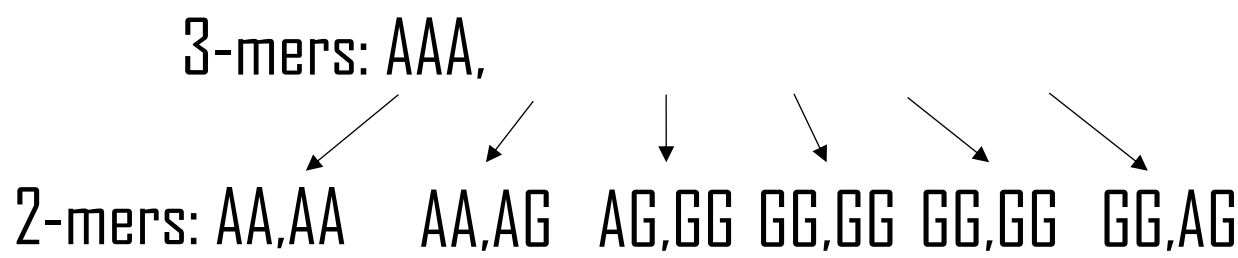
# 1B. DE BRUIJN GRAPH

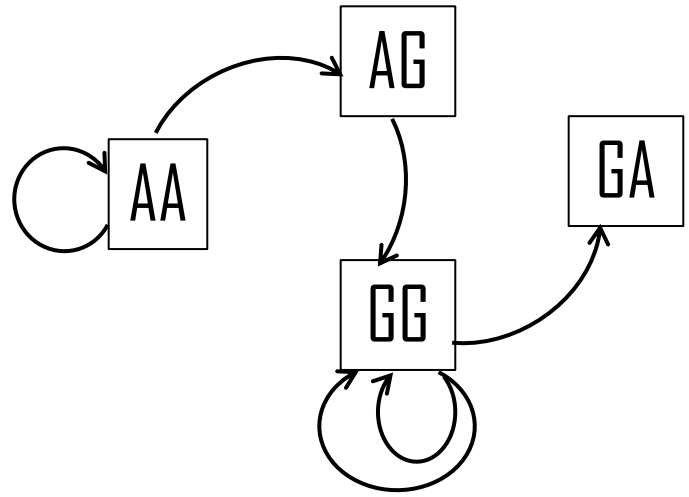
"study and study and study"



# 1B. DE BRUIJN GRAPH

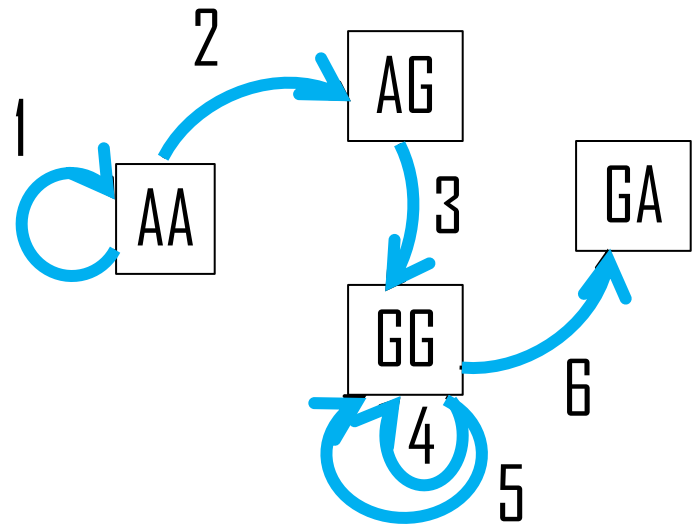
Genome: AAAGGGGA



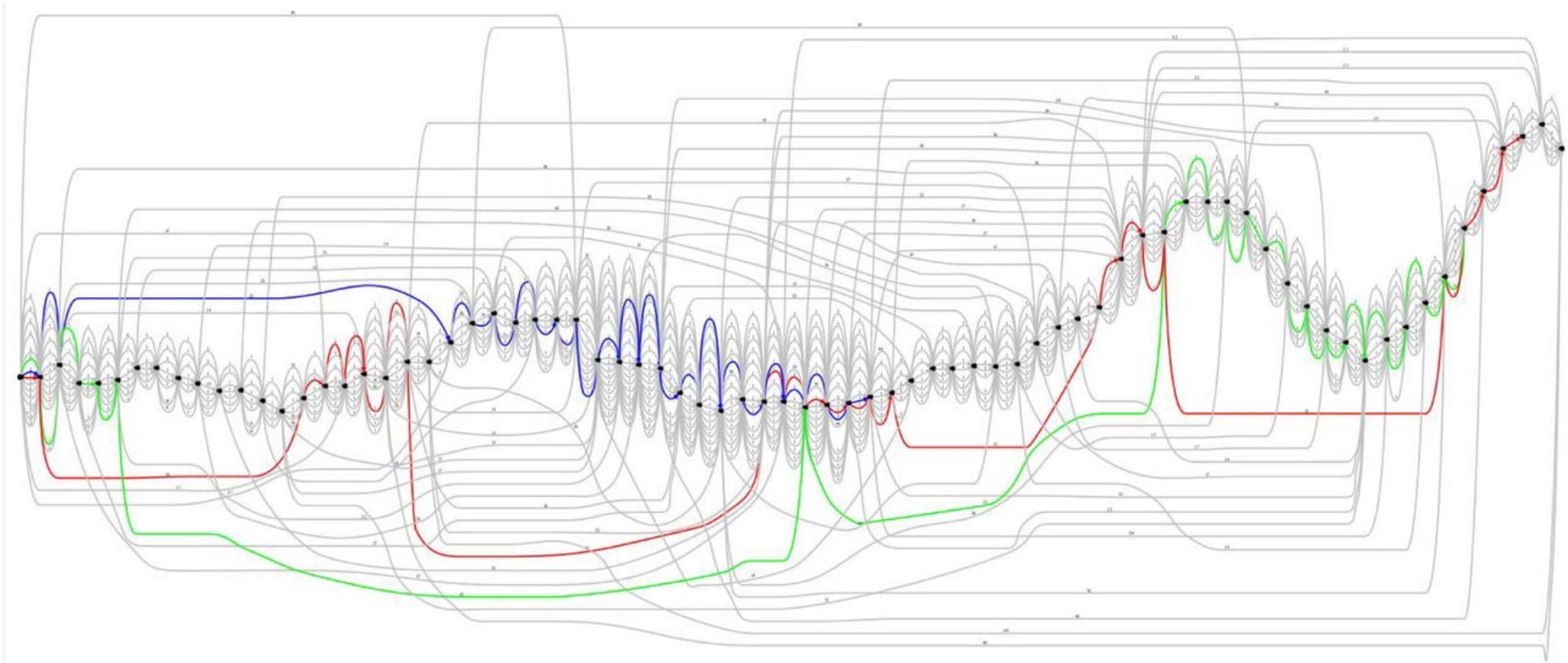

Adapted from Ben Langmead (Coursera)

# 1B. DE BRUIJN GRAPH




AAAGGGGA

# 1B. DE BRUIJN GRAPH



# 1B. DE BRUIJN GRAPH

## Using SPAdes De Novo Assembler

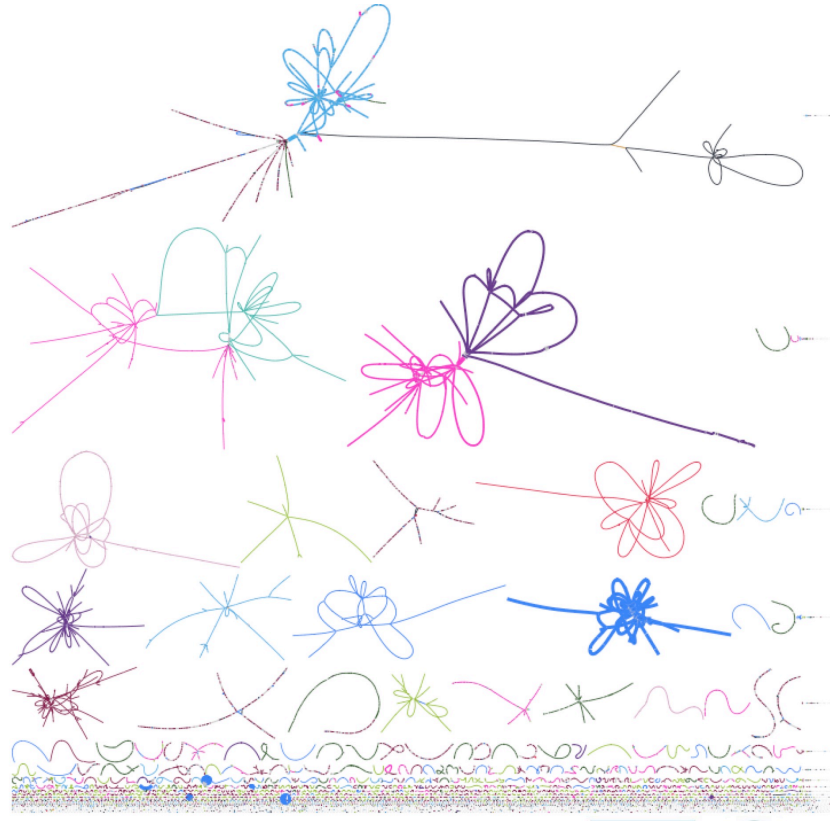
Andrey Prjibelski, Dmitry Antipov, Dmitry Meleshko, Alla Lapidus, Anton Korobeynikov

First published: 19 June 2020 | https://doi.org/10.1002/cpbi.102 | Citations: 99


```
spades.py -k 21,33,55,77,99,127 --careful --only-assembler <your reads> -o spades_output
```

# 1B. DE BRUIJN GRAPH

Bandage is a tool for assembly graph visualization
It can be used to inspect metagenomic assembly graphs for manual curation



Tutorial at:
https://tylerbarnum.com/2018/02/26/how-to-use-assembly-graphs-with-metagenomic-datasets/

# HYBRID ASSEMBLIES

README GPL-3.0 license


Unicycler is an assembly pipeline for bacterial genomes. It can assemble Illumina-only read sets where it functions as a SPAdes-optimiser. It can also assembly long-read-only sets (PacBio or Nanopore) where it runs a miniasm+Racon pipeline. For the best possible assemblies, give it both Illumina reads *and* long reads, and it will conduct a short-read-first hybrid assembly.

Read more about Unicycler here:

> **Wick RR, Judd LM, Gorrie CL, Holt KE.** Unicycler: resolving bacterial genome assemblies from short and long sequencing reads. *PLoS Comput Biol* 2017.

And read about how we use it to complete bacterial genomes here:

> **Wick RR, Judd LM, Gorrie CL, Holt KE.** Completing bacterial genome assemblies with multiplex MinION sequencing. *Microb Genom* 2017.

# GENOME ASSEMBLY I: OVERLAP-LAYOUT-CONSENSUS

Phrap/Cross_match/Swat

phrap is a program for assembling shotgun DNA sequence data. Among other features, it allows use of the entire read and not just the trimmed high quality part, it uses a combination of user-supplied and internally computed data quality information to improve assembly accuracy in the presence of repeats, it constructs the contig sequence as a mosaic of the highest quality read segments rather than a consensus, it provides extensive assembly information to assist in trouble-shooting assembly problems, and it handles large datasets. See the phrap/cross_match/swat documentation and phrap documentation for additional information.

**Methods**

## ARACHNE: A Whole-Genome Shotgun Assembler

Serafim Batzoglou,[1,2,3] David B. Jaffe,[2,3,4] Ken Stanley,[2] Jonathan Butler,[2] Sante Gnerre,[2] Evan Mauceli,[2] Bonnie Berger,[1,5] Jill P. Mesirov,[2] and Eric S. Lander[2,6,7]

*[1]Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA; [2]Whitehead Institute/MIT Center for Genome Research, Cambridge, Massachusetts 02141, USA; [4]Department of Mathematics and Statistics, University of Nebraska, Lincoln, Nebraska 68588, USA; [5]Department of Mathematics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA; [6]Department of Biology, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA*

## Whole-genome shotgun assembly and comparison of human genome assemblies

Sorin Istrail[a], Granger G. Sutton[a], Liliana Florea[a], Aaron L. Halpern[b], Clark M. Mobarry[a], Ross Lippert[a], Brian Walenz[a], Hagit Shatkay[a,c], Ian Dew[a], Jason R. Miller[a], Michael J. Flanigan[a], Nathan J. Edwards[a], Randall Bolanos[a], Daniel Fasulo[a], Bjarni V. Halldorsson[a], Sridhar Hannenhalli[a,d], Russell Turner[a], Shibu Yooseph[a,e], Fu Lu[f], Deborah R. Nusskern[f], Bixiong Chris Shue[f], Xiangqun Holly Zheng[f], Fei Zhong[f], Arthur L. Delcher[g], Daniel H. Huson[f,h], Saul A. Kravitz[b], Laurent Mouchard[f,i], Knut Reinert[f,i], Karin A. Remington[b], Andrew G. Clark[k], Michael S. Waterman[l], Evan E. Eichler[m], Mark D. Adams[f,n], Michael W. Hunkapiller[o], Eugene W. Myers[p], and J. Craig Venter[b,q]

[a]Applied Biosystems, 45 West Gude Drive, Rockville, MD 20850; [b]The Center for the Advancement of Genomics (TCAG), 1901 Research Boulevard, Suite 600, Rockville, MD 20850; [f]Celera Genomics, 45 West Gude Drive, Rockville, MD 20850; [g]The Institute for Genomic Research (TIGR), 9712 Medical Center Drive, Rockville, MD 20850; [k]Department of Molecular Biology and Genetics, Cornell University, 227 Biotechnology Building, Ithaca, NY 14853; [l]Department of Mathematics, University of Southern California, 1042 West 36th Place, DRB 155, Los Angeles, CA 90033; [m]Department of Genetics, Case Western Reserve University, 10900 Euclid Avenue, Cleveland, OH 44106; [o]Applied Biosystems, 850 Lincoln Centre Drive, Foster City, CA 94404; and [p]Computer Science Division, University of California, 775 Soda Hall, Berkeley, CA 94720

# GENOME ASSEMBLY II: DE BRUIN GRAPHS

## An Eulerian path approach to DNA fragment assembly

Pavel A. Pevzner*, Haixu Tang†, and Michael S. Waterman†‡§

*Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA; and Departments of †Mathematics and

## Using SPAdes De Novo Assembler

Andrey Prjibelski, Dmitry Antipov, Dmitry Meleshko, Alla Lapidus, Anton Korobeynikov

First published: 19 June 2020 | https://doi.org/10.1002/cpbi.102 | Citations: 99

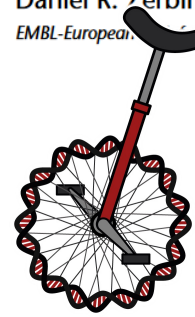
SPAdes

**Resource**

## Velvet: Algorithms for de novo short read assembly using de Bruijn graphs

Daniel R. Zerbino and Ewan Birney[1]

*EMBL-European [illegible]ics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, United Kingdom*



Unicycler

| Assembler | Description | Primary Application | Algorithm | Reference |
|---|---|---|---|---|
| **Unicycler** | Integrates short and long reads for efficient bacterial genome assembly. | Bacterial genomes | De Bruijn graph | Wick RR et al., PLoS Comput Biol, 2017. |
| **SPAdes (HybridSPAdes)** | A short-read assembler with a hybrid mode to incorporate long reads. | General-purpose | De Bruijn graph | Bankevich A et al., Journal of Computational Biology, 2012. |
| **MaSuRCA** | Merges short and long reads to assemble large genomes. | Large genomes | De Bruijn graph | Zimin AV et al., Bioinformatics, 2013. |
| **Flye (Hybrid Module)** | Primarily a long-read assembler that can utilize short reads for error correction. | General-purpose | Overlap-Layout-Consensus (OLC) | Kolmogorov M et al., Nature Methods, 2019. |
| **Canu + Pilon** | Canu assembles long reads; Pilon refines assemblies using short reads. | General-purpose | OLC (Canu), Polishing (Pilon) | Koren S et al., Nature Biotechnology, 2017. |
| **OPERA-MS** | Optimized for metagenomic hybrid assemblies, combining short and long reads. | Metagenomics | OLC | Bertrand D et al., Nature Methods, 2019. |
| **metaFlye** | A version of Flye tailored for metagenomic samples, integrating both read types. | Metagenomics | OLC | Kolmogorov M et al., Nature Methods, 2019. |
| **Wengan** | Focuses on hybrid assemblies of complex genomes, avoiding all-vs-all read comparisons. | Large genomes | OLC | Di Genova A et al., Nature Computational Science, 2021. |
| **TULIP** | A hybrid assembler optimized for plant genomes, utilizing long-range information. | Plant genomes | OLC | Mandáková T et al., Nature Communications, 2019. |
| **Cerulean** | Enhances assemblies by integrating high-quality short reads with long reads. | General-purpose | De Bruijn graph | Deshpande V et al., Nature Methods, 2013. |


**Key Output Files**

**1. spades.log** – Log file with details on the assembly process.

**2. contigs.fasta** – Assembled contigs.

**3. scaffolds.fasta** – Assembled scaffolds.

**4. assembly_graph.fastg** – Assembly graph.

**5. stats.tsv** – Summary statistics of the assembly.

# ASSEMBLY –QUALITY CHECK

Many tools can produce a QC that can be used to assess the state of your assembly (for example, Quast)

**Which one is better?**

N50: It represents the length of the shortest contig (or scaffold) in the set that contains at least 50% of the total assembly length. It provides an indication of the continuity of the assembly.

Report

| | final.contigs |
|---|---|
| # contigs (>= 0 bp) | 1299 |
| # contigs (>= 1000 bp) | 1101 |
| # contigs (>= 5000 bp) | 830 |
| # contigs (>= 10000 bp) | 600 |
| # contigs (>= 25000 bp) | 297 |
| # contigs (>= 50000 bp) | 81 |
| Total length (>= 0 bp) | 21549110 |
| Total length (>= 1000 bp) | 21436012 |
| Total length (>= 5000 bp) | 20718103 |
| Total length (>= 10000 bp) | 18993070 |
| Total length (>= 25000 bp) | 14095881 |
| Total length (>= 50000 bp) | 6605648 |
| # contigs | 1214 |
| Largest contig | 266122 |
| Total length | 21516804 |
| GC (%) | 61.48 |
| N50 | 34088 |
| N90 | 9074 |
| auN | 50048.1 |
| L50 | 183 |
| L90 | 640 |
| # N's per 100 kbp | 0.00 |

# SUMMARY

| Característica | OLC | De Bruijn Graph (DBG) |
|---|---|---|
| **Manejo de repeticiones cortas** | Bueno: Lecturas largas abarcan las repeticiones. | Muy bueno: Repeticiones <mk−mers se resuelven bien. |
| **Manejo de repeticiones largas** | Excelente: Lecturas largas resuelven regiones complejas. | Pobre a moderado: Repeticiones > lecturas o k−colapsan nodos. |
| **Tamaño de lectura ideal** | Lecturas largas (PacBio, ONT). | Lecturas cortas (Illumina). |
| **Escalabilidad computacional** | Requiere más memoria y tiempo. | Más eficiente y escalable para datos masivos. |
| **Precisión en regiones complejas** | Alta: Adecuado para genomas con mucha variabilidad. | Baja a moderada: Depende del kkk y la calidad de datos. |
| **Adaptabilidad a datos ruidosos** | Bueno: Las lecturas largas toleran errores. | Menor tolerancia: Errores en lecturas afectan nodos. |
| **Dependencia de kkk** | No depende de kkk. | Alta: kkk debe ajustarse cuidadosamente. |
| **Uso típico** | Ensamblajes de novo con lecturas largas. | Ensamblajes masivos con lecturas cortas. |
| **Tecnologías compatibles** | PacBio, Oxford Nanopore. | Illumina y tecnologías de lecturas cortas. |

# OUTLINE

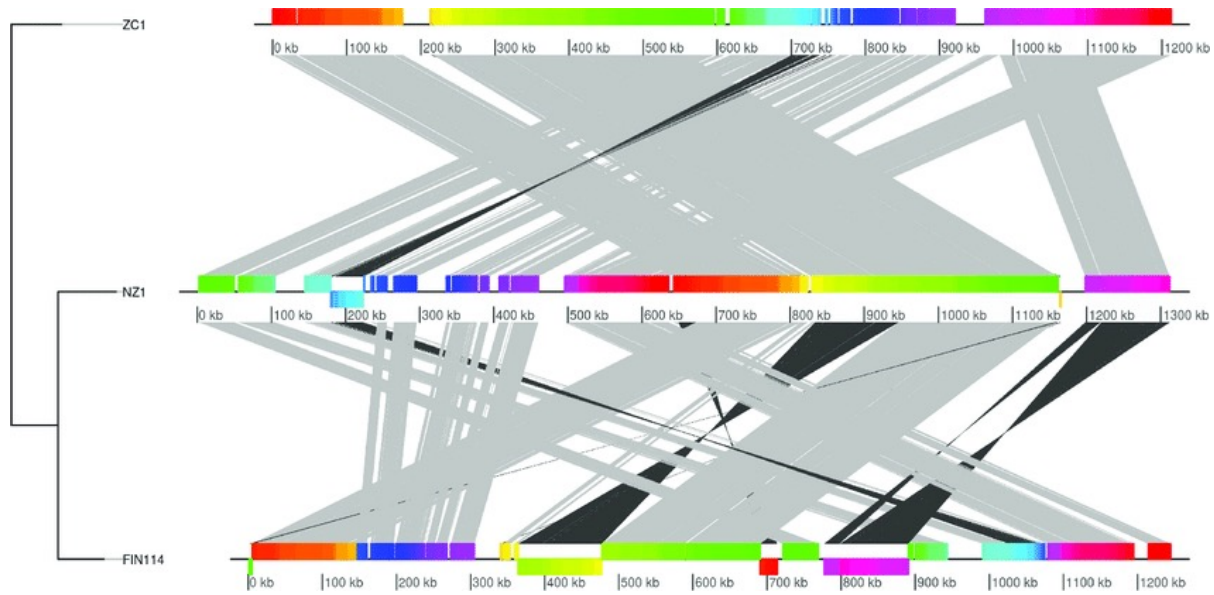
MICROBIOME TECHNOLOGIES & ANALYSIS I

1. (meta)Genome assembly -> (Li et al., 2011)
    a. Overlap-layout-consensus
    b. String graphs (de Bruijn graphs)
2. Whole genome alignments -> (Ahmed et al., 2019)
    a. Global
    b. Local
    c. Glocal
3. Phylogeny -> (Aniscar et al, 2020)

# 2. WHOLE-GENOME ALIGNMENT

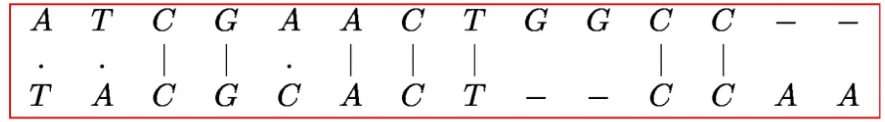
# WHAT IS SEQUENCE ALIGNMENT?

Sequence alignment is the procedure of comparing two (pair-wise alignment) or more multiple sequences by searching for a series of individual characters or patterns that are in the same order in the sequences



# 2. THESE ALIGNMENT ALGORITHMS CAN BE DIVIDED INTO THE FOLLOWING CLASSES:

***2a. Global alignment*:** also known as end-to-end alignment, the goal is to align the sequences in their entirety while maximizing the alignment score. (Needleman-Wunsch algorithm)

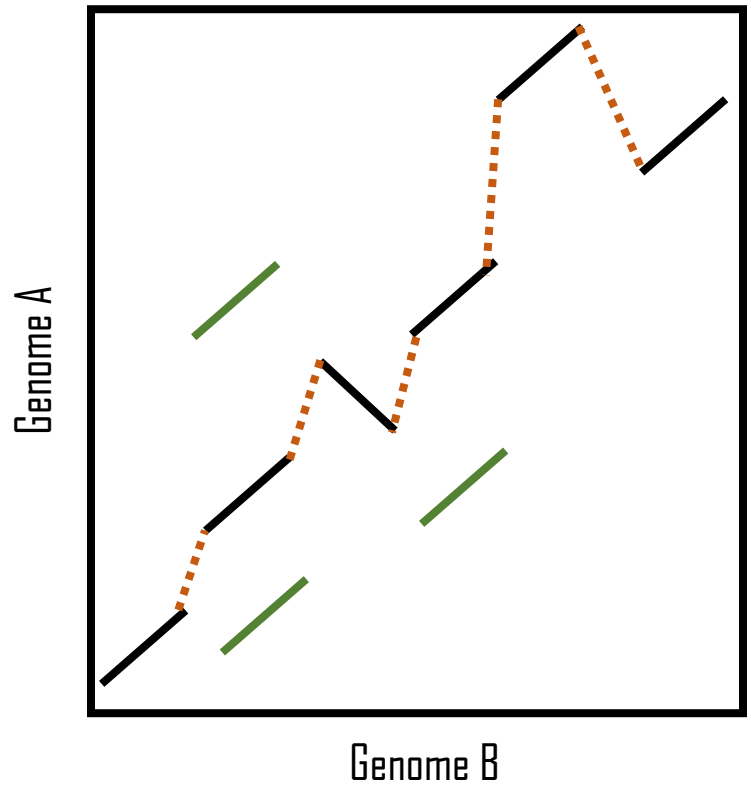



***2b. Local alignment*:** In local alignment, the goal is to align two sequences so that the alignment score is maximized. As opposed to global alignment, the final alignment may not contain the whole of the sequences. No penalty is induced by misalignments in the beginning and end of the sequences, and the score is kept positive. (Smith-Waterman algorithm)


Ahmed et al., 2019, BMC Bioinformatics

# 2C. GLOCAL ALIGNMENT

Glocal alignment allows for the possibility of duplications, inversion, and translocations. Finds the least cost transformation of one sequence into another using new operations



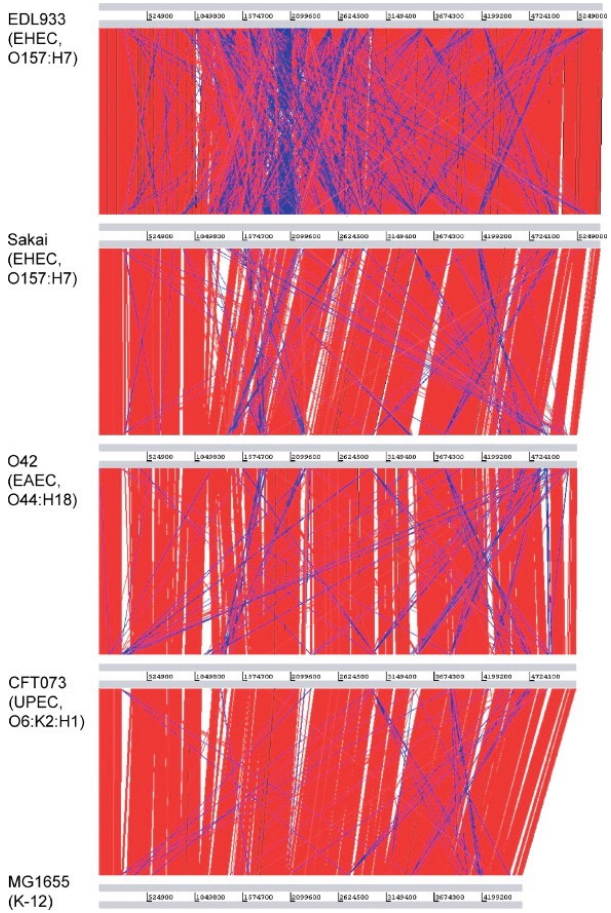
# 2C. EXAMPLES

| Alignment Type | Definition | Use Case | Algorithm | Example Scenario |
|---|---|---|---|---|
| Global Alignment | Aligns sequences from start to end. | Whole sequence comparison. | Needleman-Wunsch | Comparing full-length genes. |
| Local Alignment | Finds most similar regions. | Detecting conserved motifs. | Smith-Waterman | Identifying functional domains. |
| Glocal Alignment | One sequence globally aligned, the other locally. | Aligning short reads to genomes. | Modified NW/SW | Mapping sequencing reads. |

# 2. ALIGNMENT TOOLS (full genomes)

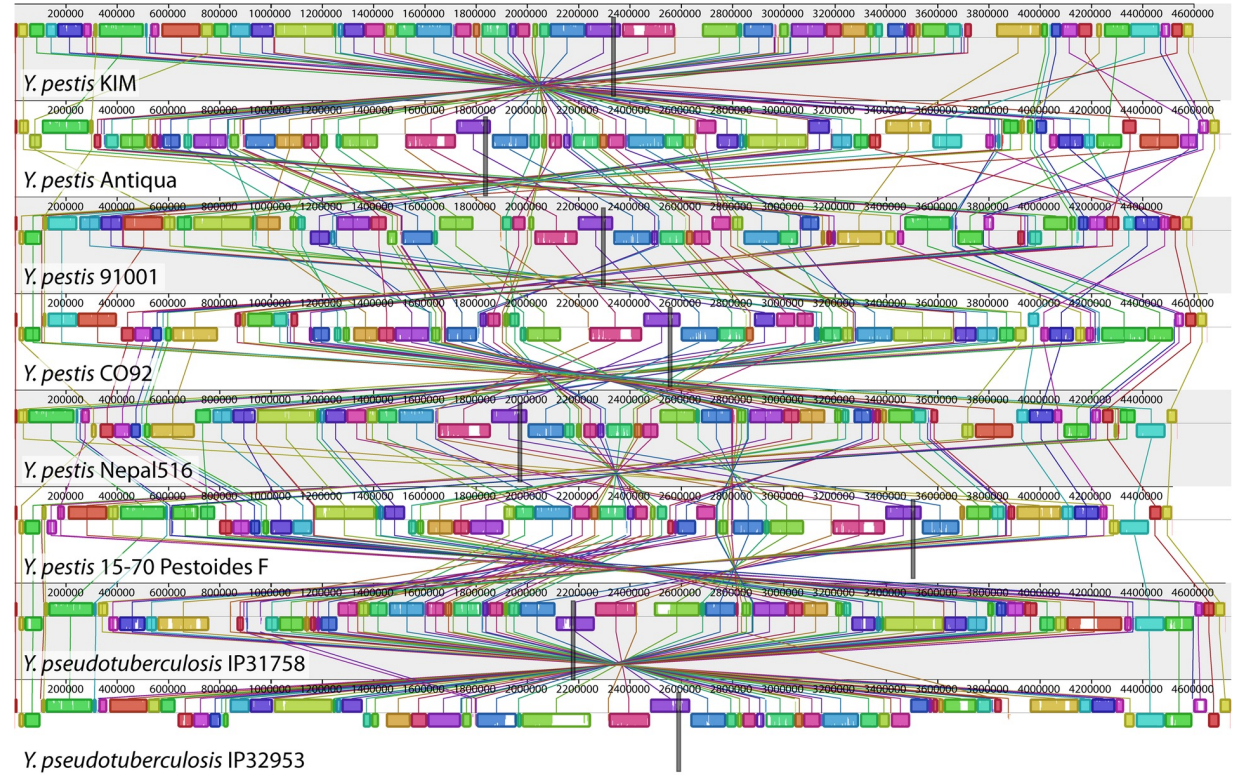
Max 500k positions/500 sequences

## WebACT




https://www.sanger.ac.uk/tool/artemis-comparison-tool-act/

## mauve




https://darlinglab.org/mauve/mauve.html

# 2. ALIGNMENT TOOLS (short genes)

Max 500k positions/500 sequences

| Method | Command |
|---|---|
| ClustalW | Clustalw |
| ClustalWqt2 | clustalw –quicktree |
| Dialign | Dialing |
| Dialign_fast | dialign –o |
| FFT_NS_2 | Mafft –retree 2 |
| FFT_NS_i | Mafft –maxiterate 1000 |
| G_INS_i | Mafft –globalair –maxiterate 1000 |
| Kalign2 | kalign2 |
| Kalign1 | kalign1 |
| L_INS_i | Mafft –localpair –maxiterate 1000 |
| Muscle | muscle |
| Muscle_fast | muscle -maxiters 1 -diags -sv -distance1 kbit20_3 |
| Parttree-1-1000 | mafft –retree 1 –parttree –partsize 1000 |
| Parttree-2-1000 | mafft –retree 2 –parttree –partsize 1000 |
| Probcons_fast | probcons -ir 0 |
| Probcons | Probcons |
| ProbconsRNA | probconsRNA |
| T_Coffee | t_coffee |
| T_Coffee_fast | t_coffee -special_mode quickaln |

Lassmann et al., 2009

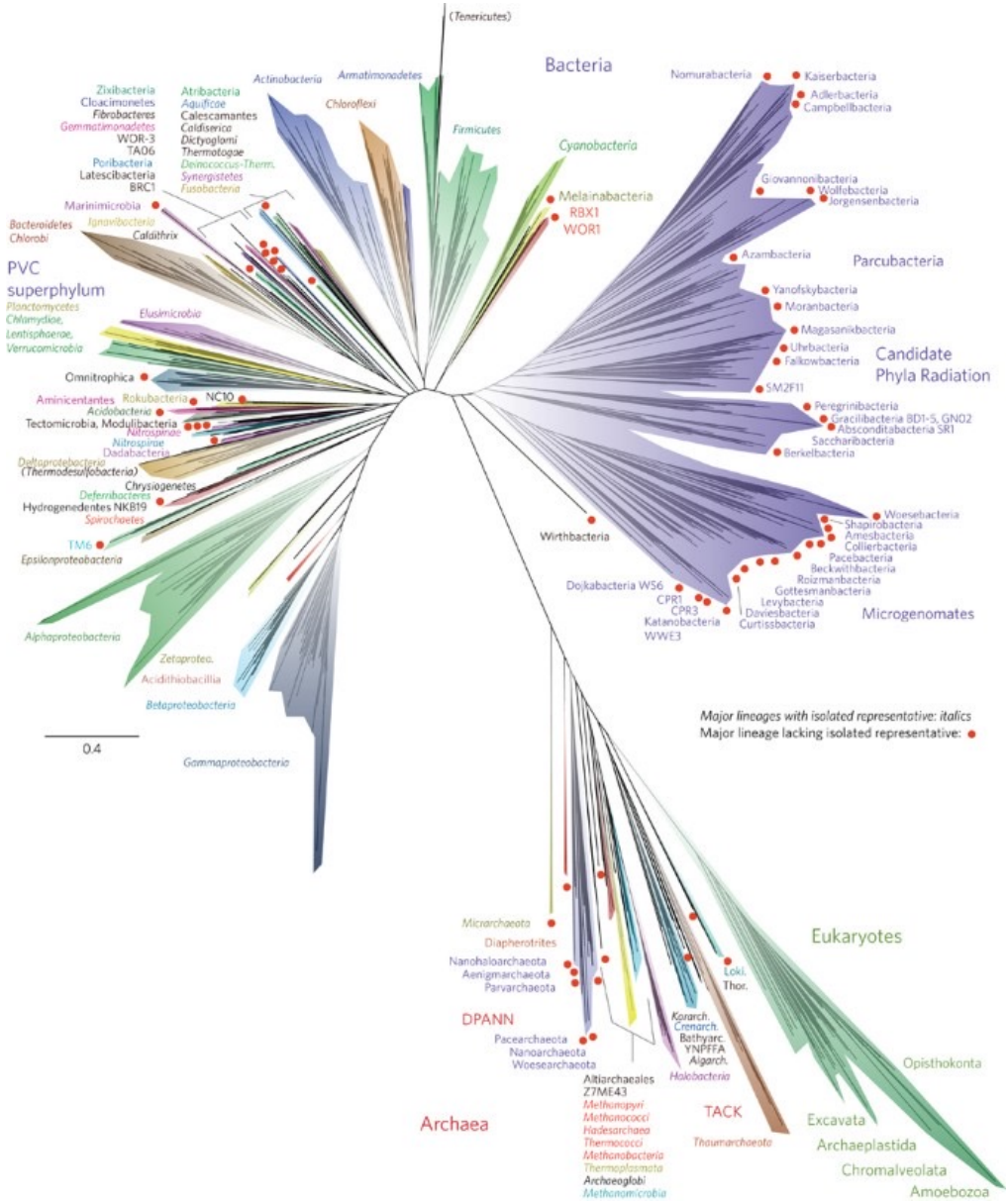
# OUTLINE

MICROBIOME TECHNOLOGIES & ANALYSIS I

1. (meta)Genome assembly -> (Li et al., 2011)
    a. Overlap-layout-consensus
    b. String graphs (de Bruijn graphs)
2. Whole genome alignments -> (Ahmed et al., 2019)
    a. Global
    b. Local
    c. Glocal
3. Phylogeny -> (Aniscar et al, 2020)

# 3. PHYLOGENY

# WHICH GENES SHOULD I USE FOR PHYLOGENY?




- Essential gene widely distributed in the tree of life
- Sequence divergence for higher resolution

  Widely distributed

  Low resolution at species level

Ribosomal subunit 16S

# PHYLOGENY (ALL IN ONE)

**AMPHORA2**
Wu et al, 2012

139 markers from an orthology analysis

**Phylophlan3**
F Aniscar et al, 2020

UniRef90 families present in >75% in 150,000 MAGs and 80,000 genomes

**AutoMLST**
M Alanjari et al, 2019

Essential genes from a PFAM-TIGRFAM analysis

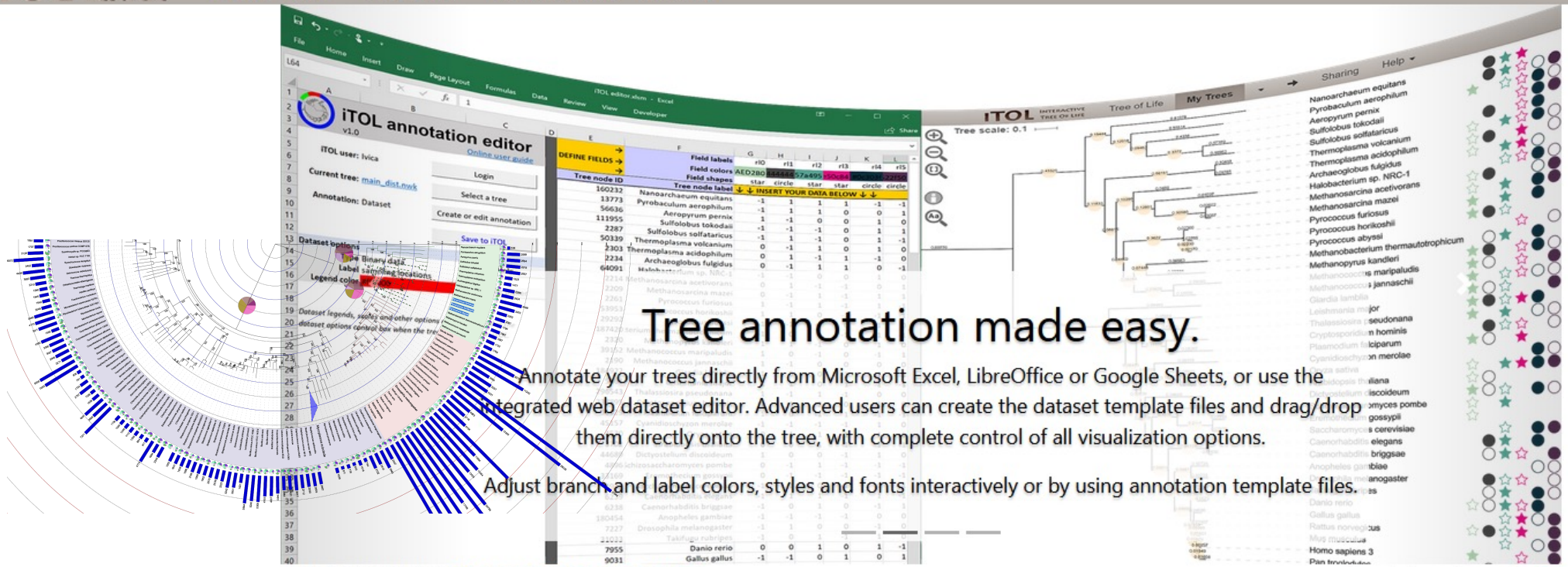
Ondov *et al. Genome Biology* (2016) 17:132
DOI 10.1186/s13059-016-0997-x

Genome Biology

SOFTWARE Open Access

## Mash: fast genome and metagenome distance estimation using MinHash

Brian D. Ondov[1], Todd J. Treangen[1], Páll Melsted[2], Adam B. Mallonee[1], Nicholas H. Bergman[1], Sergey Koren[3] and Adam M. Phillippy[3*]




# Tree annotation made easy.

Annotate your trees directly from Microsoft Excel, LibreOffice or Google Sheets, or use the integrated web dataset editor. Advanced users can create the dataset template files and drag/drop them directly onto the tree, with complete control of all visualization options.

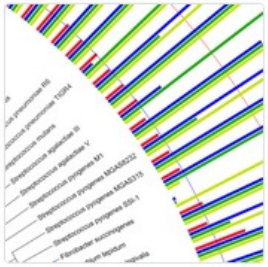
Adjust branch and label colors, styles and fonts interactively or by using annotation template files.

**Note:** See the details on iTOL access modes and subscriptions

Current changelog: version 6.6

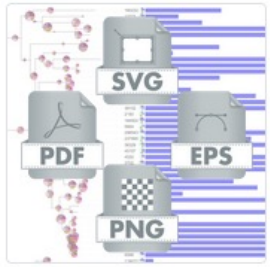
## Manage

Organize your trees into workspaces and projects, and access them from any browser. Simply drag and drop multiple tree files onto a



## Annotate

19 dataset types. Full control over branch colors, widths and styles. Individually adjustable label fonts, sizes and styles. Check our gallery of user




## Export

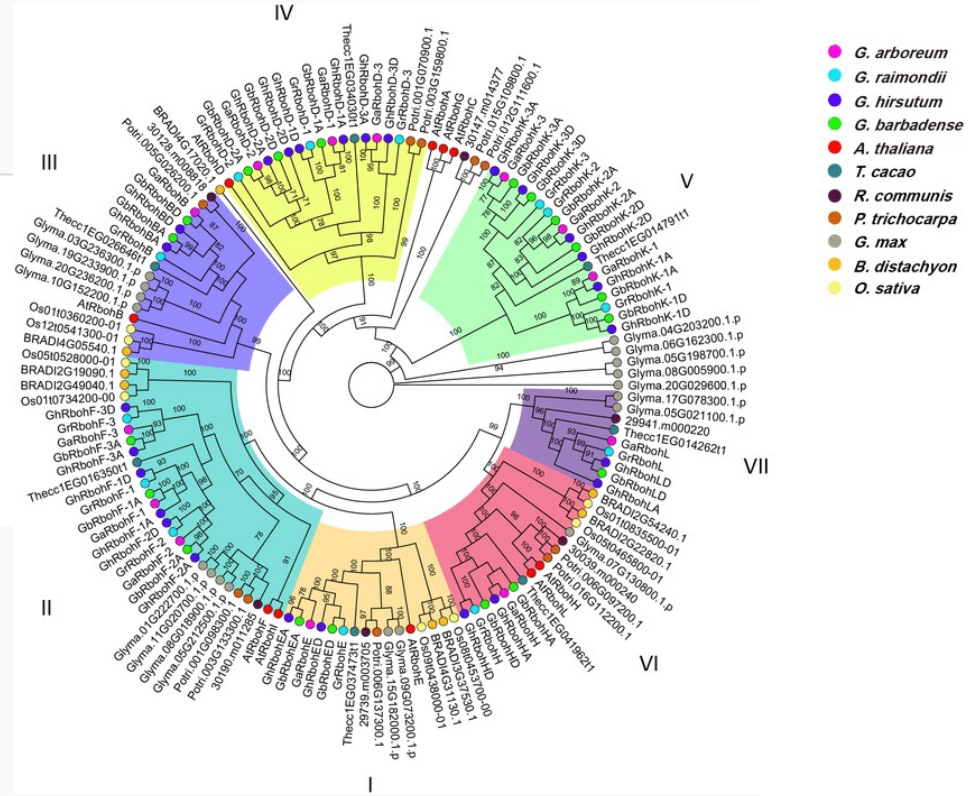
Create high quality tree figures for your publications. Direct What-You-See-Is-What-You-Get export of what is displayed on the screen.

**Citation:** Letunic and Bork (2021) *Nucleic Acids Res* **doi: 10.1093/nar/gkab301** | Privacy Policy

# GraPhlAn

GraPhlAn is a software tool for producing high-quality circular representations of taxonomic and phylogenetic trees. GraPhlAn focuses on concise, integrative, informative, and publication-ready representations of phylogenetically- and taxonomically-driven investigation.

User manual || Tutorial || Forum




# REQUIREMENTS

1. python (version >= 2.7)
2. the biopython python library ( version >=1.6)
3. the matplotlib python library (version >=1.1)

# GETTING STARTED

## Installation

GraPhlAn is available in bitbucket and should be obtained using Github.
download the GraPhlAn software from the repository:

## SUMMARY

1. First steps when our (meta)Genome is sequenced
2. (meta)Genome assembly methods & tools
3. Types of alignments & tools
4. Phylogeny & tools