

# Genome assembly

EBAll Assembly & Annotation - Roscoff June 2024

Delphine NAQUIN  
Christophe KLOPP

# What are you going to learn?

- What a genome assembly is.
- What a genome assembler is.
- Which assembly strategies can be used.
- Which are the most common genome assemblers used these days.
- Which data/assembler combination work.
- How to assemble a bacterial genome.
- How to perform eukaryote genome assembly with several software packages in Galaxy (TP).
- What are the most used parameters.

# What is a genome assembly?

A genome assembly is a set of sequences, usually in **fasta format**, representing the genome content **at the nucleotide level**.

Today it is very rare to have a chromosome in a single read. Therefore we assemble a given read **coverage** ( $nX$ ) to generate a genome assembly.

Depending on the definition assembly builds **contigs** or **scaffolds** or **chromosomes** (pseudo-molecules). Here we will stick to contigs even if some software packages perform scaffolding. For **polyploid organisms** the assembler can output  $N$  contig sets : **haplotyped assemblies**.

Assembly sequence correction, called “**polishing**” will be presented in the next section.

# Procaryote genome assembly

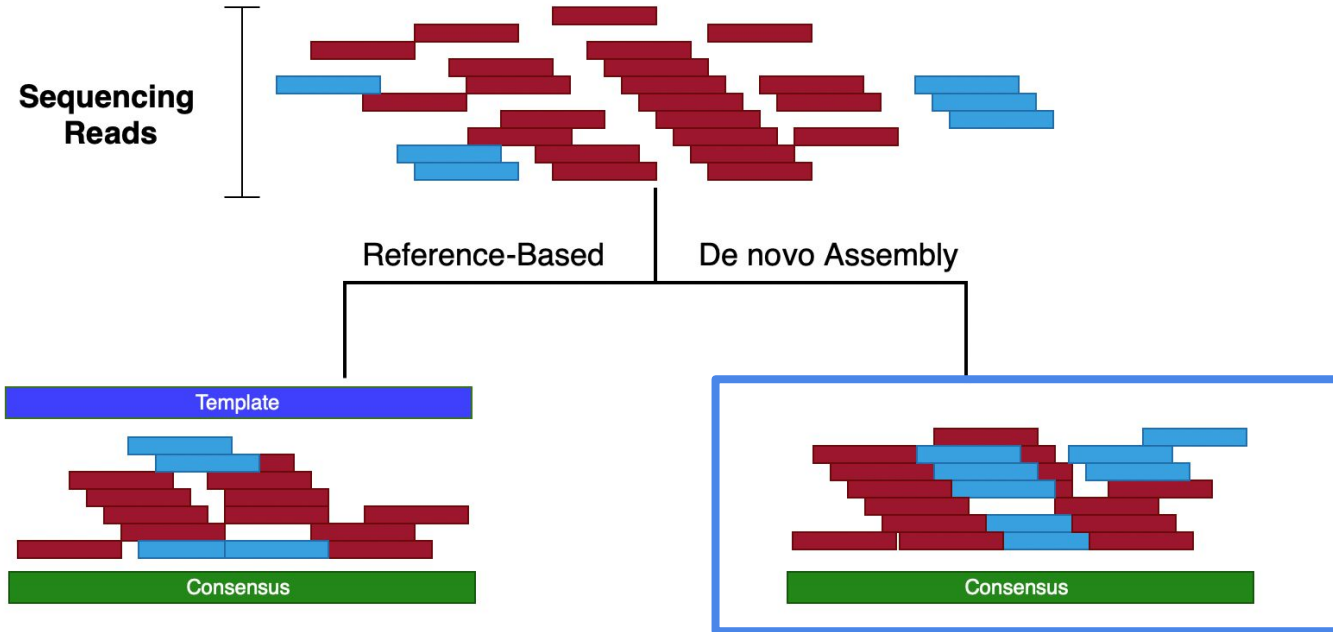
easy (compared to eucaryote) :

- smaller ( $< 12$  Mb)
- less repetition (longest  $< 10$  Kb)
- haploid

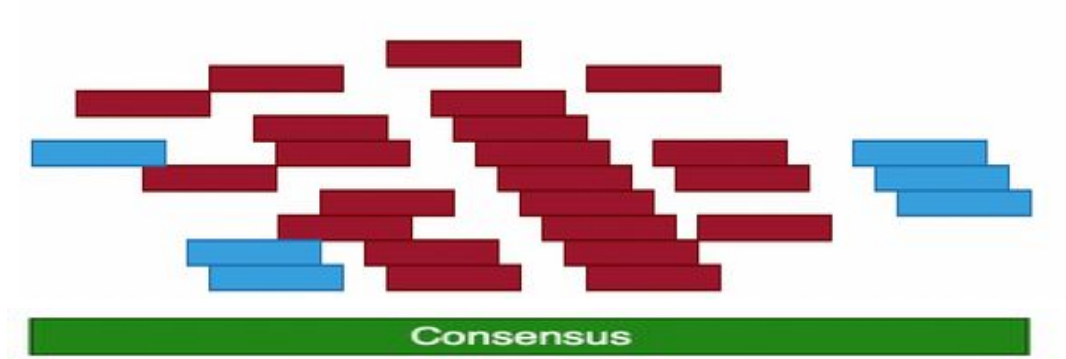
but :

- presence of a plasmid whose copy number differs from that of the chromosome -> different coverage
- circular but without a clear beginning or end

# Reference based vs de novo assembly



# What is the problem?



No unique read yet covering the complete chromosome.

Strategy : bridging chromosome with several reads using sequence similarities to organize them.

# What is a de novo genome assembler?

It is a piece of software taking reads as input and producing a set of contigs or scaffolds representing the genome content at the nucleotide level.

There are several categories of assemblers depending on the read length :

- short read assemblers
- long read assemblers
- hybrid assemblers

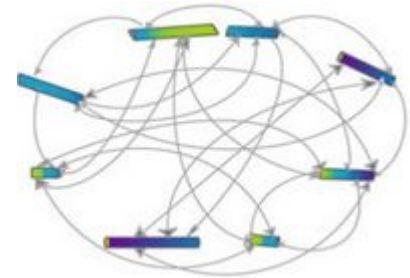
# Assembler algorithms

Several genome assembly algorithms have been imagined :

- greedy
- OLC (Overlap Layout Consensus)
- DBG (de Bruijn graph)
- string graph
- repeat graph

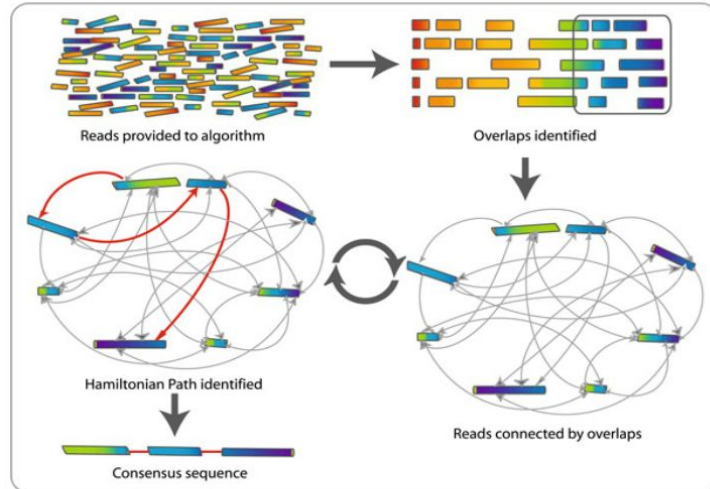
The most used ones are OLC and DBG.

Both represent the assembly as a graph with nodes and edges.



# OLC : Overlap Layout Consensus

1. Identify all overlaps (all versus all read matching)
2. Organize reads and their overlaps into a graph
3. Find a single paths that explores all **nodes** exactly once.



# de Bruijn graph



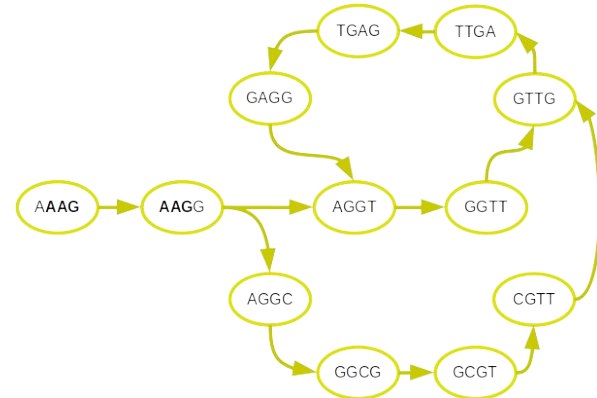
1. Construct the  $k$ -mers graph of the reads (substring of length  $k$ )
  - nodes: all  $k$ -mers present in the reads
  - a link connects 2 nodes if an overlap of length  $k-1$  exists between the 2  $k$ -mers.
2. Find a path that crosses all nodes at least once

Short read to  $k$ -mers ( $k=4$ )

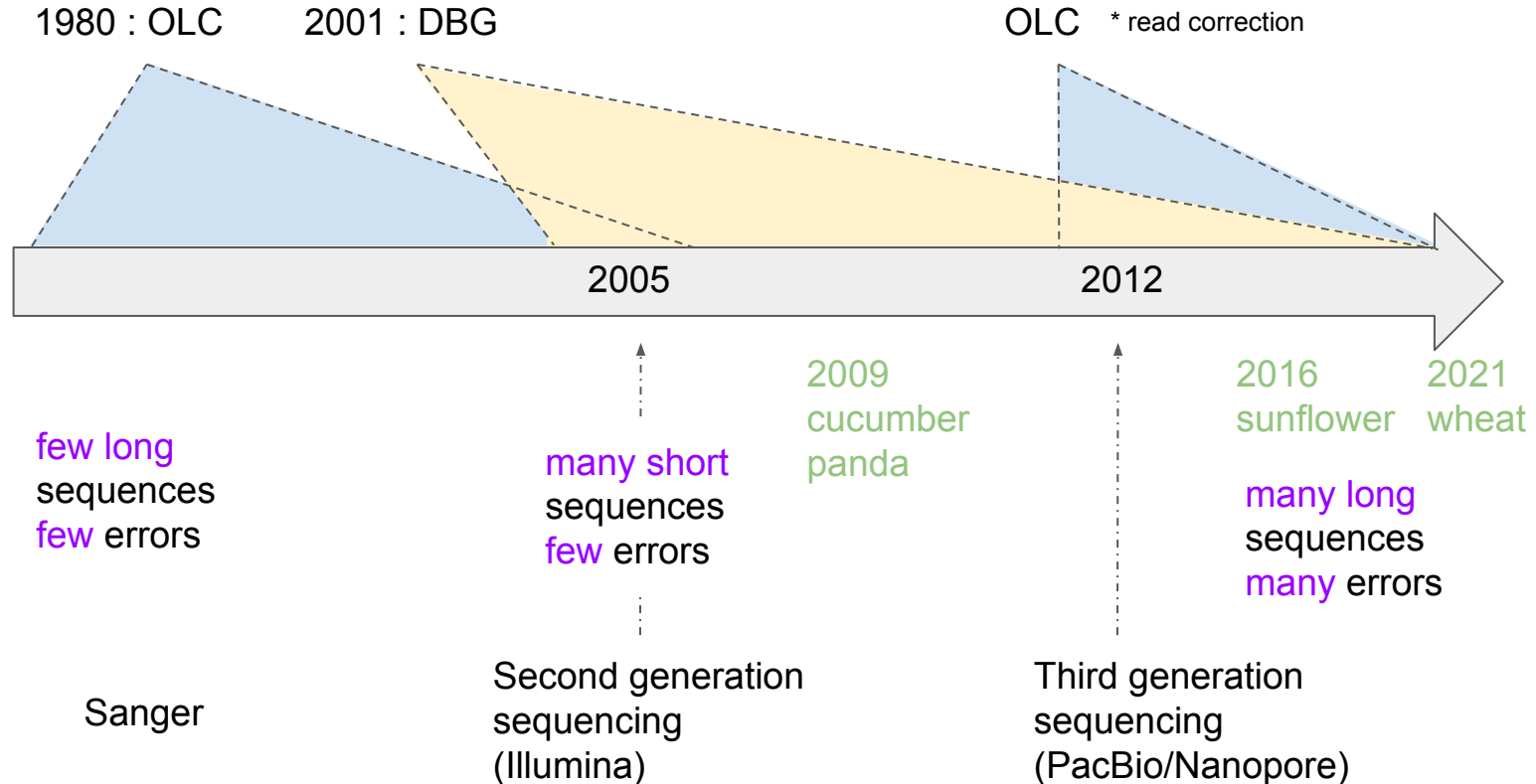
**AAAGGCGTTGAGGTT**

AAAG  
AAGG  
AGGC  
GGCG  
GCGT  
CGTT  
GTTG  
TTGA  
TGAG  
GAGG  
AGGT  
GGTT

## de Bruijn graph



# OLC and DBG over time



# Commonly used assemblers

Name	short reads	ONT, CLR reads	hifi reads	algorithm	polishing	scaffolding
<b>SPAdes</b>	X			DBG		X
<b>(Uni-Try)cycler (SPAdes, miniasm+Racon)</b>	X	X		DBG first	X	
<b>Flye</b>		X	X	OLC	X	X
<b>wtdbg2 (redbean)</b>		X	X	DBG		
<b>(Hi)Canu</b>		X	(X)	OLC		
<b>hifiasm</b>	haplotyping		X	OLC		
<b>NextDeNovo</b>		X	(X)	OLC		

# Other assemblers

**soapdenovo (DBG)** : short reads, quick and low memory consumption, short contigs

**smartdenovo (OLC)** : ONT & CLR long reads, redbean ancestor : same ideas

**MaSuRCA (Hybrid)** : long read correction with short reads

**Verkko (OLC)** : can use HiFi reads and UL ONT reads to produce T2T assembly

But also : **Peregrin, NECAT,...**

# Read length and quality / assembler combination

- Some assembler can work with different data types : usually low quality data assembler can also run on high quality data.
- Assemblers are often long reads or short reads specific. Some combine both data type for read correction but not for assembly (with some exception).
- Check if the assembler is adapted for your data and use the corresponding parameter(s).

# What should you take into account?

## Read type :

- length
- error rate

**Sequencing depth** : too low depth = fragmented assembly, too high depth can also impair assembly metrics

**Genome repeat fraction and repeat structure** : large and very similar repeats are difficult to assemble. Long high quality reads will enable to build through repeats using few variations.

**Heterozygosity** : high heterozygosity will render the assembly more difficult and you will need [more read coverage](#)

**Recent whole genome duplication and auto-polyploidy** : multiple copies of the same genome part is not taken into account by the assemblers.

**Partial endoreplication** : having parts of the genome more represented than others is not taken into account by the assemblers.

# Some advices

## **Know your genome! remember genomescope2!**

- size
- heterozygosity
- repeat content

**Try different assemblers** : large genome assembly could take weeks and sometimes months a few years ago. Now it is hours or sometimes days. Still you should try different assemblers. Try at least two.

**Do not use too much data** : Assembler have an optimal coverage range in which they perform best. Assembly metrics are going to worsen with too much data.

**Do not stick to N50** : It is better to have a lower N50 with less assembly error than the opposite. Check your assembly versus the reads and/or a reference when available. Check transcript content : BUSCO, RNA-Seq de novo contig alignment, ..

# What should you expect?

Assembly length should be close to genome size. With error prone reads you expect repeat compression and therefore a **smaller** assembly size. For heterozygous genomes you can find **longer** assemblies than expected because both haplotypes have been kept. This should be checked with kmers and corrected with `purge_dups` or `purge_haplotigs`.

Contig N50 with a correct read depth (**50X** short reads, CLR or ONT and **20X** HiFi or Q20+):

- with short reads only : 10Kb to 200Kb depending on the repeat content and genome size
- with long reads 2Mb to 50Mb depending on heterozygosity, repeat content

Average contig coverage (Nx) should be close to your sequencing coverage for most of the contig, mainly the long ones.

Most of the non error kmer found in the reads should be present in the assembly (in both haplotypes for diploid species).

# Assembler parameters

Parameters are different between assemblers.

Categories :

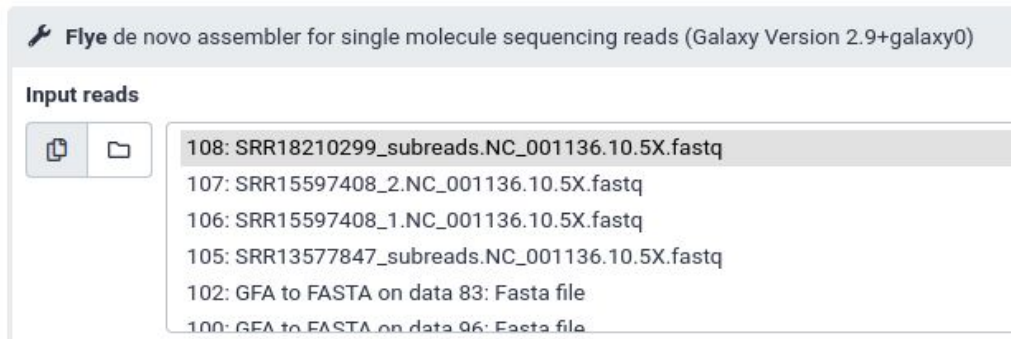
- performance related (CPU, memory)
- genome information (genome size)
- **read type** : when accepting different types
- coverage related : min coverage to keep links between reads, expect cov
- overlap related : when should two reads be seen as having an overlap
- assembly related : graph pruning, type of output : primary, haplotypes
- haplotyping related : Hi-C, trio data
  - purge related : removing duplicated contigs

# How to assemble a bacterial genome

- long reads sequencing !
- short reads sequencing
- if the long reads coverage is greater than  $\sim 60X$ , filter reads on **size** and **quality** (NanoFilt, Filtlong)
- run 2 or 3 different assemblers (specific tool : plasmidSPAdes)
- metrics comparison (N50, # contigs, assembly size...)
- choose the best assembly
- polishing the best assembly (or all and compare the BUSCO scores, reads mapping)

# Running an assembly in Galaxy

- Upload your data (usually a fastq file) or have access to it locally.
- select the assembler in the software package list (on the left).
- select your dataset in the list (available fastq datasets)
- set parameters (usually first run with default)
- hit the “execute” button



# Running a flye genome assembly in usegalaxy.fr

**Galaxy France**

Workflow Visualize Données partagées Aide Utilisateur

Tools

flye

Upload Data

Show Sections

Flye de novo assembler for single molecule sequencing reads

WORKFLOWS

All workflows

**Flye de novo assembler for single molecule sequencing reads (Galaxy Version 2.9+galaxy0)**

Input reads

- 108: SRR18210299\_subreads.NC\_001136.10.5X.fastq
- 107: SRR15597408\_2\_NC\_001136.10.5X.fastq
- 106: SRR15597408\_1\_NC\_001136.10.5X.fastq
- 105: SRR13577847\_subreads.NC\_001136.10.5X.fastq
- 102: GFA to FASTA on data 83: Fasta file
- 100: GFA to FASTA on data 06: Fasta file

Mode

Nanopore raw (-nano-raw)

Number of polishing iterations

1

Polishing is performed as the final assembly stage. By default, Flye runs one polishing iteration. Additional iterations might correct a small number of extra errors (due to improvements on how reads may align to the corrected assembly). If the parameter is set to 0, the polishing is not performed (-iterations)

Minimum overlap between reads

3

This sets a minimum overlap length for two reads to be considered overlapping. By default it is chosen automatically based on the read length distribution (reads N90) and does not require manual setting. Typical value is 3k-5k (and down to 1k for datasets with shorter read length). Intuitively, we want to set this parameter as high as possible, so the repeat graph is less tangled. However, higher values might lead to assembly gaps. In some rare cases it makes sense to manually increase minimum overlap for assemblies of big genomes with long reads and high coverage. (-min-overlap)

Keep haplotypes

No

By default, Flye collapses graph structures caused by alternative haplotypes (bubbles, superbubbles, roundabouts) to produce longer consensus contigs. This option retains the alternative paths on the graph, producing less contiguous, but more detailed assembly. (-keep-haplotypes)

Enable scaffolding using graph

No

Starting from the version 2.9 Flye does not perform scaffolding by default, which guarantees that all assembled

Perform metagenomic assembly

No

It is designed for highly non-uniform coverage and is sensitive to underrepresented sequence at low coverage (assembled more contiguous bacterial consensus sequence, while the metagenome mode was slightly more fragmented)

Reduced contig assembly coverage

Disable reduced coverage for initial disjointing assembly

Typically, assemblies of large genomes at high coverage require a hundreds of RAM. For high coverage assemblies, the memory bottleneck is usually at the initial disjointing stage (usually, the memory bottleneck)

Generate a log file

No

Email notification

No

Send an email notification when the job completes.

Execute

Purpose

History

Rechercher des données

EBAI Assembly

74 shown, 35 deleted, 1 hidden

2.84 GB

- 84: Hifiasm on data 74: alternate assembly contig graph
- 83: Hifiasm on data 74: primary assembly contig graph
- 82: Hifiasm on data 74: processed unitig graph
- 81: Hifiasm on data 74: haplotype-resolved raw unitig graph
- 80: SPAdes on data 76 and data 75: Scaffolds
- 79: SPAdes on data 76 and data 75: Contigs
- 78: SPAdes on data 76 and data 75: Assembly graph with scaffolds
- 77: SPAdes on data 76 and data 75: Assembly graph
- 76: SRR15597408\_2\_NC\_001136.10.70X.fastq
- 75: SRR15597408\_1\_NC\_001136.10.70X.fastq
- 74: SRR13577847\_subreads.NC\_001136.10.70X.fastq
- 73: WTBG2 on data 72: assembled contigs
- 72: SRR18726953\_1\_NC\_001136.10.70X.fastq

# Conclusions

**Running an assembly is easy now.**

**Know your genome** before producing your data and adapt your data production (data type, coverage) to your genome characteristics.

With the right data type(s) genome assembly is now usually easy for genomes up to 1Gb. This kind of assemblies can now be performed by a **single scientist** using a large enough computing infrastructure.

Generate **several** assemblies (different software packages, different coverages (nX),...) and **compare them** to select the best.

Check your assemblies (metrics, protein content, organelles, telomeric repeats,...)

# Hands-on with *S. cerevisiae* (~12 Mb, 16 chromosomes)

You will split in two teams (or more).

Your missions is to perform, compare and give information about different assemblies :

- using different data types (short reads (MiSeq PE), ONT reads (R9.4), HiFi reads),
- different coverages (from 5X to 30X),
- different software packages (SPAdes, flye, wtdbg2, Hifiasm).

You will compare the results using quast and dgenies (<https://dgenies.toulouse.inra.fr/>).

You will fill the table : <https://lite.framacalc.org/f952lz03mb-9vey>

You will present you findings (impact of data types, coverage, software package,...) : 5 minutes per team

# SPAdes parameters

Tools

spades

Upload Data

Show Sections

rnaSPAdes de novo transcriptome assembler

rnaviralSPAdes de novo assembler for transcriptomes, metatranscriptomes and metaviromes

SPAdes genome assembler for genomes of regular and single-cell projects

metaplasmidSPAdes extract and assembly plasmids from metagenomic data

metaSPAdes metagenome assembler

metaviralSPAdes extract and assembly viral genomes from metagenomic data

plasmidSPAdes extract and assembly plasmids from WGS data

biosyntheticSPAdes biosynthetic gene cluster assembly

coronaSPAdes SARS-CoV-2 de novo genome assembler

Create assemblies with Unicycler

Bandage Image visualize de novo assembly graphs

Bandage Info determine statistics of de novo assembly graphs

WORKFLOWS

All workflows

SPAdes genome assembler for genomes of regular and single-cell projects (Galaxy Version 3.15.4+galaxy1)

Operation mode

Assembly and error correction

To run read error correction, reads should be in FASTQ format.

Single-end or paired-end short-reads

Single-end

It assumes that all samples belong to the same library. If you want to use samples from two different libraries, include the second library as additional set of short-reads.

Please provide a value for this option.  
FASTA/FASTQ file(s)

No fastq, fastq.gz, fastqsanger.gz, fasta or fasta.gz dataset available.

Use an additional set of short-reads

Disabled

Enable this option if you want to combine to data sources (e.g. single and paired reads).

Additional read files

Pipeline options

☐ Select/Unselect all

☐ Disable repeat resolution (--disable-r)

☐ Single cell mode: required for MDA (single-cell) data (--sc)

☐ Isolate: highly recommended for high-coverage isolate and multi-cell data (--isolate)

☐ Careful: ties to reduce the number of mismatches and short indels. Only recommended for small genomes (--careful)

☐ Iontorrent: required when assembling IonTorrent data (--iontorrent)

Error correction requires FASTQ input files.

Set coverage cutoff option

Off

When set to 'auto' SPAdes automatically computes coverage threshold using conservative strategy (--cov-cutoff)

Select k-mer detection option

Auto

If --sc is set the default values are 21,33,55. For multicell datasets K values are automatically selected using maximum read length. Comma-separated list, all values must be odd, less than 128 and listed in ascending order. (-k)

Set Phred quality offset

Auto

# Flye parameters

Tools

flye

Upload Data

Show Sections

Flye de novo assembler for single molecule sequencing reads

WORKFLOWS

All workflows

Flye de novo assembler for single molecule sequencing reads (Galaxy Version 2.9+galaxy0)

Please provide a value for this option.

Input reads

No fasta, fasta.gz, fastq, fastq.gz, fastqsanger.gz or fastqsanger dataset available.

Mode

Nanopore raw (~nano-raw)

Number of polishing iterations

1

Polishing is performed as the final assembly stage. By default, Flye runs one polishing iteration. Additional iterations might correct a small number of extra errors (due to improvements on how reads may align to the corrected assembly). If the parameter is set to 0, the polishing is not performed (~iterations)

Minimum overlap between reads

This sets a minimum overlap length for two reads to be considered overlapping. By default it is chosen automatically based on the read length distribution (reads N90) and does not require manual setting. Typical value is 3k-5k (and down to 1k for datasets with shorter read length). Intuitively, we want to set this parameter as high as possible, so the repeat graph is less tangled. However, higher values might lead to assembly gaps. In some rare cases it makes sense to manually increase minimum overlap for assemblies of big genomes with long reads and high coverage. (~min-overlap)

Keep haplotypes

No

By default, Flye collapses graph structures caused by alternative haplotypes (bubbles, superbubbles, roundabouts) to produce longer consensus contigs. This option retains the alternative paths on the graph, producing less contiguous, but more detailed assembly. (~keep-haplotypes)

Enable scaffolding using graph

No

Starting from the version 2.9 Flye does not perform scaffolding by default, which guarantees that all assembled sequences do not have any gaps (~scaffold)

Perform metagenomic assembly

No

It is designed for highly non-uniform coverage and is sensitive to underrepresented sequence at low coverage (as low as 2x). In some examples of simple metagenomes, we observed that the normal mode assembled more contiguous bacterial consensus sequence, while the metagenome mode was slightly more fragmented, but revealed strain mixtures (~meta)

Reduced contig assembly coverage

Disable reduced coverage for initial disjointing assembly

Typically, assemblies of large genomes at high coverage require a hundreds of RAM. For high coverage assemblies, you can reduce memory usage by using only a subset of longest reads for initial contig extension stage (usually, the memory bottleneck)

Generate a log file

No

Execute

# wtdbg2 parameters

Tools

wtdbg2

Upload Data

Show Sections

WTDBG2 Fast de novo sequence assembler for long noisy reads

WORKFLOWS

All workflows

Sequencing technology used to generate reads

Genome size

Estimated genome size. k/m/g suffix is allowed - eg a 4500000bp ecoli genome can be written as 4.5m. For a human genome, use 3.2g

Assembly Options

Read depth

50.0

(-X) [float] Choose the best [float] depth from input reads. ie if the estimated genome size is 5m, setting this value to 50.0 would select the best 2.5mb worth of reads.

Min read length

0

(-L) [int] Choose the longest subread and drop reads shorter than [int]

Kmer size

0

(-k) [int] Kmer size,  $0 \leq k \leq 23$

Homopolymer-compressed kmer size

21

(-p) [int] Homopolymer-compressed kmer size,  $0 \leq p \leq 23$

Max kmer frequency

1000.0

(-K) [float] Filter high frequency kmers where frequency > [float]

Min read similarity

0,05

(-s) [float] Min similarity between reads to label as related, calculated by kmer matched length / aligned length

Min edge depth

3

(-e) [int] Min read depth of a valid edge

Realignment

No

(-R) Enable realignment mode

Contained reads

No

(-A) Keep contained reads during alignment

Consensus Options

Execute

# Hifiasm parameters

**Tools**

hifiasm

Upload Data

Show Sections

Hifiasm haplotype-resolved de novo assembler for PacBio HiFi reads

**WORKFLOWS**

All workflows

**Bits for bloom filter**

37

A value of 0 disables the bloom filter (-f)

**Assembly options**

Specify

**Cleaning rounds**

4

(-B)

**Length of adapters to be removed**

0

(-Z)

**Minimum contig bubble size**

10000000

Pop contig graph bubbles smaller than this value (-m)

**Minimum unitig bubble size**

100000

Pop unitig graph bubbles smaller than this value (-p)

**Tip unitigs**

3

Keep only tip unitigs with a number of reads greater than or equal to this value (-n)

**Maximum overlap drop ratio**

0.8

This option is used with -r. Given a node N in the assembly graph, let max(N) be the length of the largest overlap of N. Hifiasm iteratively drops overlaps of N if their length/max(N) are below a threshold controlled by -x. Hifiasm applies -r rounds of short overlap removal with an increasing threshold between -x and -y (-y)

**Minimum overlap drop ratio**

0.2

This option is used with -r. Given a node N in the assembly graph, let max(N) be the length of the largest overlap of N. Hifiasm iteratively drops overlaps of N if their length/max(N) are below a threshold controlled by -y. Hifiasm applies -r rounds of short overlap removal with an increasing threshold between -x and -y (-y)

**Skip post join contigs step**

No

May improve N50 (-u)

**Ignore error corrected reads and overlaps**

No

Ignore error corrected reads and overlaps saved in prefix.\*.bin files. Apart from assembly graphs, hifiasm also outputs three binary files that save overlap information during assembly step. With these files, hifiasm can avoid the time-consuming all-to-all overlap calculation step, and do the assembly directly and quickly. This might be helpful when users want to get an optimized assembly by multiple rounds of



**Homozygous read coverage**

(--hom-cov)

**Options for purging duplicates**

Leave default

**Options for Hi-C-partition**

Leave default

**Advanced options**

Specify

**Hifiasm k-mer length**

51

(-k)

**Minimizer window size**

51

(-W)

**Drop k-mers**

5.0

K-mers that occur more than this value multiplied by the coverage will be discarded (-D)

**Maximum overlaps to consider**

100

The software selects the larger of this value and the k-mer count multiplied by coverage (-N)

**Correction rounds**

3

(-r)

**Minimum count threshold**

When analyzing the k-mer spectrum, ignore counts below this value (--min-hist-ct)

**Maximum k-mer occurrence**

20000

Employ k-mers occurring less than INT times to rescue repetitive overlaps (--max-kocc)

**Estimated haploid genome size**

Estimated haploid genome size used for inferring read coverage. If not provided, this parameter will be inferred by hifiasm. Common suffixes are required, for example, 100m or 3g (--hg-size)

**Output log file?**

# And

Processing will be performed using : <https://usegalaxy.fr/>

The data files are located at : shared libraries/EBAII A&A 2022/Assembly

The screenshot shows the Galaxy France web interface. At the top, there is a navigation bar with the 'Galaxy France' logo and various menu items like 'Workflow', 'Visualize', 'Données partagées', 'Aide', 'Utilisateur', and a 'Using 25%' indicator. Below the navigation bar is a search bar with the text 'Search' and a checkbox for 'exclude restricted'. A dropdown menu is open, showing options: 'Bibliothèque de données', 'Histories', 'Workflows', 'Visualisations', and 'Pages'. The main content area displays a table of datasets. The table has columns for 'Name', 'Description', and a globe icon. The datasets listed are: 'ProteoRE' (ProteoRE datasets), 'covid-19', 'GTN - Material' (Galaxy Training Network Material), 'workflow4metabolomics' (Workflow4Metabolomics referenced histori ...), 'Roscoff 2021' (Data for Assembly and Annotation trainin ...), and 'EBAII A&A 2022' (Ecole EBAII Assemblage & Annotation sept ...). A red arrow points to the 'EBAII A&A 2022' dataset. At the bottom of the table, there is a pagination bar showing '1' out of 6 total items.

Name	Description	
ProteoRE	ProteoRE datasets	
covid-19		
GTN - Material	Galaxy Training Network Material	Galaxy Training Network Material. See ht ... (more)
workflow4metabolomics	Workflow4Metabolomics referenced histori ... (more)	https://workflow4metabolomics.org/refere ... (more)
Roscoff 2021	Data for Assembly and Annotation trainin ... (more)	
EBAII A&A 2022	Ecole EBAII Assemblage & Annotation sept ... (more)	Edit

You can ask question using :

<https://semestriel.framapad.org/p/abgi5b9vdm-9vey?lang=fr>

Let's go!