UNIVERSITÄT BERN

Whole genome assembly using long read sequences

Heidi Tschanz-Lischer, Interfaculty Bioinformatics Unit (IBU)

17.02.2021

SIB Swiss Institute of Bioinformatics

De novo genome assembly

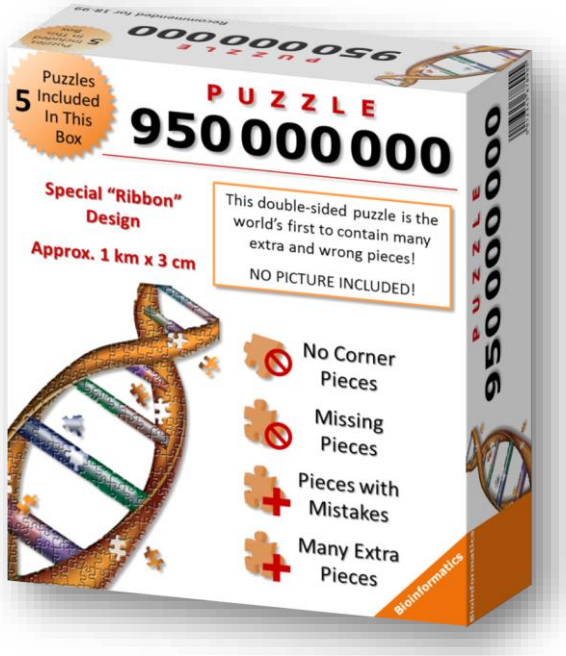
UNIVERSITÄT BERN

Why do we need whole genome sequences?

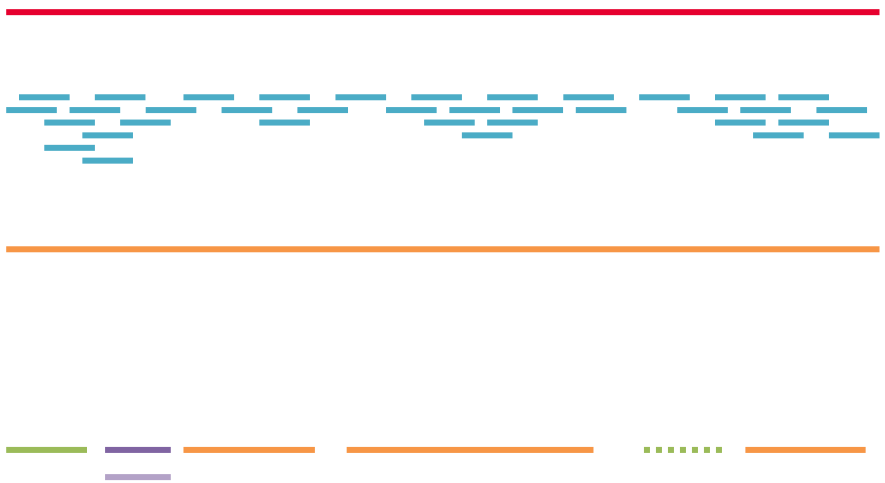
- Better understand variations within and between species
- Reduces the costs of future sequencing projects
  - Lower coverage required (e.g.: population genomic or genome wide association studies)
  - Simplify bioinformatics analyses

De novo assembly:

- Reconstructing the original DNA sequence from fragmented reads
- is like a big and complicated jigsaw puzzle
  - Millions of small pieces
  - Missing and pieces with mistakes (sequencing errors)
  - Polymorphisms (diploid)
  - Long repetitive parts



De novo genome assembly



Genome
(unknown)

reads

Ideal world:
Perfect assembly

Fragmented
(missing reads),
wrong size
(repeats,
split haplotypes)

What is needed for a good assembly?

- Low heterozygosity DNA
- High coverage
- High read lengths
- Good read quality

Current sequencing technologies do not have all

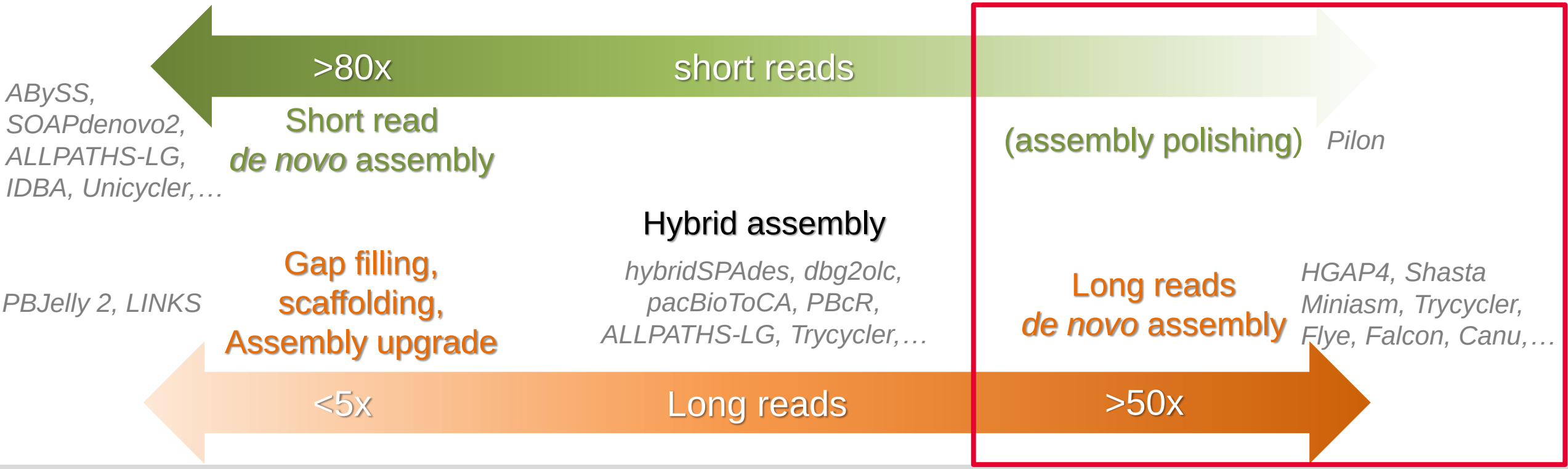
- Illumina: good quality reads, but short
- PacBio / Nanopore: very long reads, but lower quality
- PacBio HiFi: long reads and good quality, but expensive

→ Genome assembly is still a difficult problem and requires high computational resources

Choosing assembly strategy

The choice of algorithms depends on

- how much long reads (PacBio/Nanopore) can be obtained
- how much short read data are available

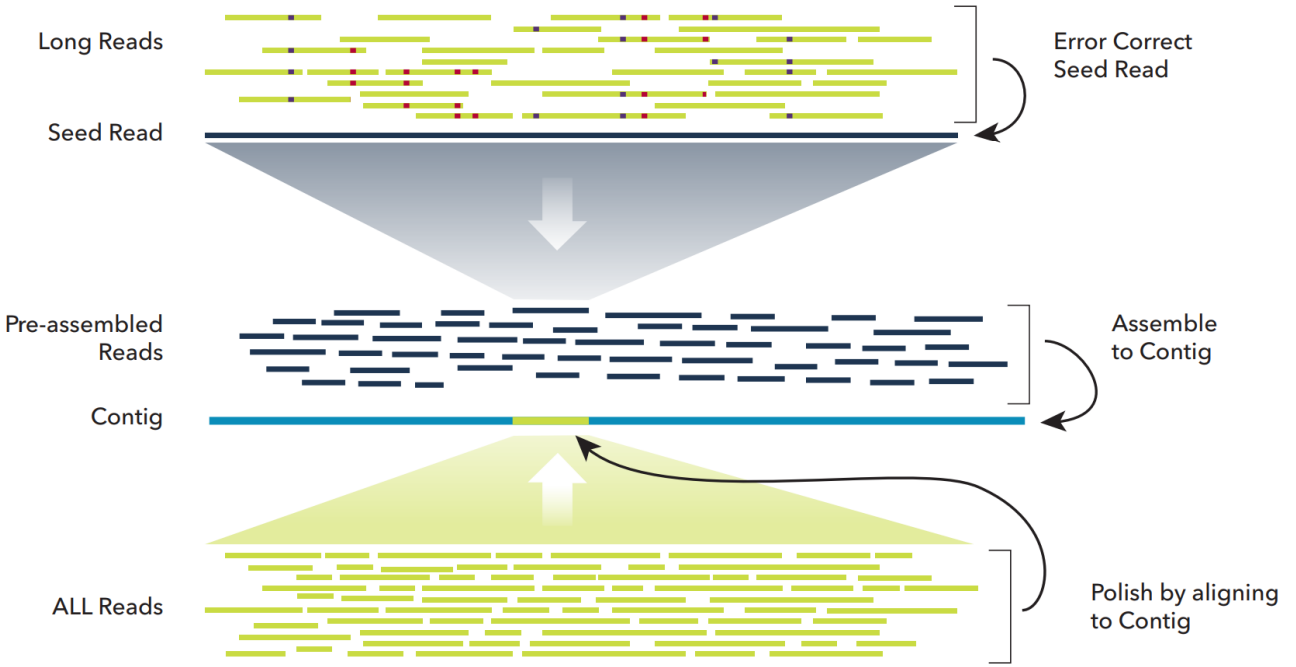


Long read de novo assemblers

UNIVERSITÄT BERN

HGAP: Hierarchical Genome Assembly Process
(https://github.com/PacificBiosciences/Bioinformatics-Training/wiki/HGAP)

- PacBio, included in the SMART Analysis software (GUI based)
- developed to allow the complete and accurate assembly of bacterial sized genomes (<100 Mb)
- 3 step process
  - Preassembly:
    generate long and highly accurate reads
  - Assembly:
    Overlap-layout-consensus (OLC)
  - Consensus polishing:
    reduce remaining Indels and
    SNP errors (Quiver)



Long read de novo assemblers

UNIVERSITÄT BERN

Miniasm (https://github.com/lh3/miniasm)

- Very fast OLC-based (overlap-layout-consensus) de novo assembler for noisy long reads
- Outputs only assembly graphs (GFA format) → no consensus calling

Trycycler (https://github.com/rrwick/Trycycler/wiki)

- hybrid or long-read-only (Flye, Miniasm, Raven) assemblies
- Takes multiple separate long-read assemblies and produces a consensus long-read assembly
- Not especially fast, but circularizes genomes without a separate tool (e.g. Circlator)

Both: → for bacterial genomes

→ PacBio or Nanopore reads

→ easy and straight forward to use

Long read de novo assemblers

Shasta (https://github.com/chanzuckerberg/shasta)

- Nanopore reads
- Default parameters optimized for coverage ~60x
- RAM requirements: around 4-6 bytes per input base
- Accuracy is comparable or better than alternative assemblers
- Very fast and simple to use
- 60x human genome: 5 hours (128 virtual CPUs) and 2 TB RAM

Raven (https://github.com/lbcb-sci/raven)

- PacBio or Nanopore
- Fast and simple to use
- 44x Nanopore human genome: 500 CPU hours, 380 GB RAM

Long read de novo assemblers

UNIVERSITÄT BERN

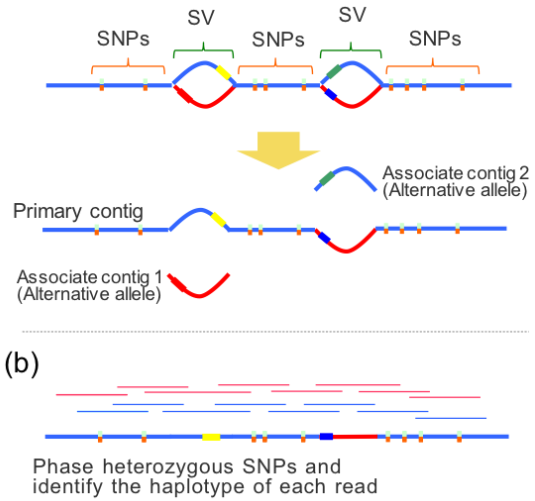
Wtdbg2 (redbean) (https://github.com/ruanjue/wtdbg2)

- PacBio or Nanopore
- Assembles reads without error correction and builds consensus sequences
- Based on fuzzy Bruijn graph method
- Fast and easy to use assembler
- from small bacterial projects to large mammalian-scale assemblies
    - E. coli (4.6 Mb) 50x PacBio: 10m CPU time, 1 Gb RAM
    - Human (3 Gb) 28x PacBio (HiFi): 290h CPU time, 113 Gb RAM
    - Human (3 Gb) 35x Nanopore: 1025h CPU time, 215 Gb RAM

Long read de novo assemblers

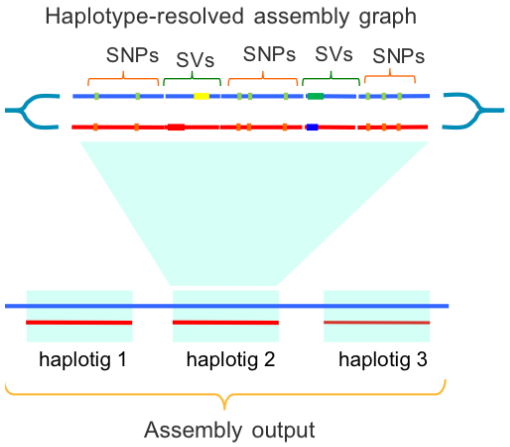
FALCON / pb-assembly (https://pb-falcon.readthedocs.io/en/latest/)

- PacBio Assembly tool Suite
- diploid-aware assembler → follows HGAP
- optimized for large genome assembly
- >30-50x per haplotype (highly heterozygous diploid → require the double)
- extensive configuration file required
    - not easy to understand parameters
    - A few example files → can be used as a basis for modification



FALCON-Unzip:

- phase the genome and perform phased-polishing with Arrow
- partially-phased primary contigs and fully-phased haplotigs (haplotypes)



Long read de novo assemblers

Canu (fork of Celera Assembler; https://canu.readthedocs.io/en/latest/index.html)

- PacBio or Nanopore
- 3 phases: correction → trimming (get high-quality sequences) → assembly
- follows the hierarchical genome assembly process (HGAP)
- >30-60x
- automatically takes full advantage of grid systems (cluster) → submitting itself for execution
- consensus sequences:
    - >99.99% identity for PacBio HiFi data
    - >99% identity for PacBio and Nanopore (accuracy varies depending on pore and basecaller version)
- Easy and straight forward to use
    - Good manual with recommendations for parameter values (PacBio, Nanopore, low coverage data)

Long read de novo assemblers

UNIVERSITÄT BERN

Flye (https://github.com/fenderglass/Flye)

- PacBio or Nanopore
- from small bacterial projects to large mammalian-scale assemblies
  - E. coli (4.6 Mb) 50x PacBio: 2h CPU time, 2 Gb RAM
  - Human (2.9 Gb) 30x PacBio (HiFi): 780h CPU time, 300 Gb RAM
  - Human (2.9 Gb) 35x Nanopore: 3100h CPU time, 141 Gb RAM
- complete pipeline: raw reads → polished contigs
- Include special mode for metagenome assembly
- Very easy and straight forward to use

Post-assembly correction

→ improves quality and removes errors

- Nanopore → Nanopolish:
  - calculates an improved consensus sequence
  - nanopolish call-methylation: predict methylated genomic bases
  - nanopolish variants: detect SNPs and indels

  → Medaka:
  - Faster than Nanopolish (can use GPU)
  - Create consensus sequence and variant calls → based on a neural network approach

- PacBio → Arrow (former Quiver):
  - Get improved consensus → based on a hidden Markov model approach
  - get variant calls

- Illumina → Pilon:
  - Automatically improve draft assemblies (SNPs, small/large indels, gap filling, local misassemblies)
  - Find variations, including large event detection

Genome annotation

Bacteria

Prokka

- rapid prokaryotic genome annotation
- quickly annotate bacterial, archaeal and viral genomes
- outputs standard-compliant files

RAST

- Rapid Annotation using Subsystem Technology
- fully-automated service for annotating complete or nearly complete bacterial and archaeal genomes
- Webservice (http://rast.theseed.org/FIG/rast.cgi)

glimmer3

- Gene Locator and Interpolated Markov ModelER
- a system for finding genes in microbial DNA

Genome annotation

Eukaryotes

- RepeatMasker: screens genome for interspersed repeats and low complexity DNA sequences
- MAKER:
  - Genome annotation pipeline → allow smaller projects to independently annotate their genomes
  - identifies repeats
  - aligns ESTs and proteins to a genome
  - produces ab-initio gene predictions
- PASA (Program to Assemble Spliced Alignments):
  - exploits spliced alignments of transcripts to automatically model gene structures and splice variations
- Augustus:
  - find genes and their structures
  - can be used as an ab initio program → bases its prediction purely on the sequence.
  - also incorporate hints from extrinsic sources (e.g.: EST, MS/MS, protein alignments, …)
  - fully automatic annotation pipeline available

Example

Cichlid genome *(Pundamilia nyererei)*

PunNye1.0 (Broad Institute):

- Nb scaffolds: 7,236
- N50: 2.5 Mb
- Total length: 830.1 Mb
- Total length: (without N) 698.8 Mb

→ 126x Illumina read

→ ALLPATHS-LG

PunNye2.0 (Feulner *et al.*):

- Nb scaffolds: 6,876
- N50: 29.8 Mb
- Total length: 856.2 Mb
- Total length: (without N) 698.8 Mb

→ Linkage map - 1,597 SNP markers

→ ALLMAPS

Raw data

PacBio (Sequel)

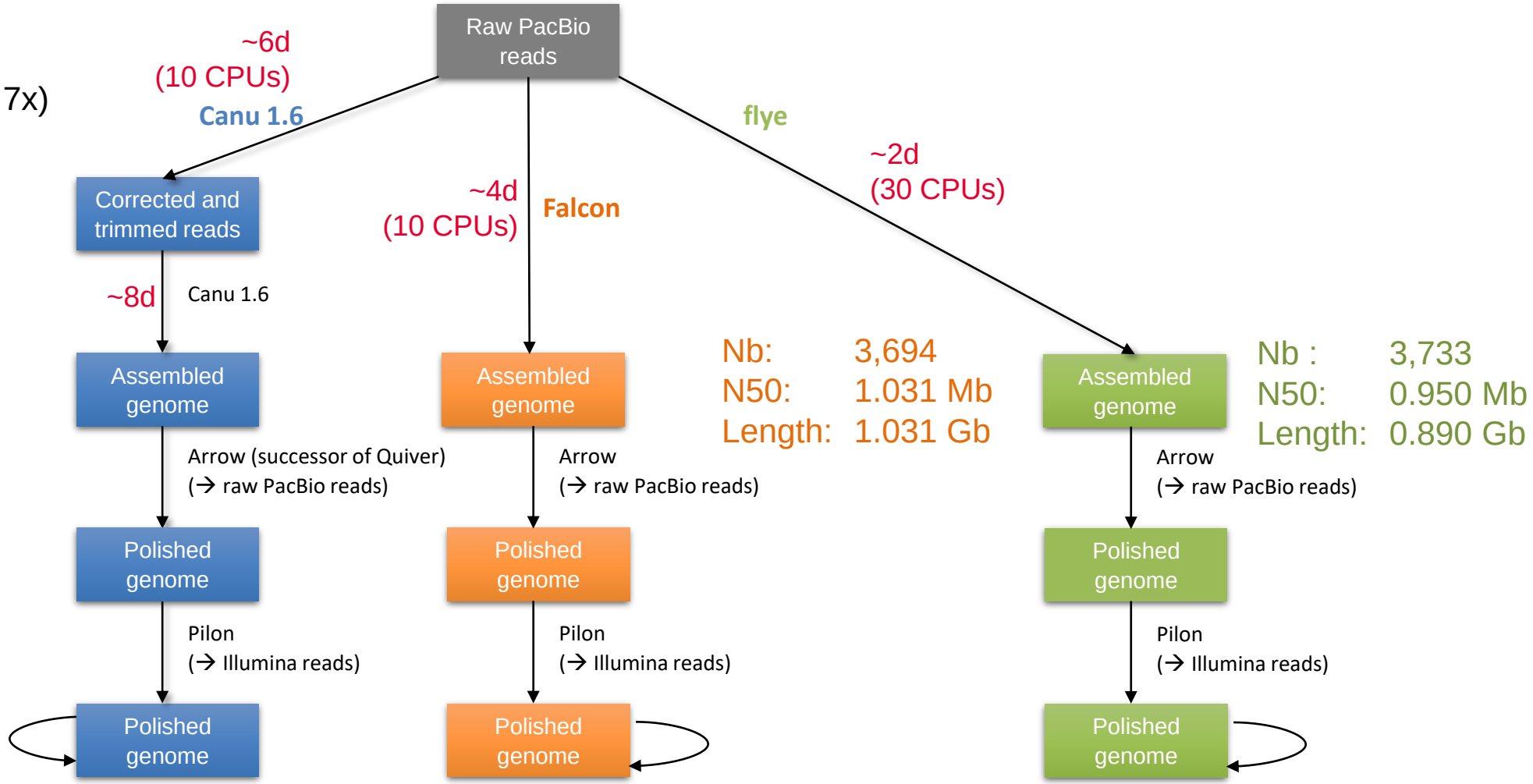
- Nb reads: 4,020,155
- Min length: 50 bp
- Max length: 143,514 bp
- Mean length: 10,538 bp
- Total length: 42.35 Gb → estimated coverage 42.7x

Illumina reads

- 4 closely related samples (380bp insertion):
  - Nb reads: 520,955,224
  - Total length: 78.14 Gb → estimated coverage ~78.7x (each sample 15-20x)
- SRA samples (used in original PunNye1.0 assembly):
  - 3 kb libraries: 709,783,284 (72.2x coverage)
  - 6-14 kb libraries: 721,087,418 (51.2x coverage)
  - 40 kb FOSILLs4: 36,341,216 (3.7x coverage)

Pipeline – genome assembly

Nb reads: 4,020,155
Mean length: 10,538 bp
Total length: 42.36 Gb (~42.7x)

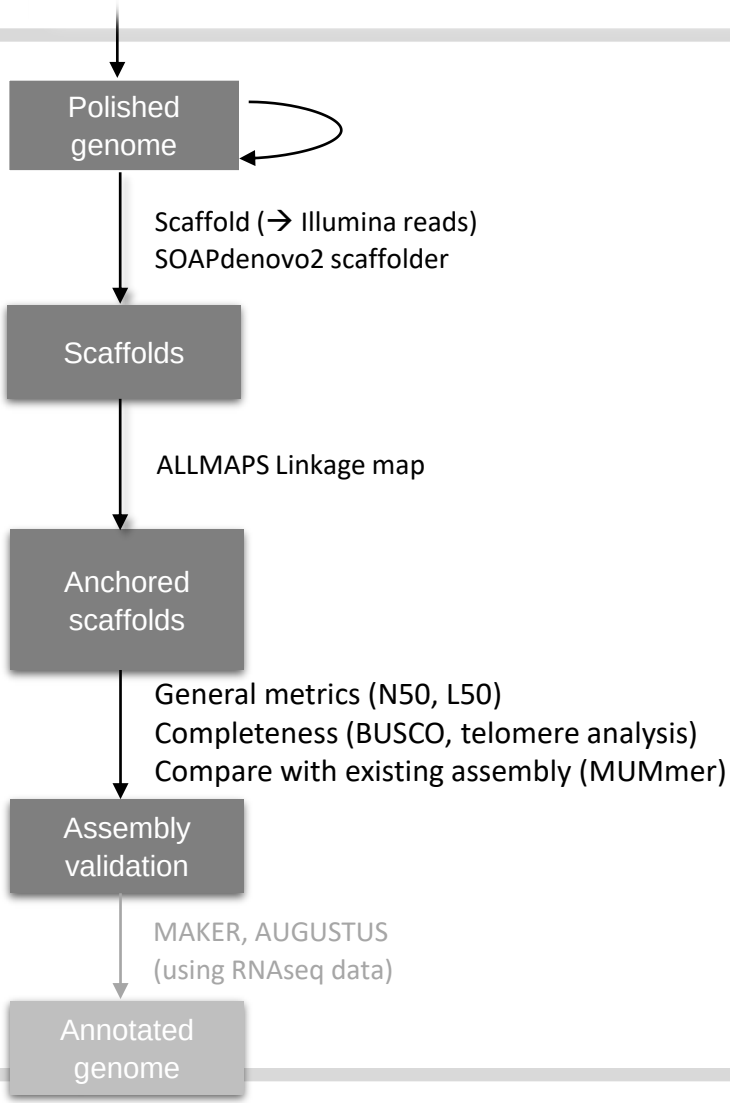


Nb contigs: 6,732
N50: 0.920 Mb
Total length: 1.111 Gb

Nb: 3,694
N50: 1.031 Mb
Length: 1.031 Gb

Nb : 3,733
N50: 0.950 Mb
Length: 0.890 Gb

Pipeline – genome assembly



Canu 1.6

Nb scaffolds: 5,753
N50: 27.607 Mb
Total length: 1.130 Gb
(without N: 1.083 Gb)

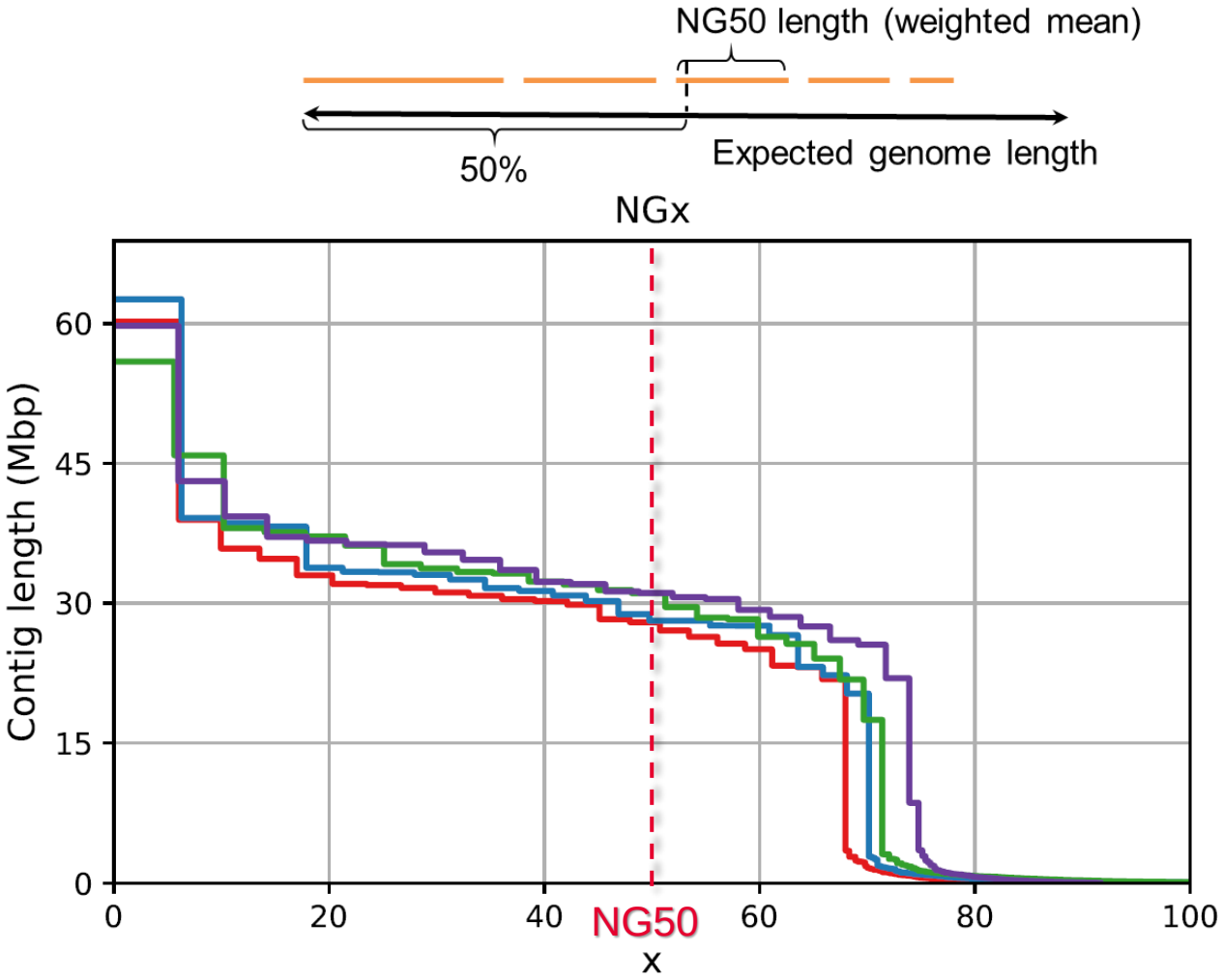
Falcon

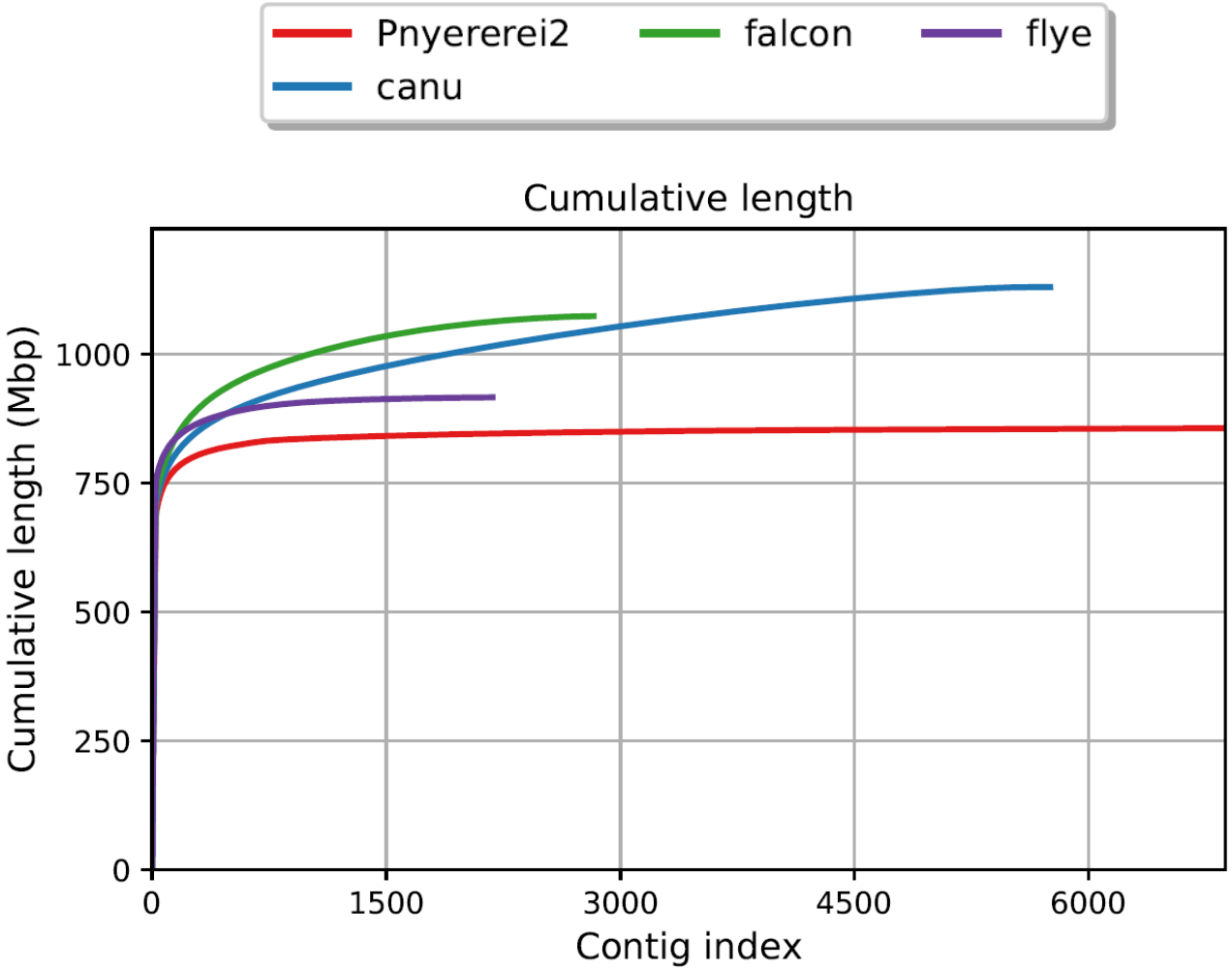
Nb scaffolds: 2,832
N50: 29.594 Mb
Total length: 1.074 Gb
(without N: 1.033 Gb)

Flye

Nb scaffolds: 2,201
N50: 31,265 Mb
Total length: 0.916 Gb
(without N: 0.891 Gb)

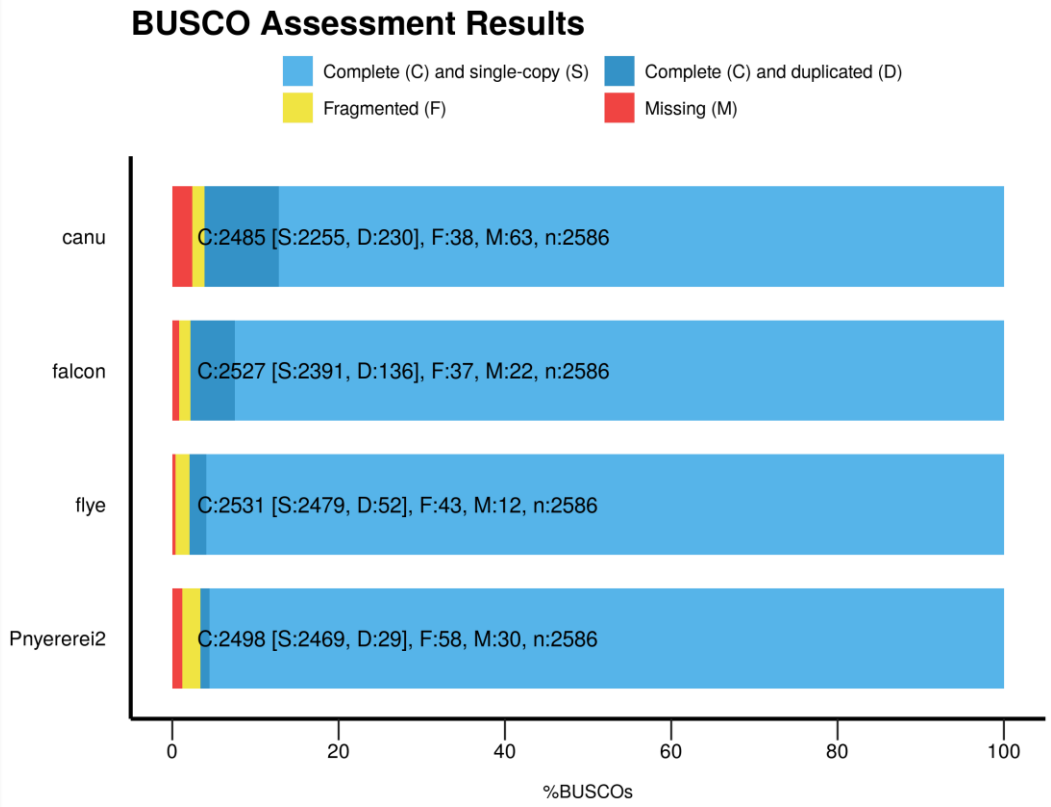
Assembly comparison





Assembly comparison

	Stats	Pnyererei2	canu	falcon	flye
QUAST					
	nb scaffolds	6’876	5’753	2’832	2’201
	N50	29’830’996	27’606’787	29’593’783	31’265’002
	NG50	27’967’145	28’136’703	31’084’370	31’116’785
	max length	60’199’168	62’610’257	55’908’494	59’759’838
	total length	856’242’559	1’130’373’166	1’073’822’959	916’080’688
	total length without N	698’778’000	1’083’432’443	1’032’560’317	890’889’752
	N's per 100kb	18’390	4’153	3’843	2’750
BUSCO					
	complete	2’498	2’484	2’528	2’532
	complete single copy	2’469	2’256	2’394	2’481
	complete duplicated	29	228	134	51
	fragmented	58	39	36	42
	missing	30	63	22	12



Conclusion

- (high quality) long read sequences are important for high quality draft genomes
- >50x coverage is required for long reads only assemblies
  → still expensive for large genomes, but prices will come down even more in near future
- No single best assembly strategy/program, depends on
  - Input data (quality, coverage)
  - Species (heterozygosity, complexity)
- Assembly evaluations are not straight forward
  - Longer assembles (higher N50/NG50) are not always the best assemblies
  - Always use a combination of metrics
  - Only a few tools work without a known reference
    (e.g.: BUSCO (Waterhouse et al. 2017), QUAST (Gurevich et al. 2013),
    ALE (Clark et al. 2013), REAPR (Hunt et al. 2013))