b
UNIVERSITÄT
BERN

Whole genome assembly using long read sequences

Heidi Tschanz-Lischer, Interfaculty Bioinformatics Unit (IBU)

11.06.2019

Swiss Institute of Bioinformatics

Why do we need whole genome sequences?

- Sequencing the genome is an important step towards understanding it
- Better understand variations within and between species
- Makes it easier to study
  - Cause of diseases
  - Morphological variation
  - Environmental adaptation
  - Genomic basis for evolutionary speciation
  - Gene expression divergence
  - Epigenetic modifications
  - …

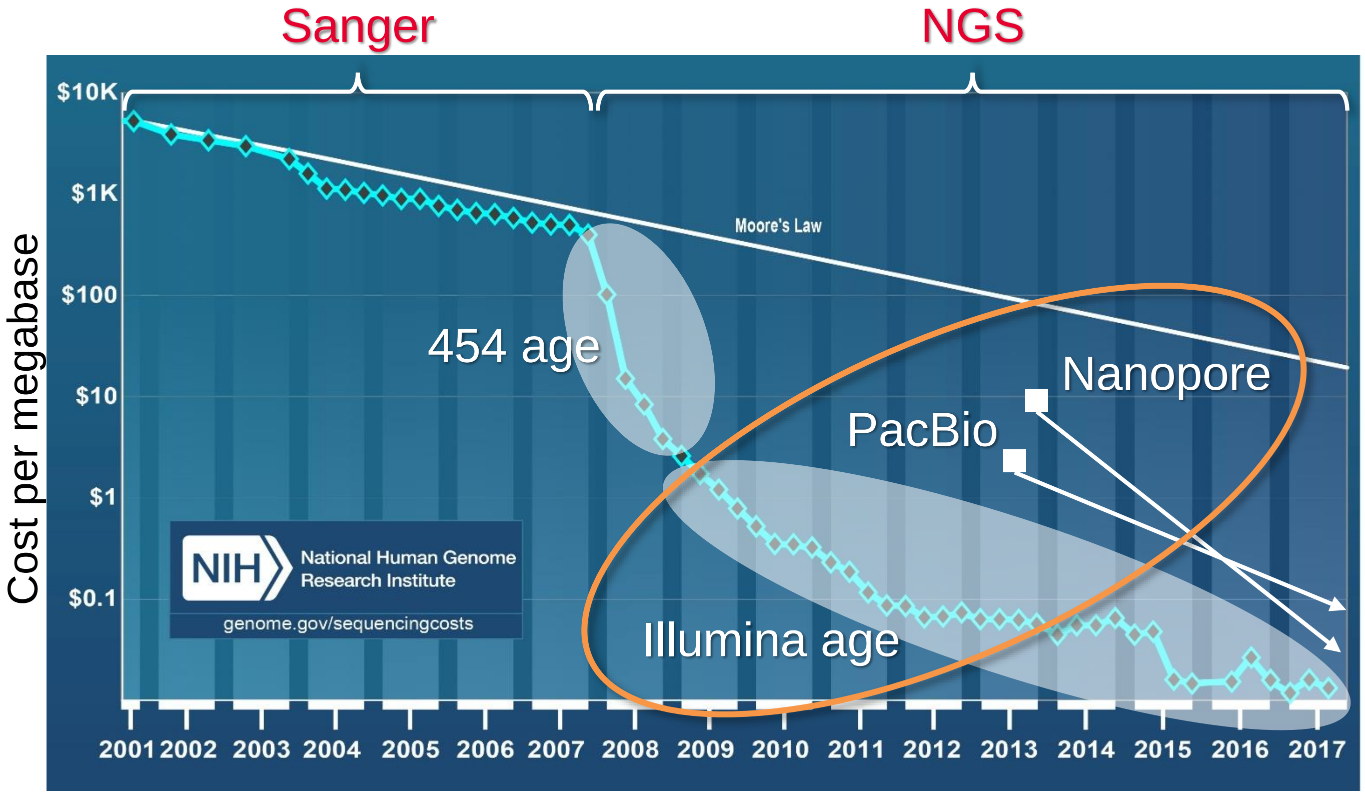
- Reduces the costs of future sequencing projects
  - Lower coverage required → population genomic or genome wide association studies
  - Simplify bioinformatics analyses

Genome sequencing costs dropped

Next-Generation Sequencing

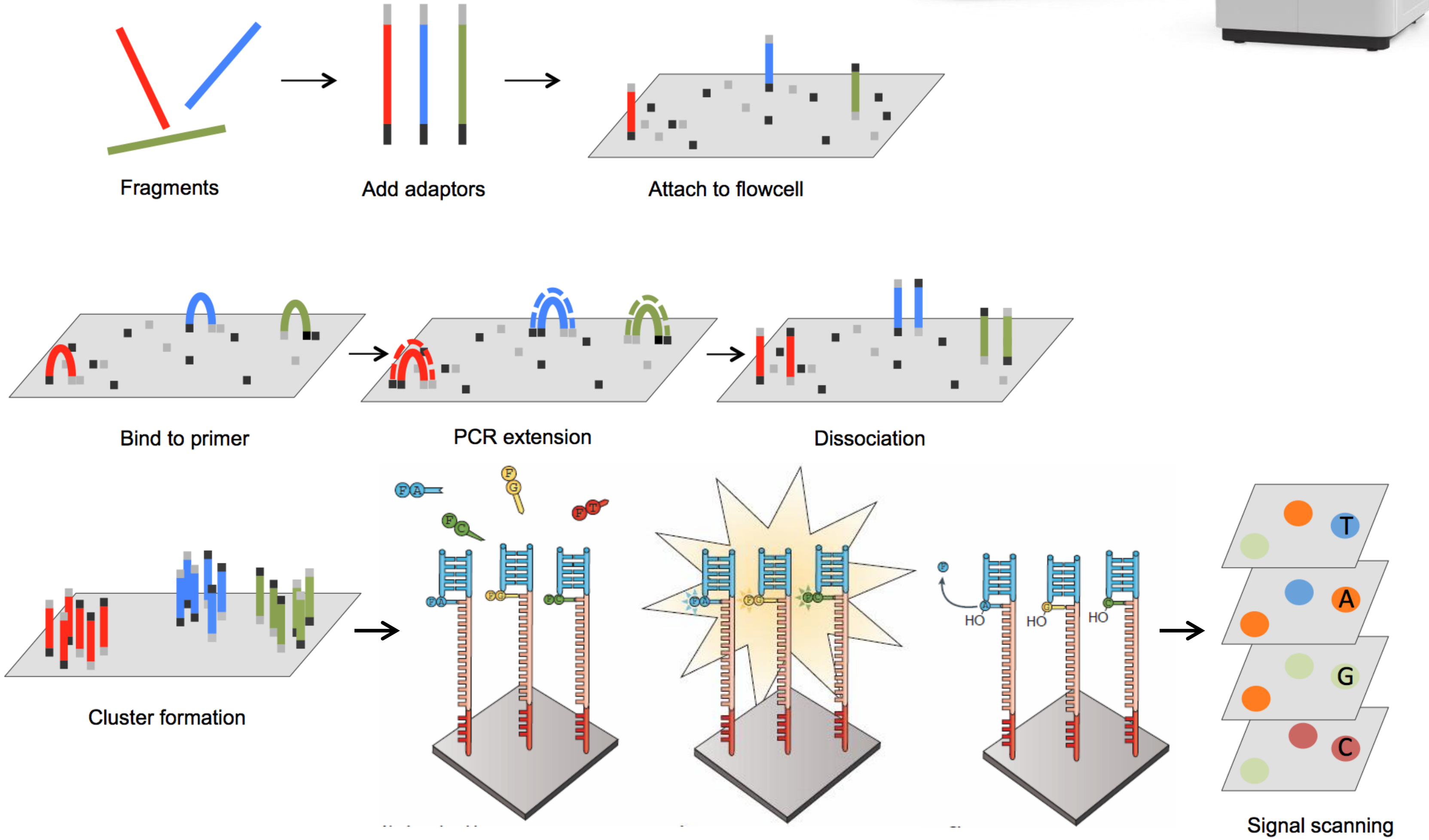
- Highly parallelized sequencing
- Reduces cost and time

→ main challenge in genome sequencing has shifted from data generation to reconstruction of genomes



Illumina

UNIVERSITÄT BERN


Fragments

Add adaptors

Attach to flowcell

Bind to primer

PCR extension

Dissociation

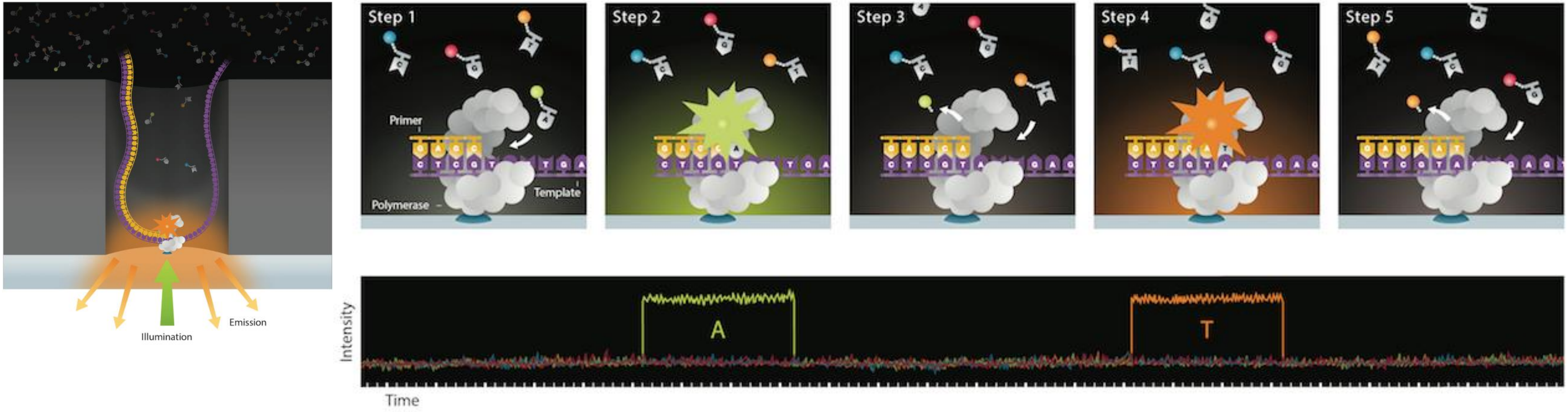
Cluster formation

Signal scanning

Sequencing by synthesis (termination)

- 50-300 bp length (PE)
- Error rate: 0.1%
- Less susceptible to homopolymer errors
- Under-representation in AT-rich and GC-rich regions
- Tendency towards substitution errors
- High troughput: 15-3000 Gb

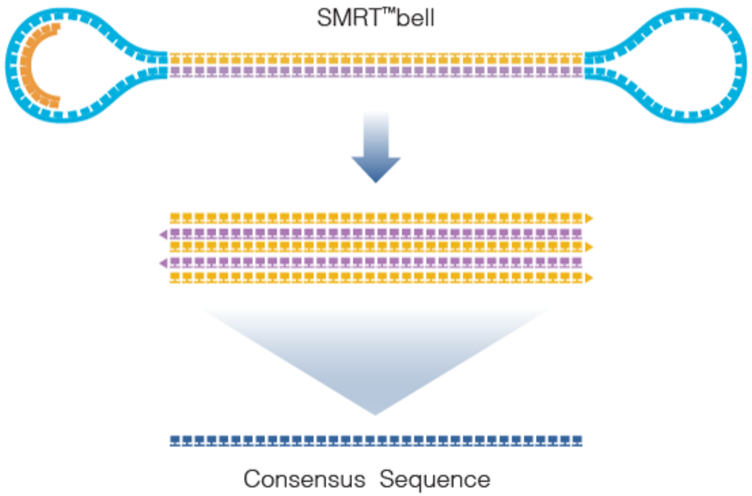
Single-molecule real-time (SMRT), PacBio



- Slowed down DNA polymerase
- Measure colored light emission

Single-molecule real-time (SMRT), PacBio

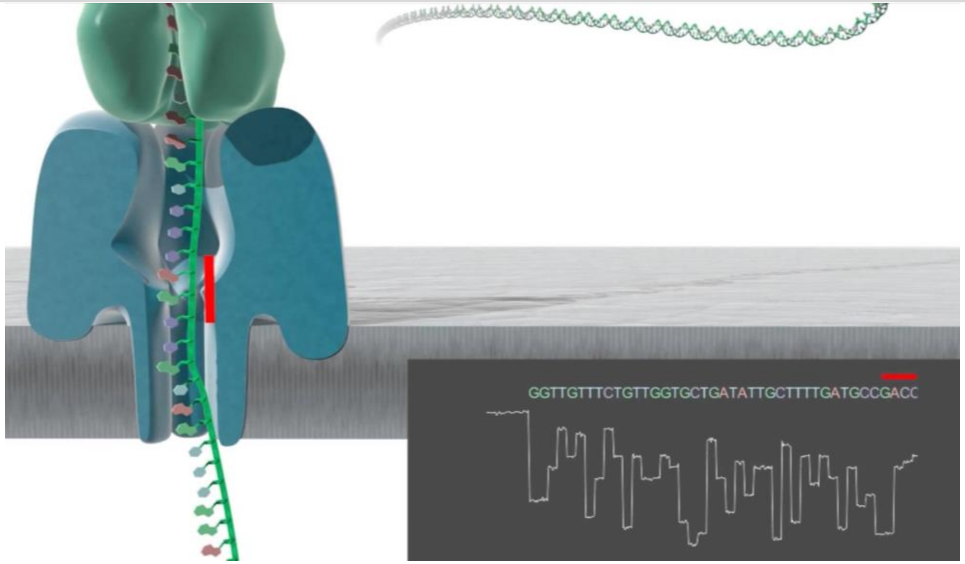
- Average read length (Sequel II): ~50 kb (up to reads of 175 kb length)
- Error rate: 10-15%
- Bias towards indel errors
- Errors are supposed to be random
  → high coverage can overcome high error rate
- Higher cost
- Unique circular template
  - → allows each template to be sequenced multiple times
  - → create a circular consensus sequence (HiFi reads)
  - → reduces error rate to <1%
    (10 passes: ~99.8% accuracy)

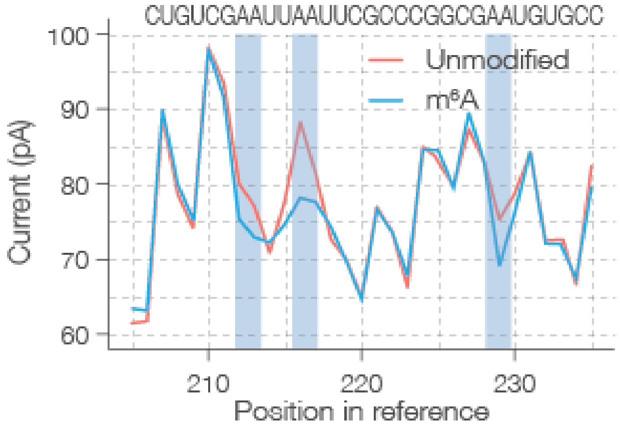


Oxford Nanopore sequencing

Nano-scale pores

- Drag DNA/RNA through the pore
- Electrical current flows through the hole
  → disruption is measured if DNA
  pass through
- Shifts in voltage
  → characteristic for particular sequence
  in pore (k-mer)
  → Modifications to the primary DNA or RNA
  (e.g. cytosine methylation) can be
  detected directly
- Problem:
  Multiple bases influence the current
  passing through the pore



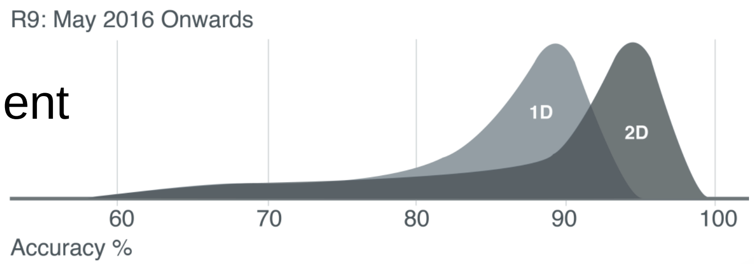


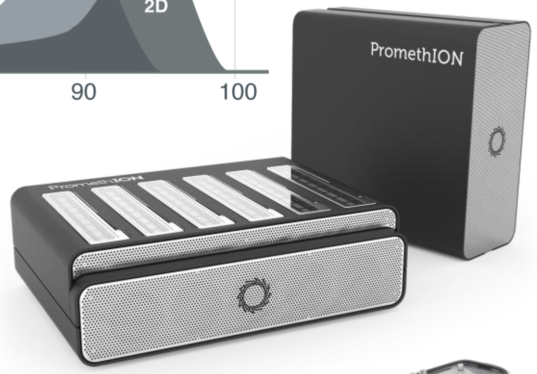
Oxford Nanopore sequencing

UNIVERSITÄT BERN

MinION: a portable sequencing device

- 512 channels
- Small (USB based device), runs off a personal computer
- real-time sequencing data (~150bp per seconds; 500bps fast mode)
  → no fixed run time: *Run until...* sufficient data (15-30 Gb)
- Long reads >6 kb (up to 2mb) → DNA molecule length dependent
- Error rate: 5-12% (dominated by indels, homopolymers)


R9: May 2016 Onwards



PromethION: small benchtop system

- contains docking for 48 flow cells, each with 3000 nanopores
  → Total: 144,000 nanopores (< 4.8-8.6 Tb)

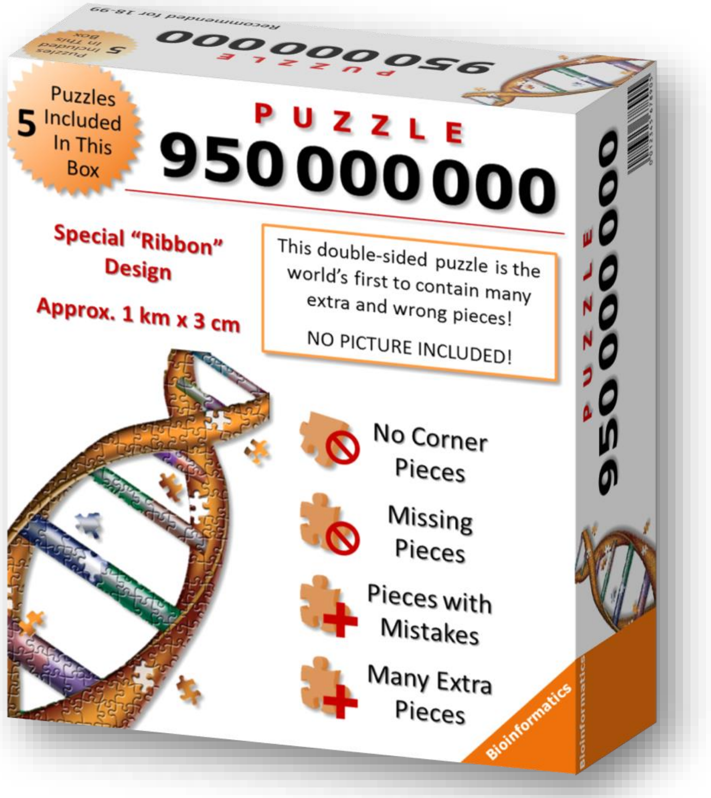
VolTRAX v2: programmable, portable device for automatic sample and library preparation

De novo genome assembly

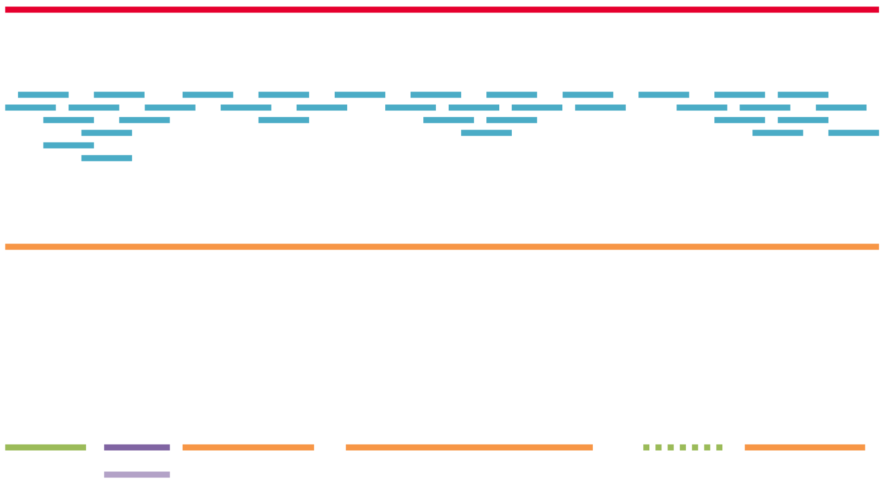
De novo assembly:

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

→ Different algorithms and assemblers developed
(e.g. ABySS, SOAPdenovo2, ALLPATHS-LG, Falcon, Canu, …)



De novo 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

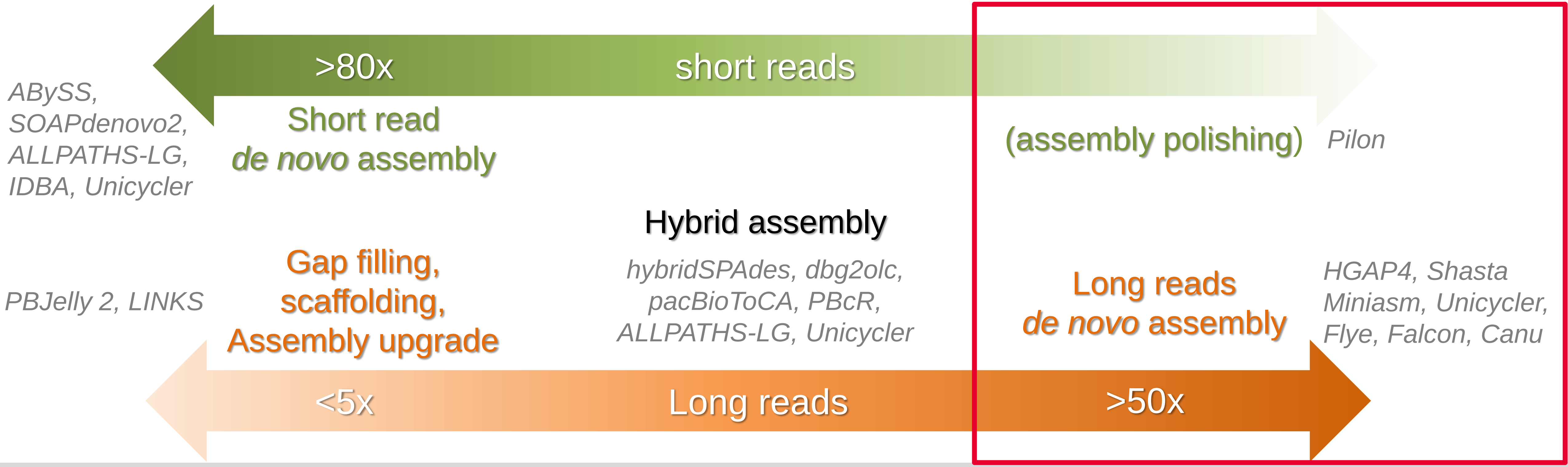
→ Genome assembly is **still a difficult problem** and requires high computational resources

Choosing assembly strategy

UNIVERSITÄT BERN

The choice of algorithms depends on

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



>80x short reads

ABySS, SOAPdenovo2, ALLPATHS-LG, IDBA, Unicycler

Short read *de novo* assembly

(assembly polishing) Pilon

Hybrid assembly

hybridSPAdes, dbg2olc, pacBioToCA, PBcR, ALLPATHS-LG, Unicycler

PBJelly 2, LINKS

Gap filling, scaffolding, Assembly upgrade

Long reads *de novo* assembly

HGAP4, Shasta Miniasm, Unicycler, Flye, Falcon, Canu

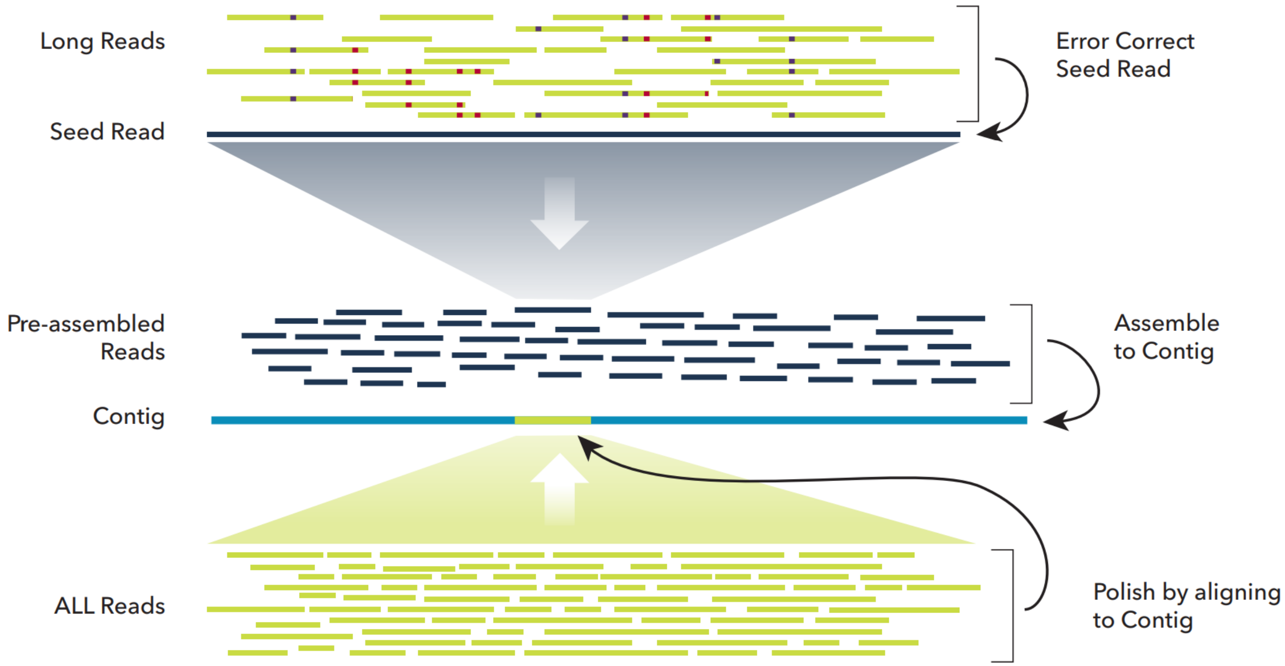
<5x Long reads >50x

Long read de novo assemblers

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

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

- Nanopore reads
- Very fast and simple to use
- Default parameters optimized for coverage ~60x
- Output assembly in FASTA and GFA 1.0 (assembly graphs)
- RAM requirements: around 5-8 bytes per input base
- Early indications are that Shasta accuracy is at least comparable to alternative assemblers
- Designed for *de novo* assembly of human genomes
    - RAM: 1 TB (60x coverage)
    - Runtime: 6 hours (128 virtual CPUs)

Long read de novo assemblers

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

Unicycler (https://github.com/rrwick/Unicycler)

- Illumina-only (SPAdes optimizer), hybrid or long-read-only (miniasm + Racon) assemblies
- Can cope with very repetitive genomes
- 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

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:	900h CPU time, 300 Gb RAM
Human (2.9 Gb) 35x Nanopore:	5000h CPU time, 600 Gb RAM

- complete pipeline: raw reads → polished contigs
- Include special mode for metagenome assembly
- Easy and straight forward to use

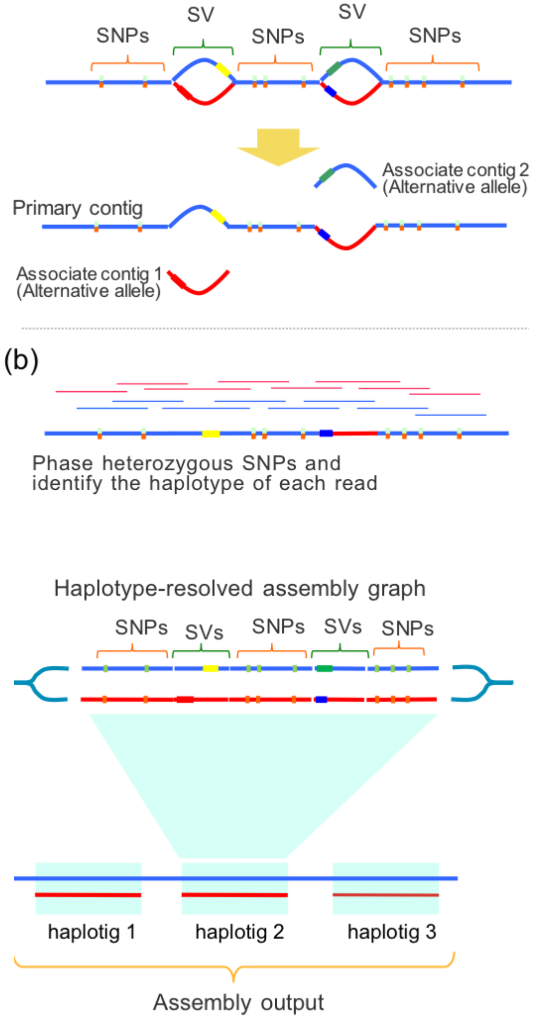
Long read de novo assemblers

FALCON / pb-assembly (https://github.com/PacificBiosciences/pb-assembly)

- 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% identity for PacBio
  - >98% identity for 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)

Post-assembly correction

→ improves quality and removes errors

Polish assembly with long reads

- Nanopore → Nanopolish:
  - calculates an improved consensus sequence
  - nanopolish call-methylation: predict methylated genomic bases
  - nanopolish variants: detect SNPs and indels
- PacBio → Arrow (former Quiver):
  - Get improved consensus → based on a hidden Markov model approach
  - get variant calls

Polish assembly with Illumina reads:

- 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)

Genome annotation

Eukaryotes

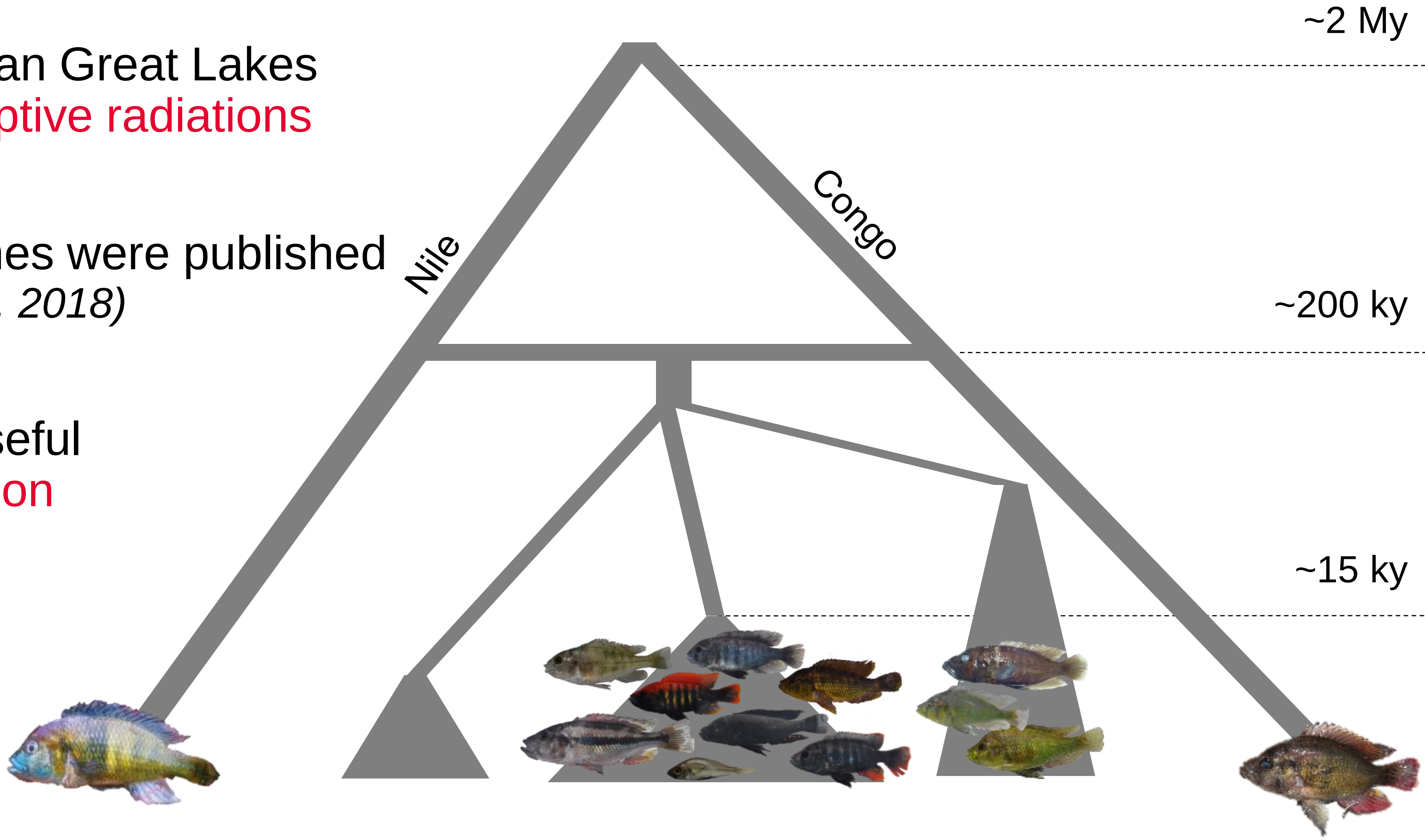
- RepeatMasker: screens genome for interspersed repeats and low complexity DNA sequences
- MAKER:
  - Genome annotation pipeline → allow smaller projects to independently annotate their genomes
  - identifies repeats
  - aligns ESTs and proteins to a genome
  - produces ab-initio gene predictions
  - especially useful for projects with minimal bioinformatics expertise and computer resources
- 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, …)

Example

Cichlid genome assembly

- Cichlid lineage of the East African Great Lakes is famous for forming large adaptive radiations in exceptionally short time
- Recently, several cichlid genomes were published *(Brawand et al. 2014, Feulner et al. 2018)*

→ Improved genome will be useful for future studies of adaptation and speciation of Lake Victoria cichlids



Cichlid genome resources

Pundamilia nyererei

UNIVERSITÄT BERN

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)

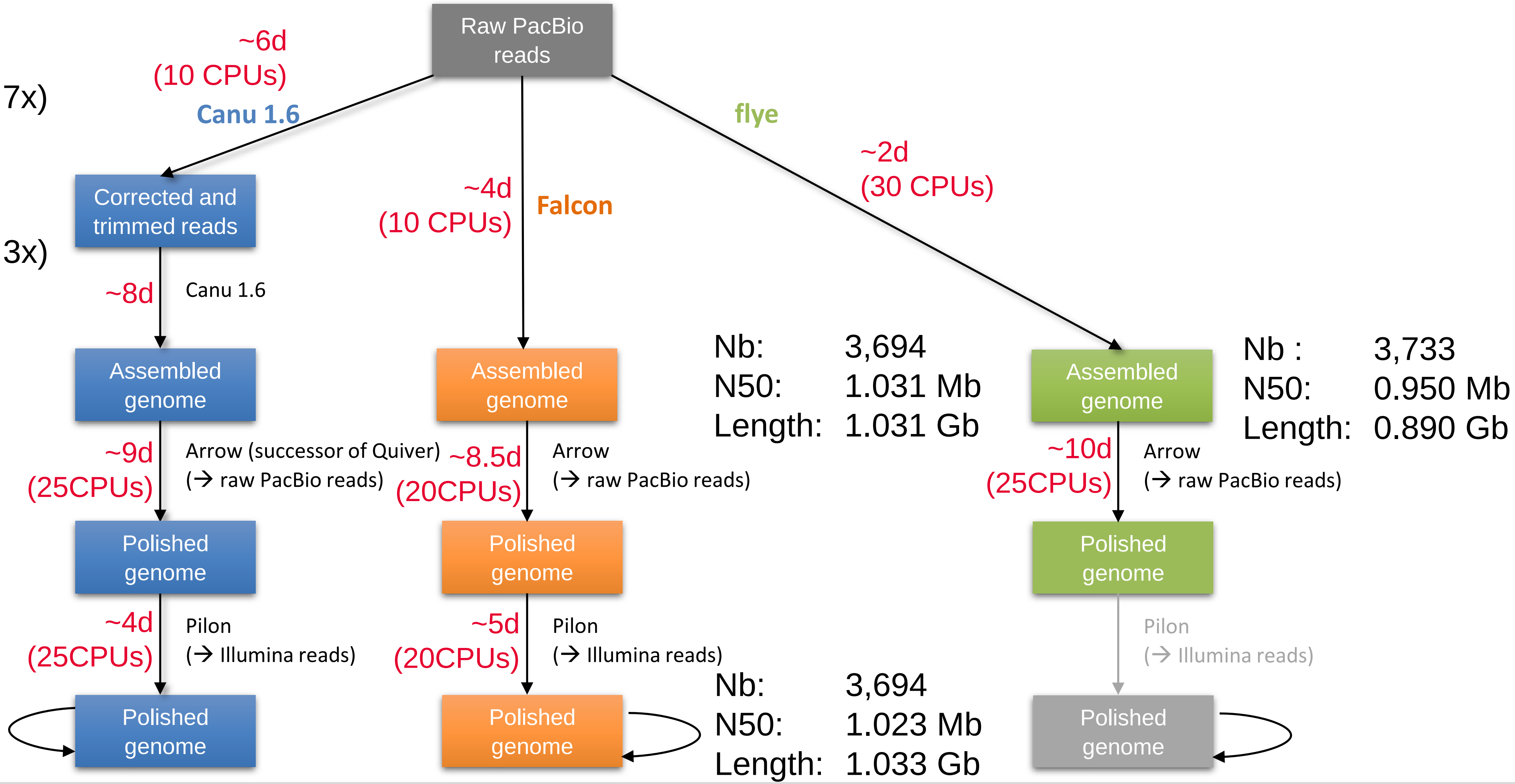
Pipeline – genome assembly

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

Nb reads: 2,228,798
Mean length: 13,505 bp
Total length: 30.10 Gb (~30.3x)

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

Nb contigs: 6,584
N50: 0.908 Mb
Total length: 1.083 Gb



Raw PacBio reads

Canu 1.6 ~6d (10 CPUs) → Corrected and trimmed reads → ~8d Canu 1.6 → Assembled genome → ~9d (25CPUs) Arrow (successor of Quiver) (→ raw PacBio reads) → Polished genome → ~4d (25CPUs) Pilon (→ Illumina reads) → Polished genome

Falcon ~4d (10 CPUs) → Assembled genome → ~8.5d (20CPUs) Arrow (→ raw PacBio reads) → Polished genome → ~5d (20CPUs) Pilon (→ Illumina reads) → Polished genome

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

Nb: 3,694
N50: 1.023 Mb
Length: 1.033 Gb

flye ~2d (30 CPUs) → Assembled genome → ~10d (25CPUs) Arrow (→ raw PacBio reads) → Polished genome → Pilon (→ Illumina reads) → Polished genome

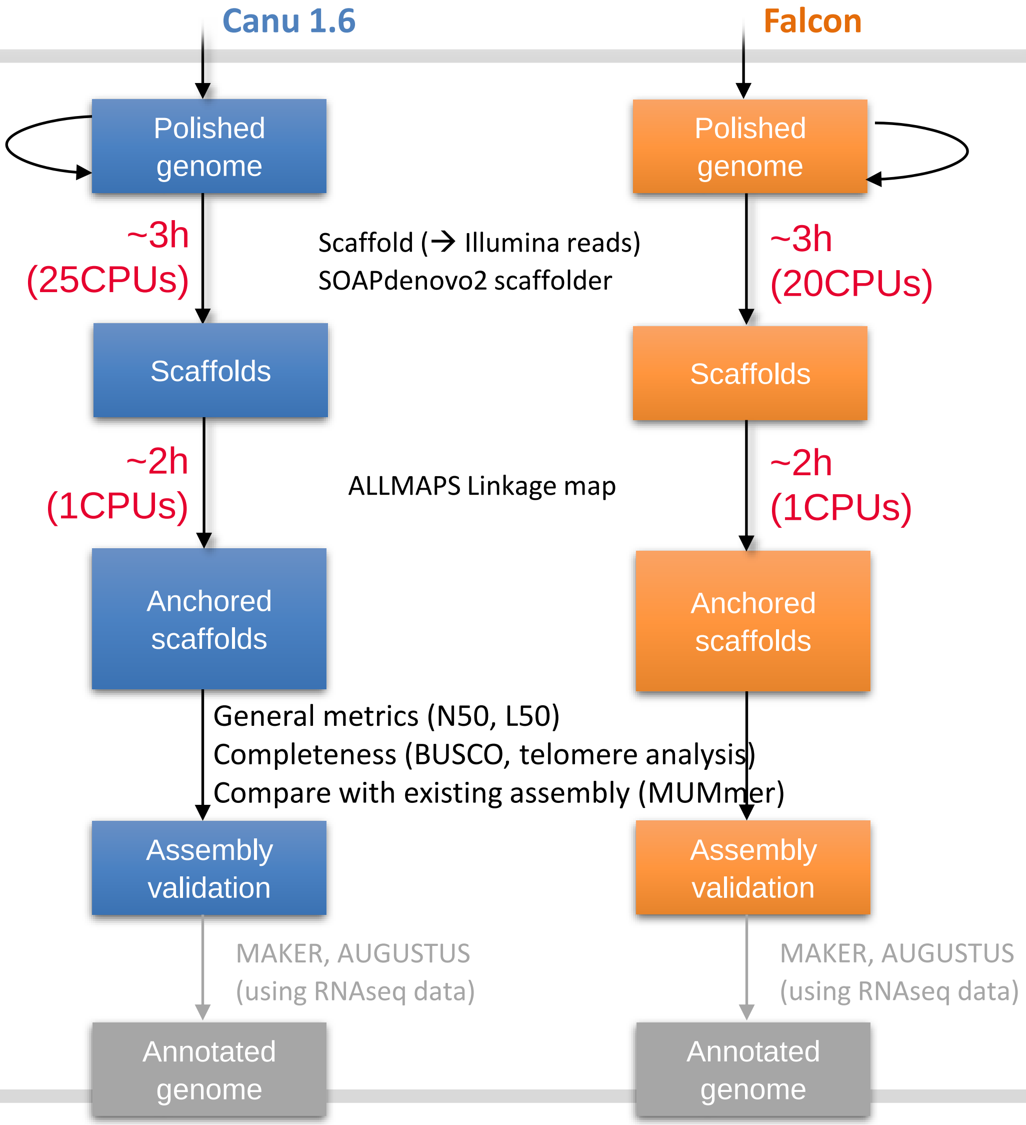
Nb : 3,733
N50: 0.950 Mb
Length: 0.890 Gb

Pipeline – genome assembly

Nb contigs:	6,584
N50:	0.908 Mb
Total length:	1.083 Gb

Nb scaffolds:	6,250
N50:	1.212 Mb
Total length:	1.085 Gb

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

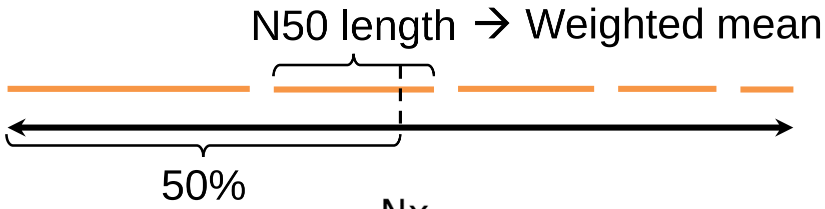


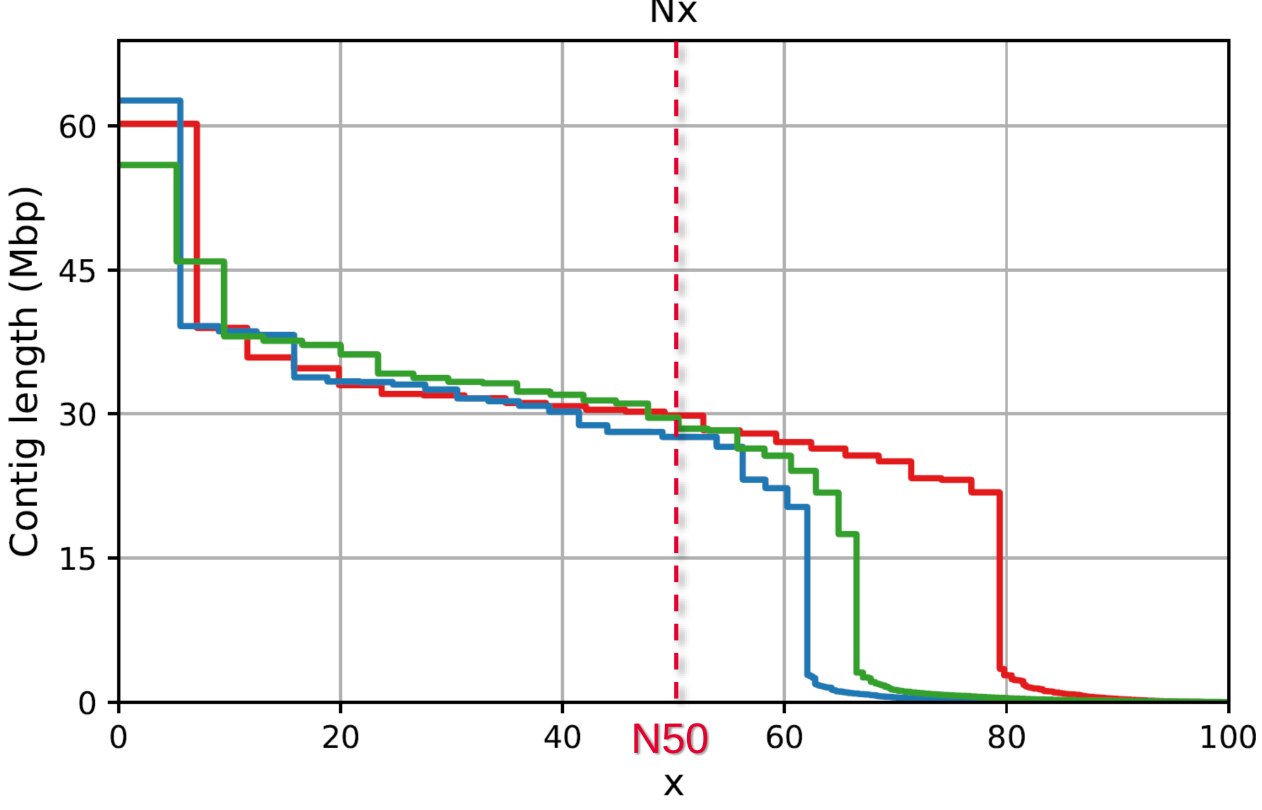
Nb contigs:	3,694
N50:	1.023 Mb
Total length:	1.033 Gb

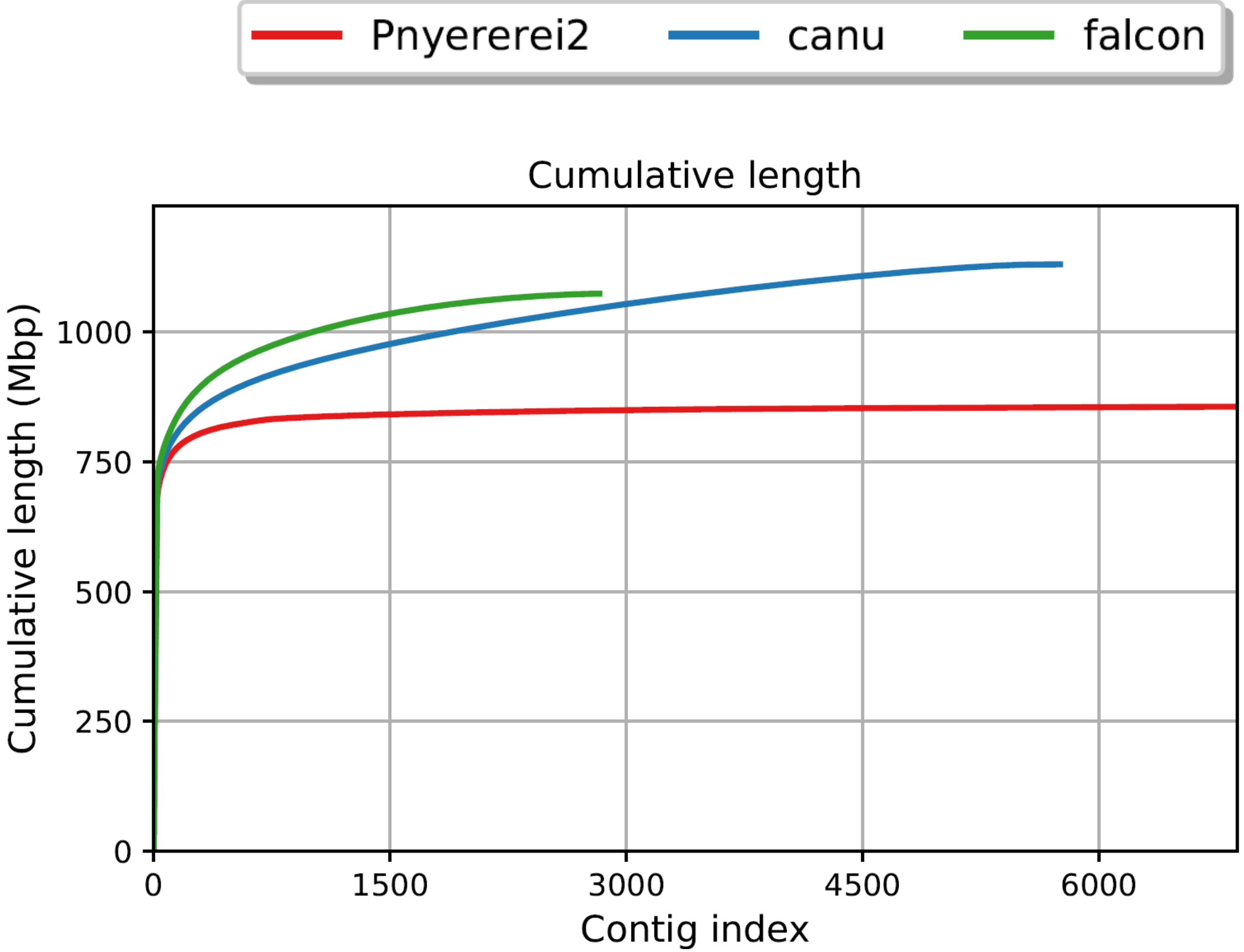
Nb scaffolds:	3,335
N50:	1.324 Mb
Total length:	1.036 Gb

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

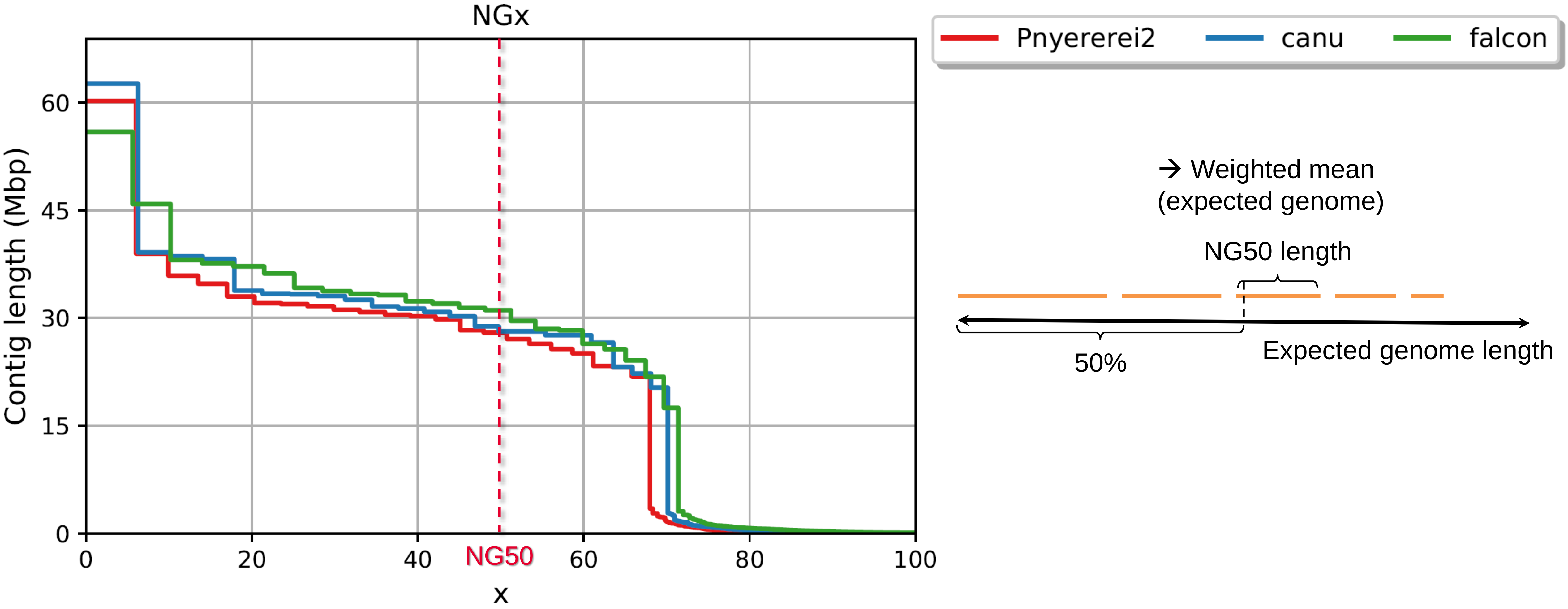
Assembly comparison







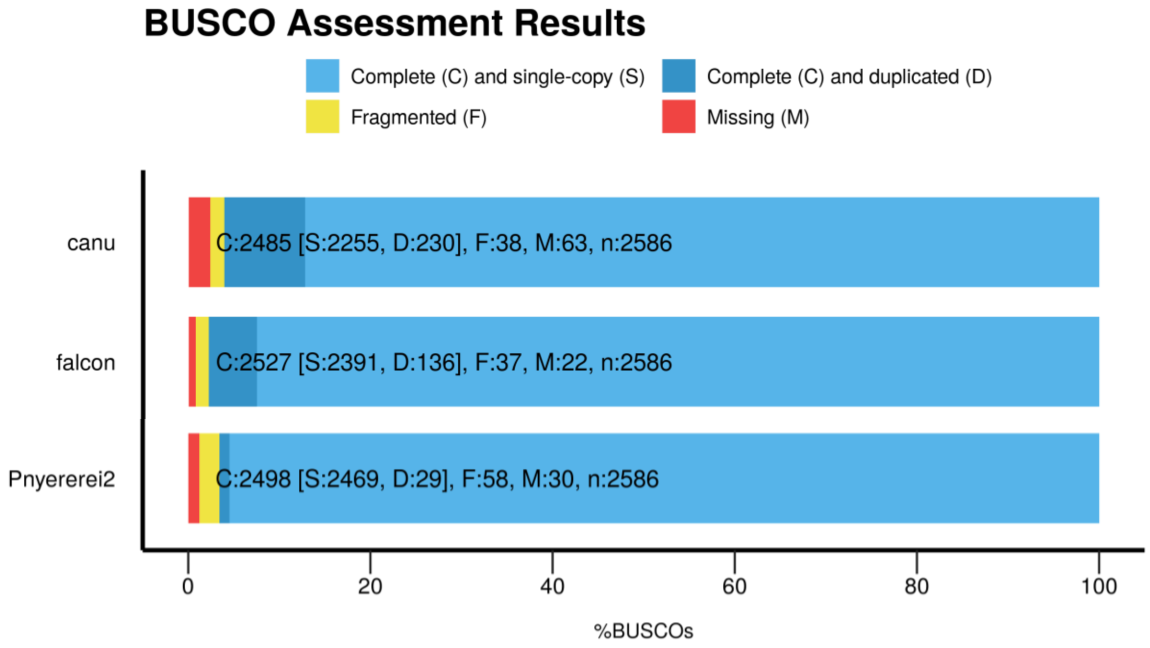
Assembly comparison



→ Weighted mean (expected genome)

Assembly comparison

Stats		Pnyererei2	canu	falcon
QUAST				
	nb scaffolds	6’876	5’753	2’832
	N50	29’830’996	27’606’787	29’593’783
	NG50	27’967’145	28’136’703	31’084’370
	max length	60’199’168	62’610’257	55’908’494
	total length	856’242’559	1’130’373’166	1’073’822’959
	total length without N	698’778’000	1’083’432’443	1’032’560’317
	N's per 100kb	18’390	4’153	3’843
BUSCO				
	complete	2’498	2’485	2’527
	complete single copy	2’469	2’255	2’391
	complete duplicated	29	230	136
	fragmented	58	38	37
	missing	30	63	22



Conclusion

- 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))