Using Oxford Nanopore for Long Read Sequencing and for Methylation Analysis

Jaz Sakr

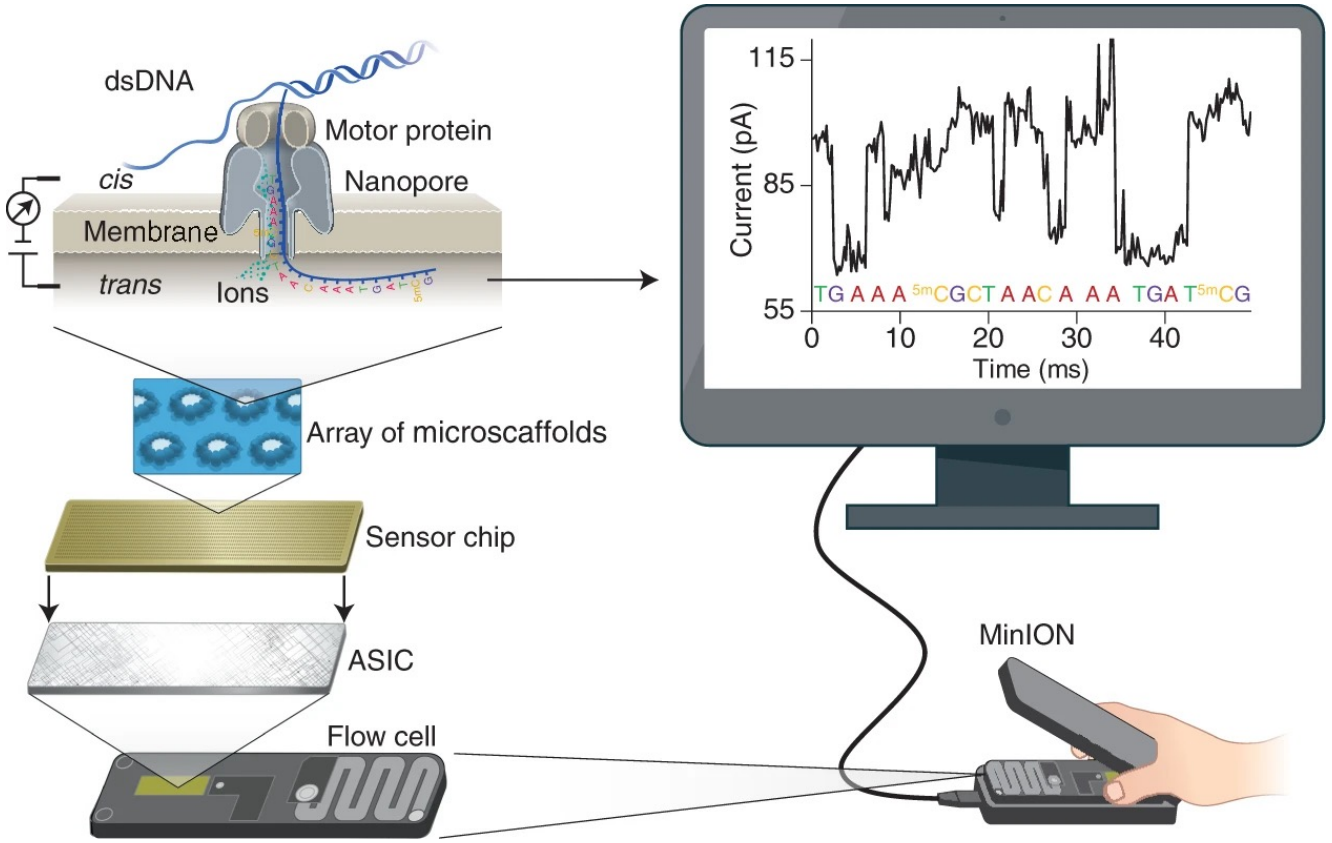
Ph.D. Candidate in Mortazavi Lab

Genomics Research and Technology Hub (GRT Hub) Workshop

DNA Methylation Data Analysis

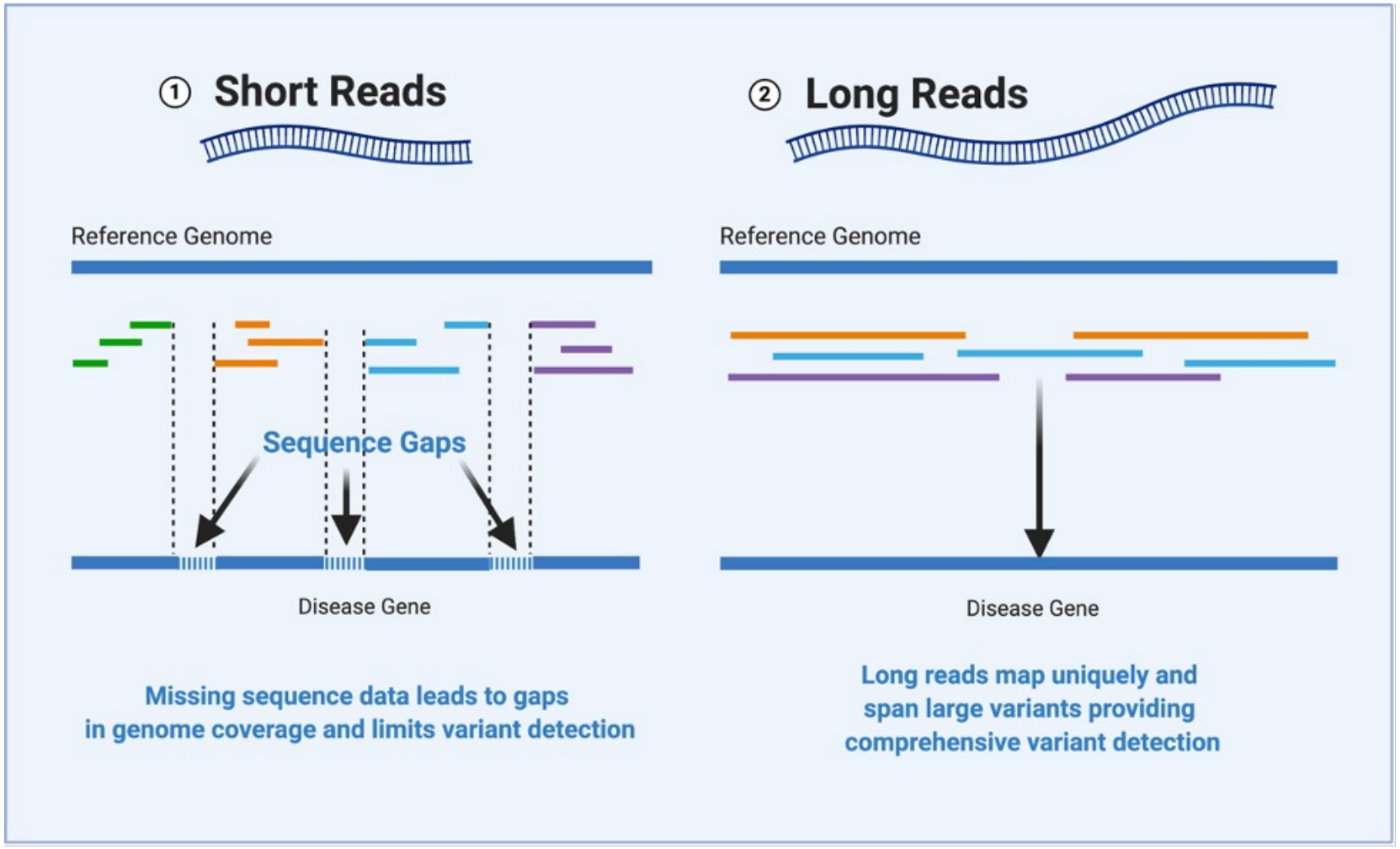
Outline

- Introduction
- Overview of Nanopore sequencing
- Nanopore Libraries
- Data acquisition
- Base calling
  - Guppy & Remora
- Methylation calling
  - Megalodon
- Visualization with IGV
- Differential Methylation Analysis



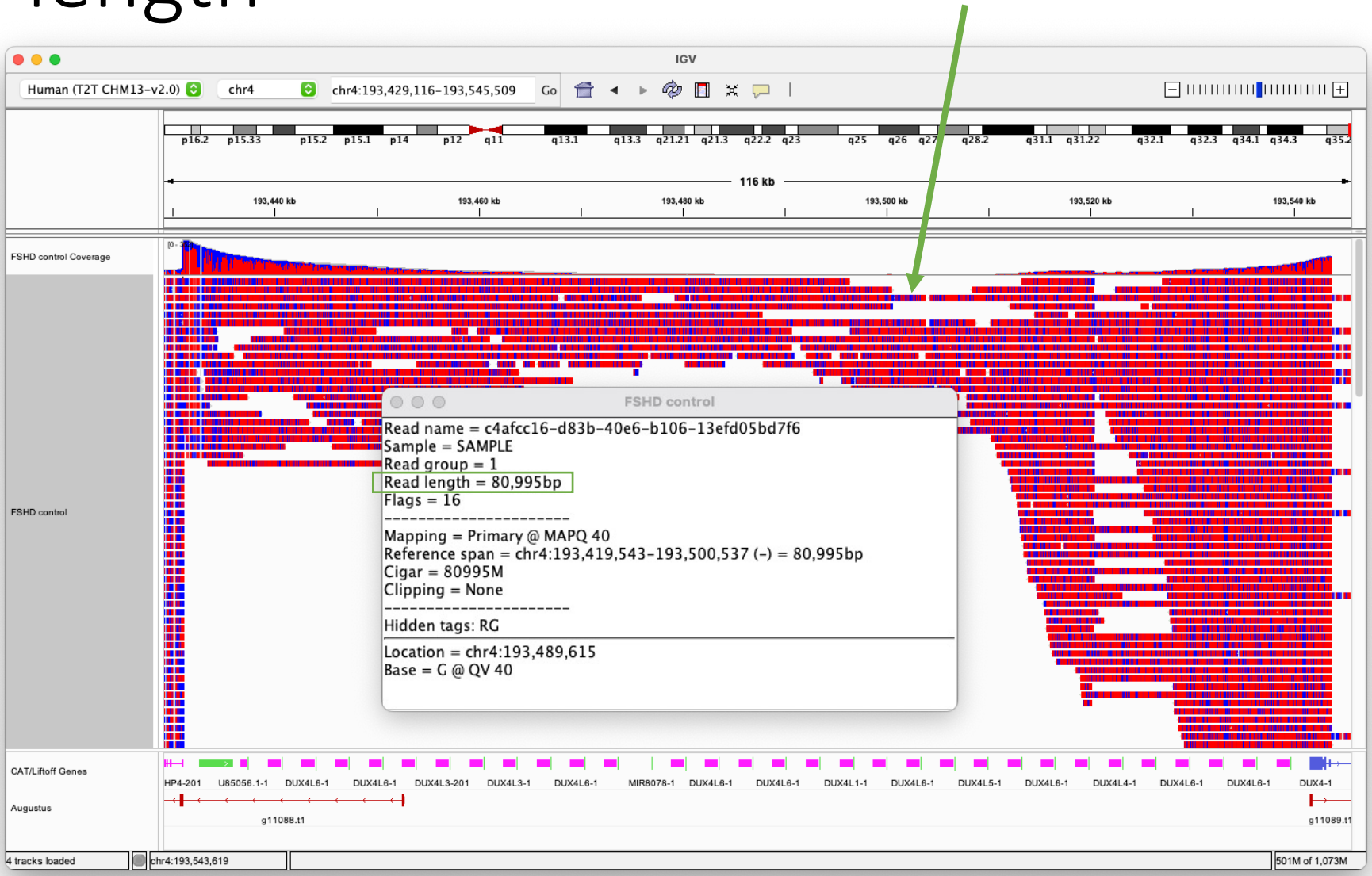
Wang et al., 2021

Long reads allow detection of genomic variation



HudsonAlpha 2021

Nanopore sequences DNA fragments tens of kb in length



Kong et al., 2023 submitted

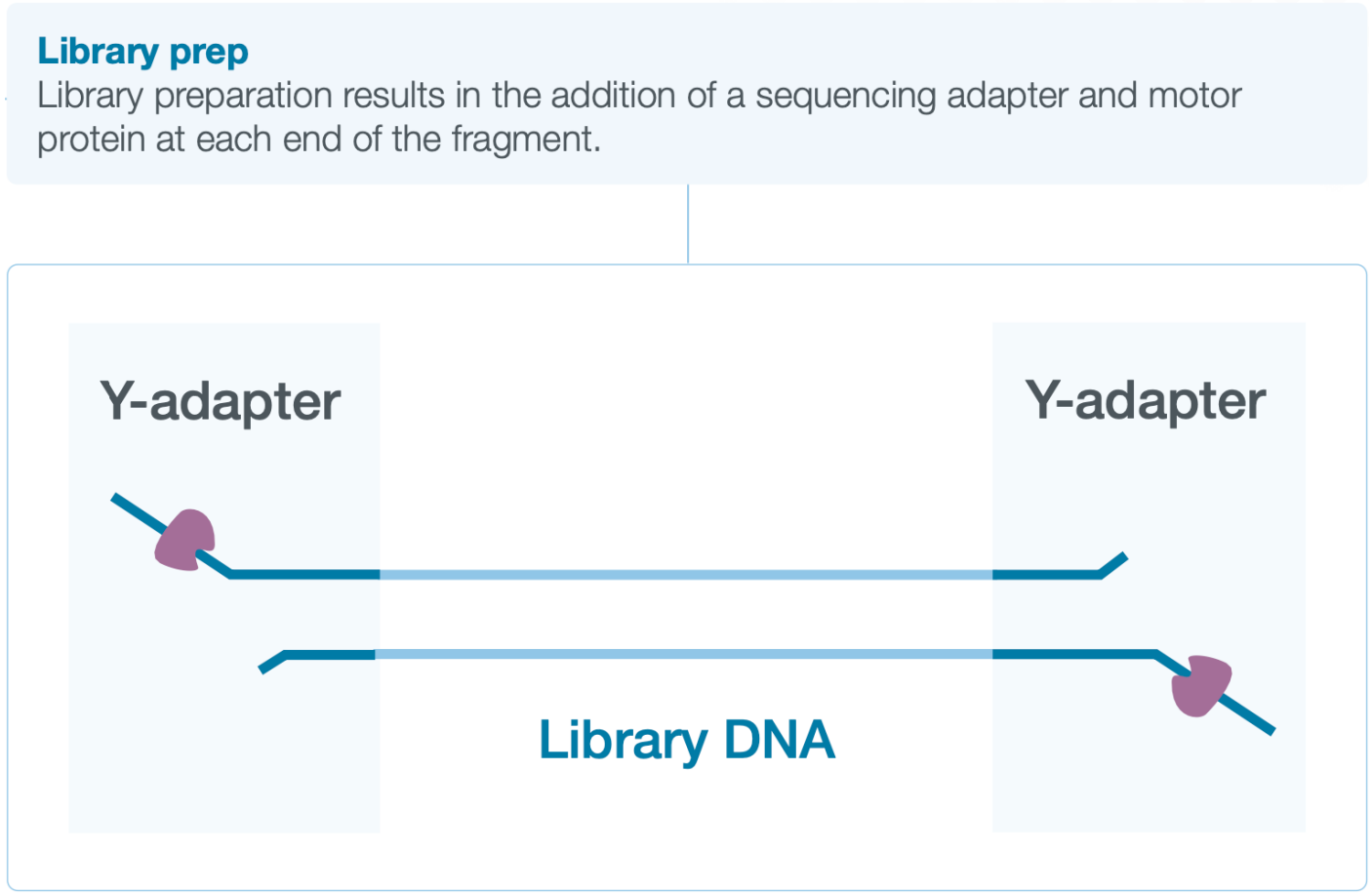
Nanopore is a long-read sequencing platform that sequences both DNA and RNA

	Native DNA			Amplified DNA		RNA		Targeted		
	Ligation	**Rapid/Field**	**Ultra-Long**	**PCR**	**Rapid PCR**	**PCR-cDNA**	**Direct RNA**	**16S**	**Cas9**	**Adaptive sampling**
Prep time	60 mins	10 mins	90 min + 1 x O/N incubation	60 mins + PCR	15 mins + PCR	160 mins + PCR	105 mins	10 mins + PCR	110 mins	–
Input	1,000 ng dsDNA	From 50 ng HMW gDNA	6M cells / 1 ml blood	100 ng dsDNA	1–5 ng gDNA	4 ng poly-A⁺ RNA, or 200 ng total RNA	500 ng RNA	10 ng gDNA	1–10 µg dsDNA	–
Multiplexing options	Yes	Yes	–	Yes	Yes	Yes	In development	Yes	Coming soon	–
Read length	Equal to fragment length	Random distribution, dependent on input fragment length	N50 >50 kb	Equal to fragment length post-PCR	~2 kb	Enriched for full-length cDNA	Equal to RNA length	Full-length 16S gene (~1.5kb)	Equal to fragment length	Equal to fragment length
PCR required	No	No	No	Yes	Yes	Yes	No	Yes	No	–
Product range highlights	Detect modified bases for free. Automatable workflows and XL kits enable production-scale sequencing			Ideal for low input amounts		Detect modified bases for free with direct RNA kits		Read more about adaptive sampling on page 15. Generate your own or view pre-defined panels on the Nanopore Community		

MinION and Flongle Flow Cell compatible

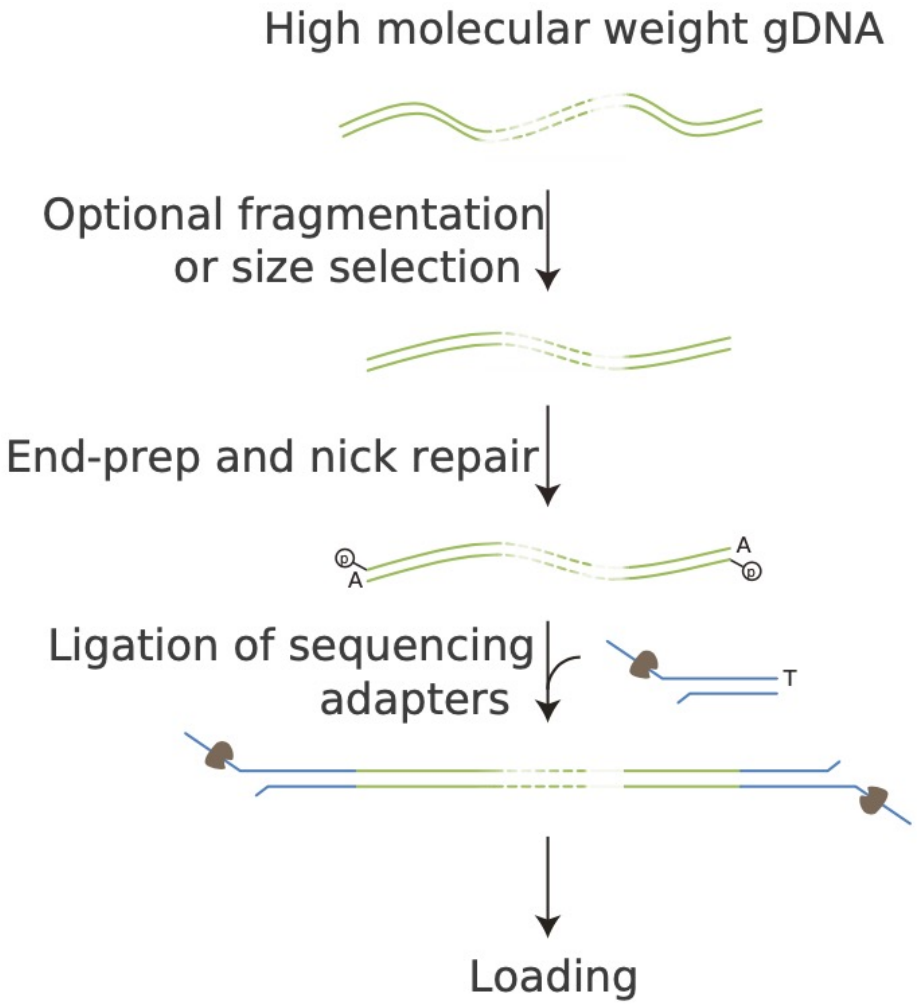
PromethION Flow Cell compatible

Nanopore library adapters contain characteristic motor protein



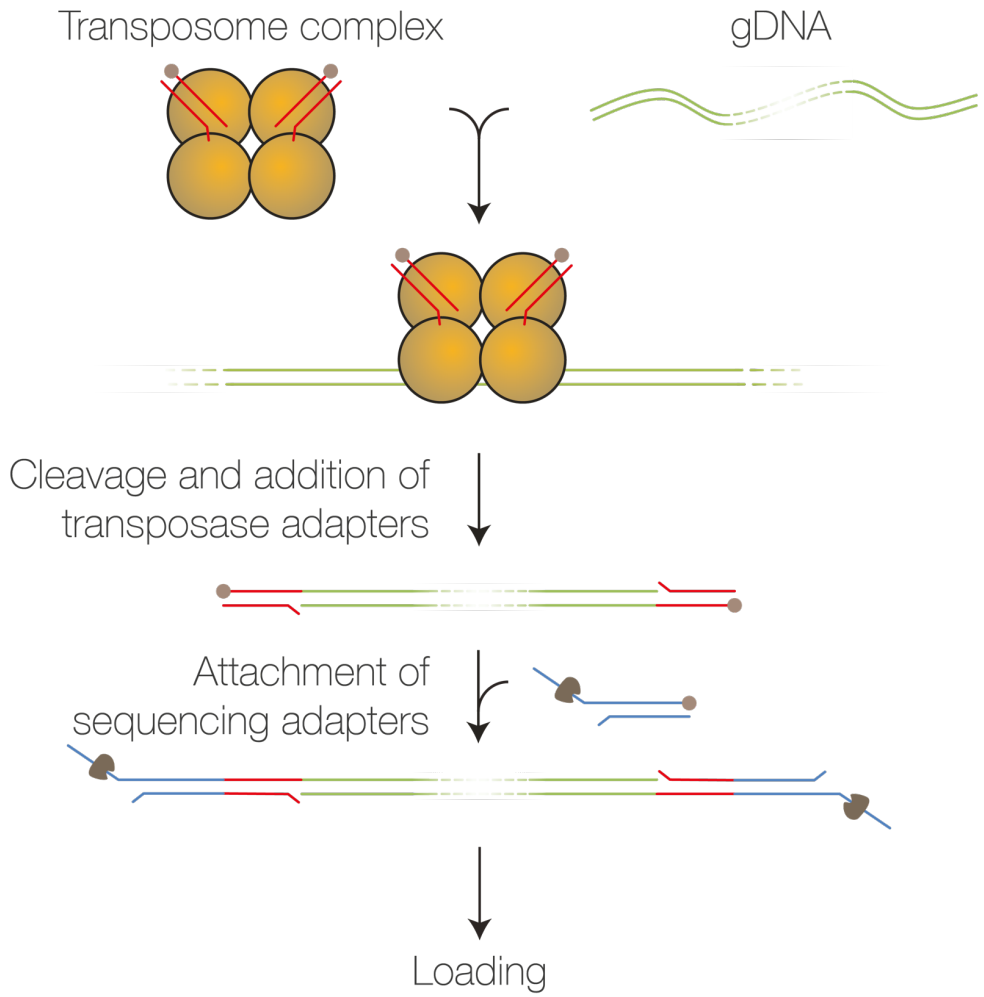
Nanopore directly sequences DNA

- Different kits for sequencing DNA
  - **Ligation**
  - Rapid/Field
  - Ultra-Long
  - Cas9



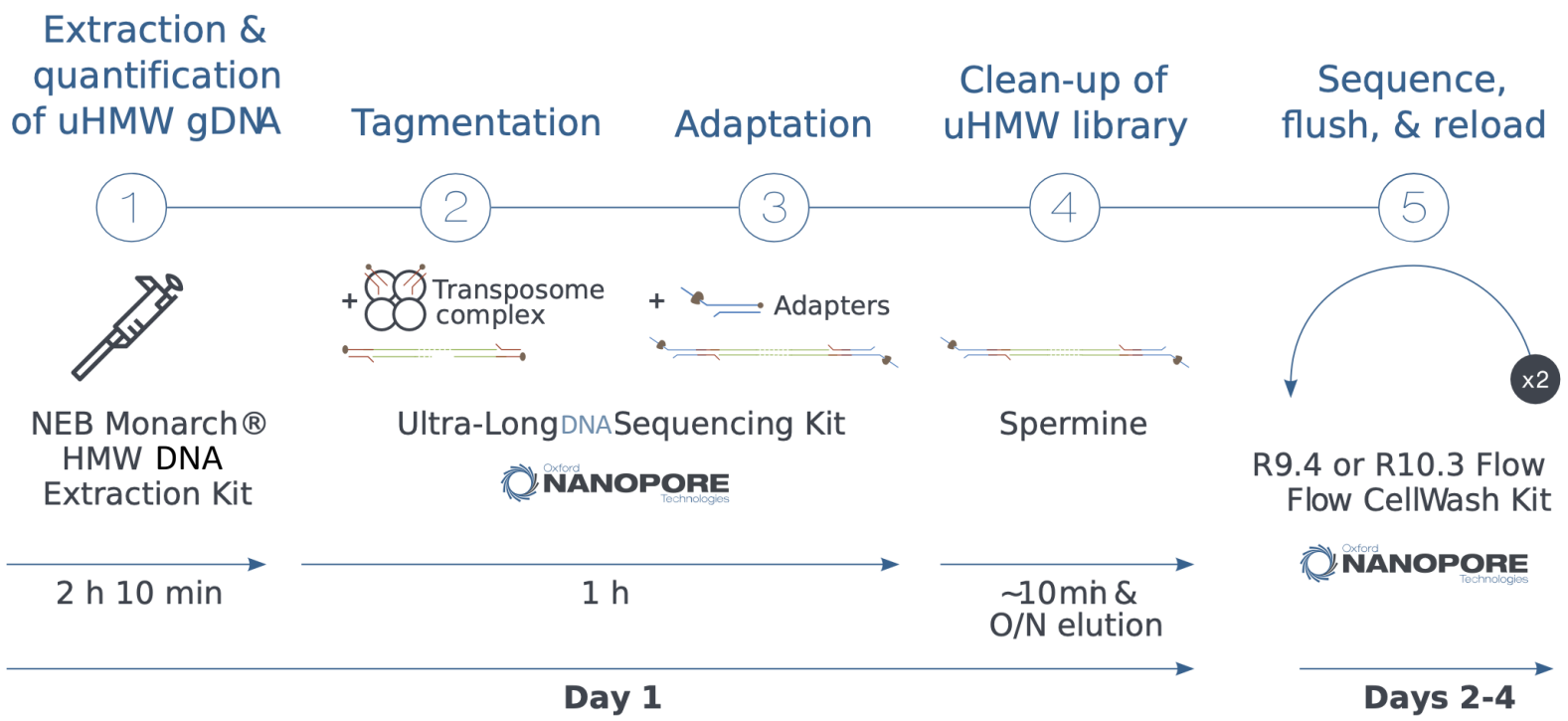
Nanopore directly sequences DNA

- Different kits for sequencing DNA
  - Ligation
  - **Rapid/Field**
  - Ultra-Long
  - Cas9



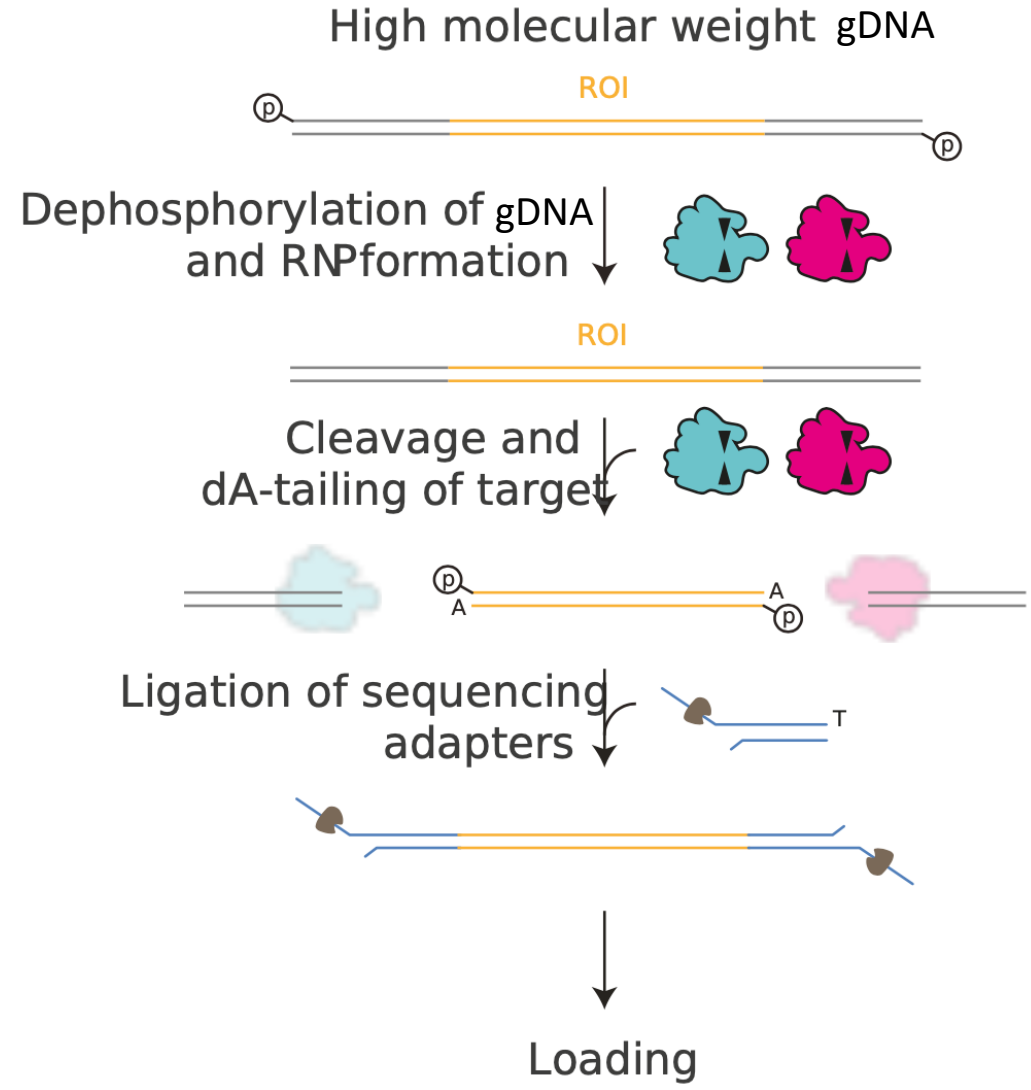
Nanopore directly sequences DNA

- Different kits for sequencing DNA
  - Ligation
  - Rapid/Field
  - **Ultra-Long**
  - Cas9



Nanopore directly sequences DNA

- Different kits for sequencing DNA
  - Ligation
  - Rapid/Field
  - Ultra-Long
  - **Cas9**

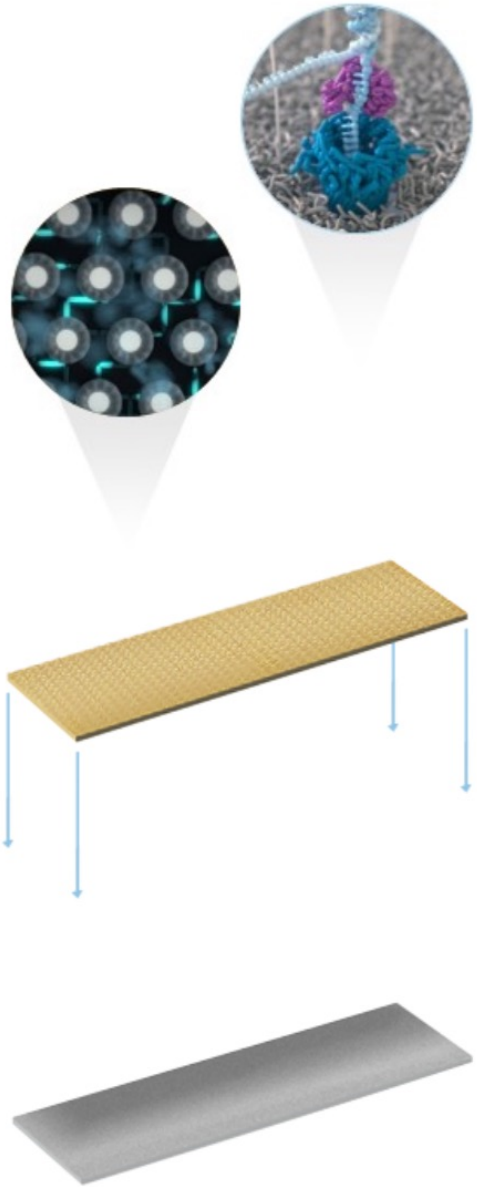


Libraries are loaded onto a flow cell

Flongle flow cell
Up to 2.8 Gb

MinION flow cell
Up to 50 Gb

PromethION flow cell
Up to 290 Gb



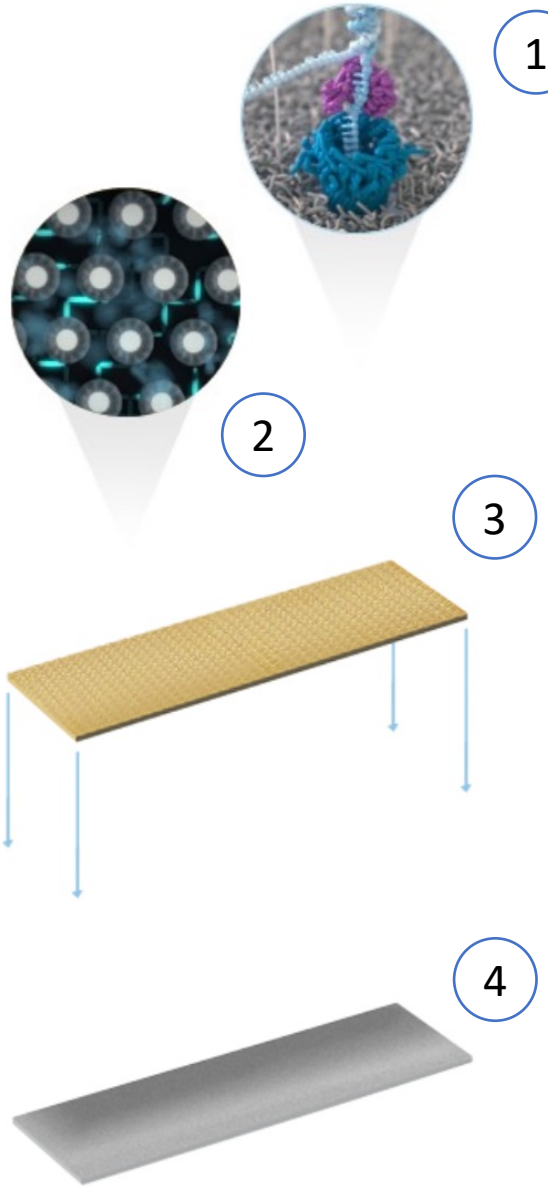
Libraries are loaded onto a flow cell

1. **Nanopore**
   A protein nanopore is set in an electrically-resistant polymer membrane.

2. **Array of microscaffolds**
   Each microscaffold supports a membrane and embedded nanopore.

3. **Sensor chip**
   Each microscaffold corresponds to its own electrode that is connected to a channel in the sensor array chip.

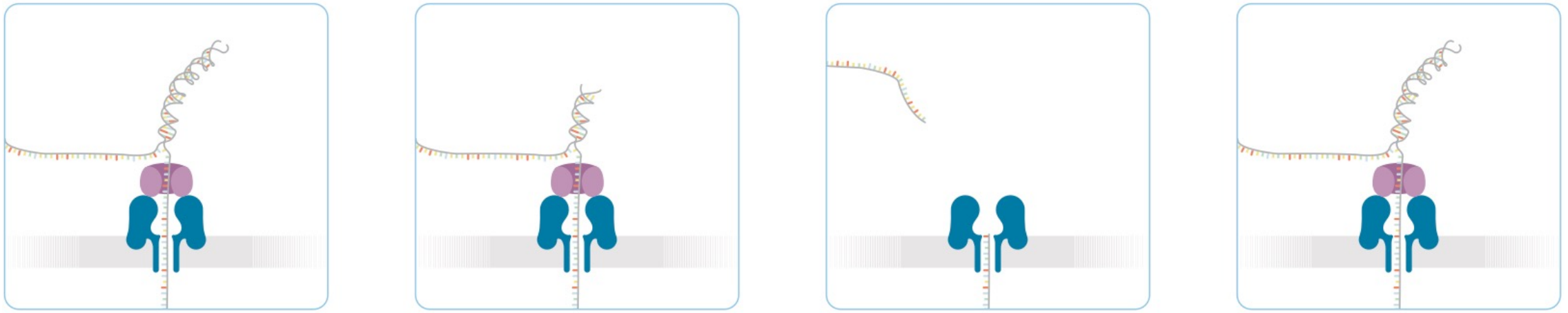
4. **ASIC**
   Each nanopore channel is controlled and measured individually by the bespoke Application-Specific Integrated Circuit (ASIC). This allows for multiple nanopore experiments to be performed in parallel.



Libraries are passed through a nanopore

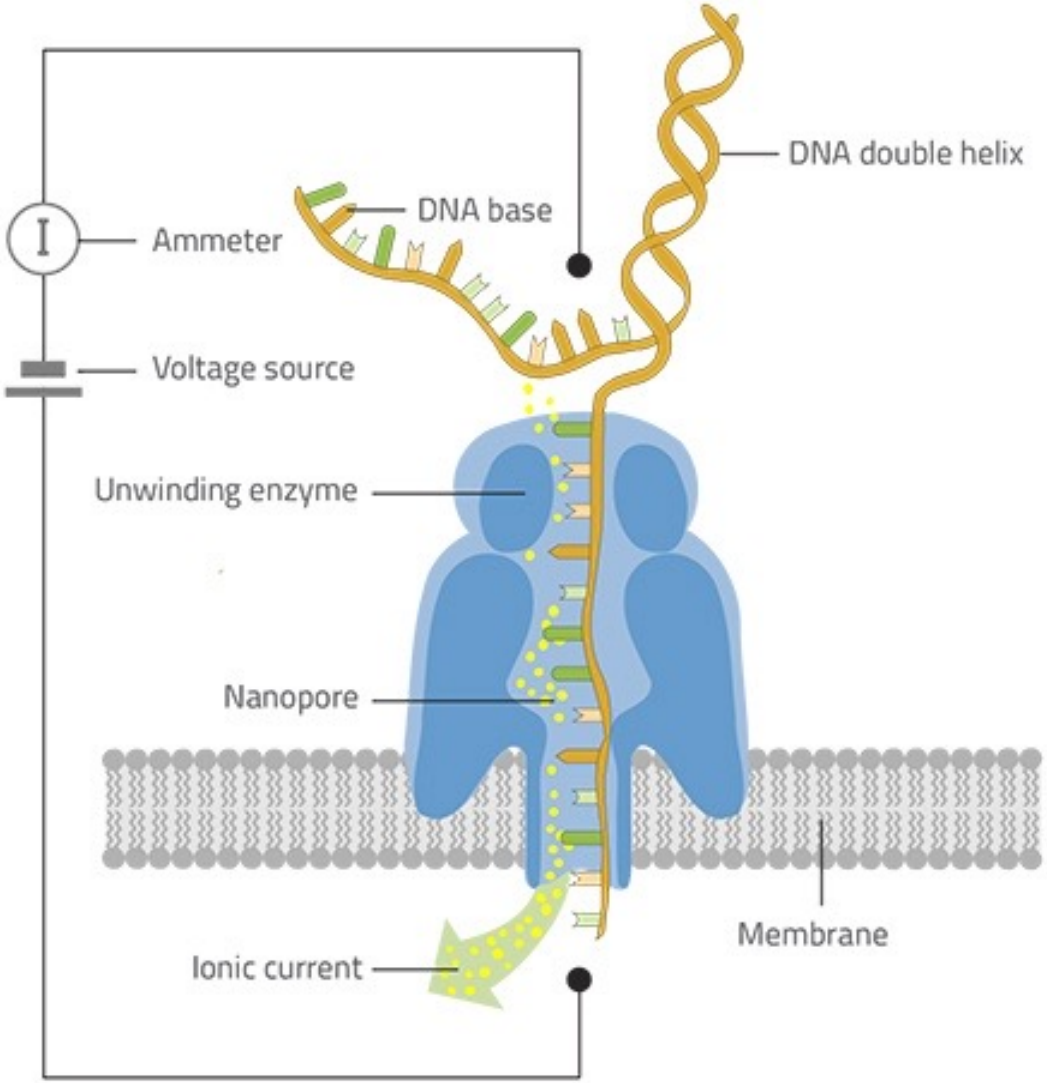
Translocation

Both the template and complement strands carry the motor protein which means both strands are able to translocate the nanopore.



Template... ...Template... (Exit) Next molecule…

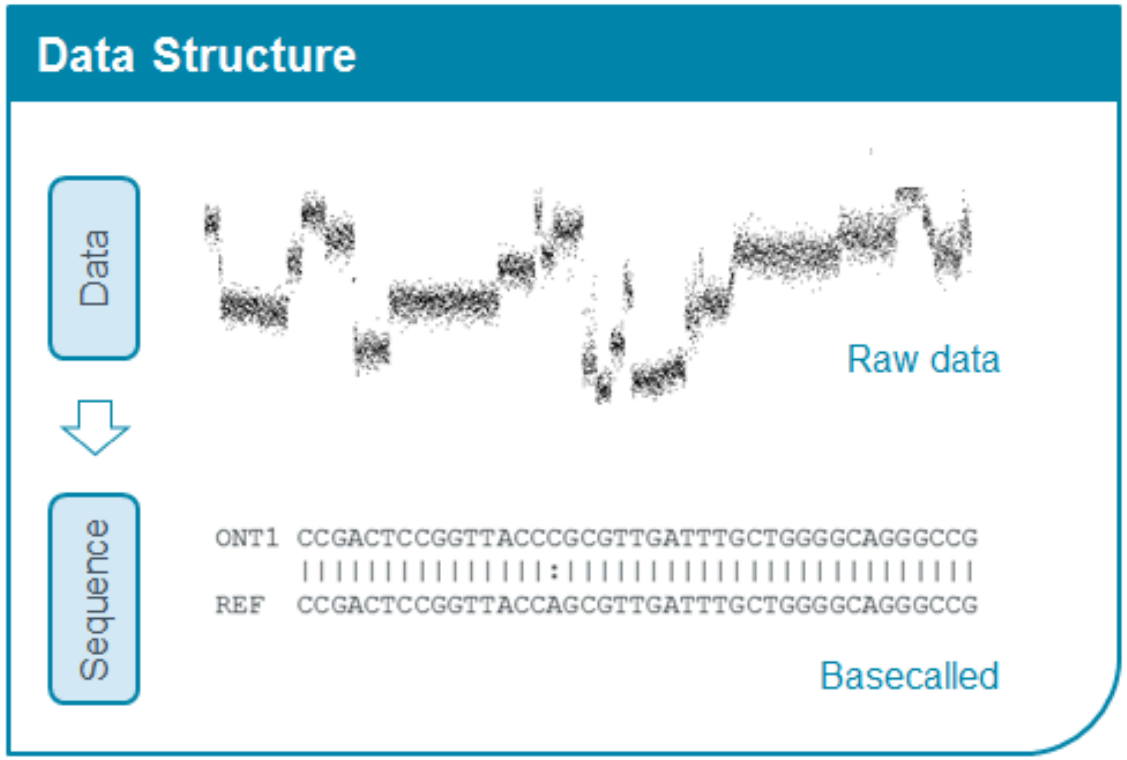
Libraries are passed through a nanopore



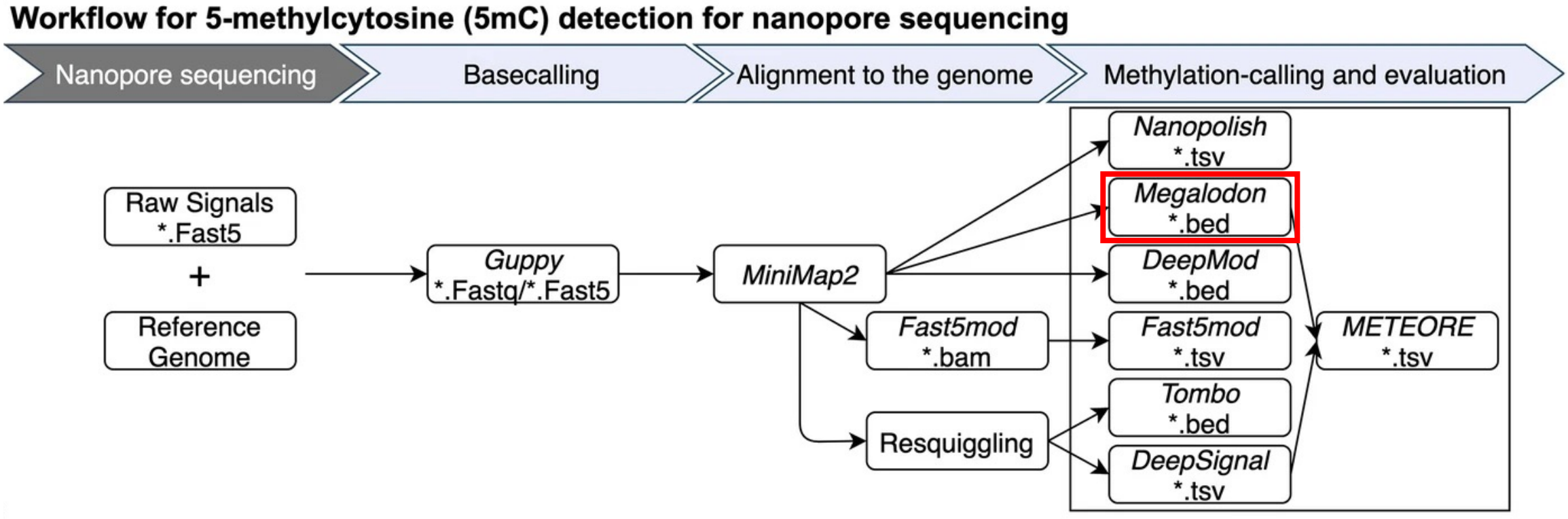
Göpfrich & Judge, 2018

MinKNOW sequencing software records voltages changes as raw signal

- Raw signal is recorded as Fast5 files
- Fast5 files are in the HDF5 format



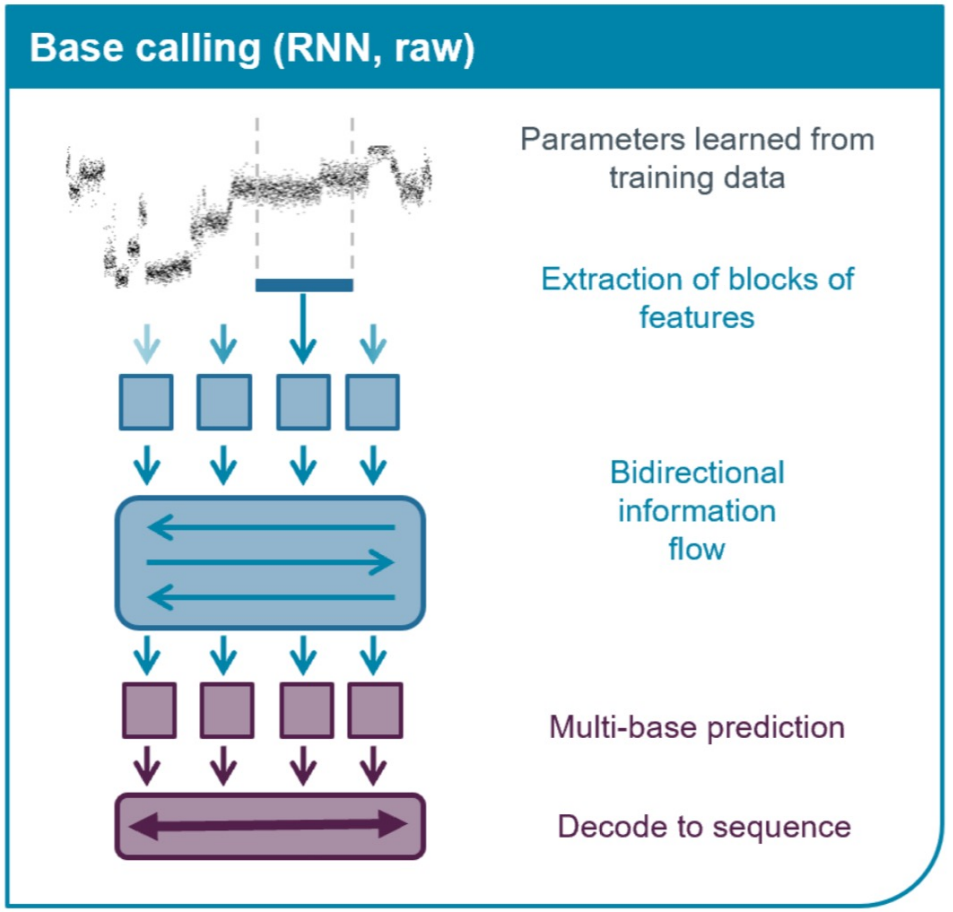
Methylation analysis workflow



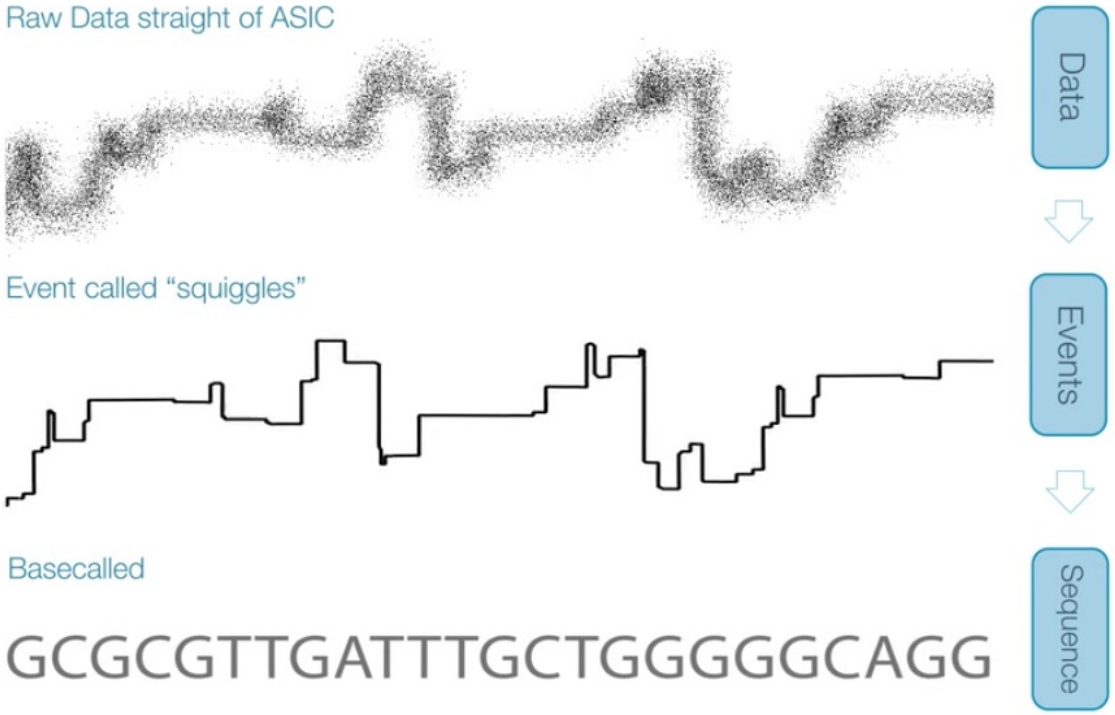
Liu et al., 2021

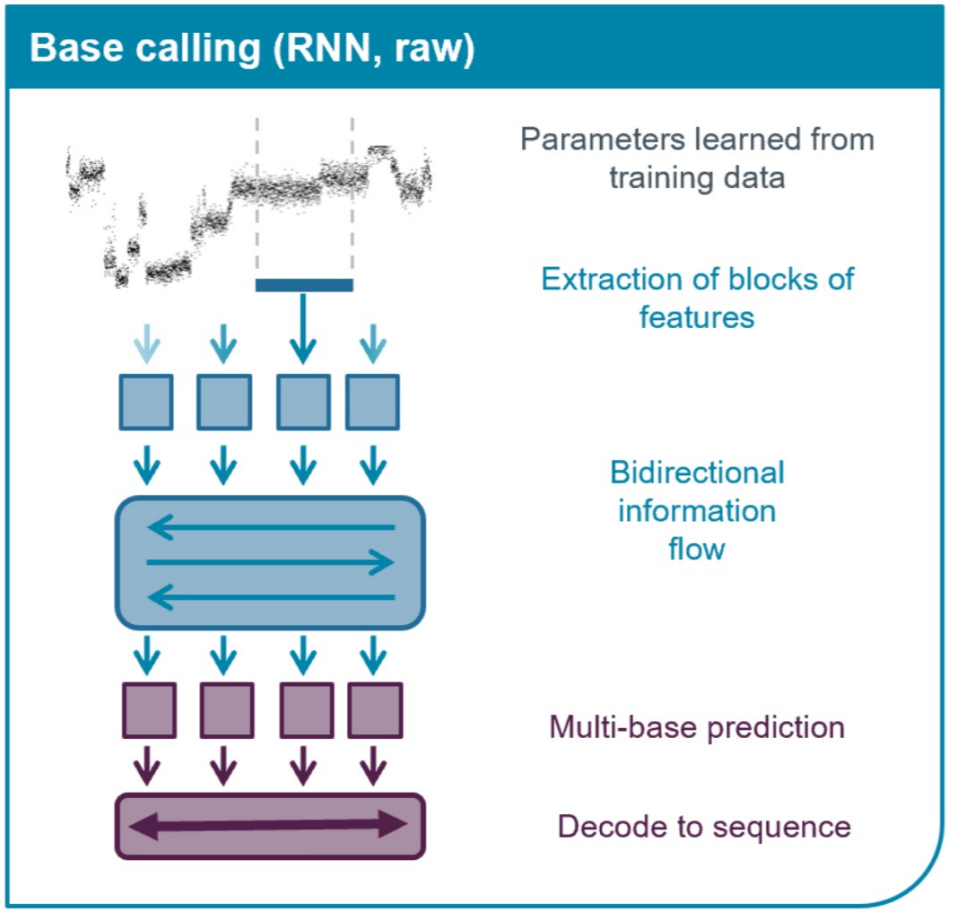
Guppy uses neural networks for basecalling.

- Guppy is the **default basecaller** on all Oxford Nanopore sequencing devices
- A **recurrent neural network** is a class of neural networks in which the output is dependent on past computations.
- A **bi-directional RNN** can set data in the context of what comes both before *and* after in the signal.



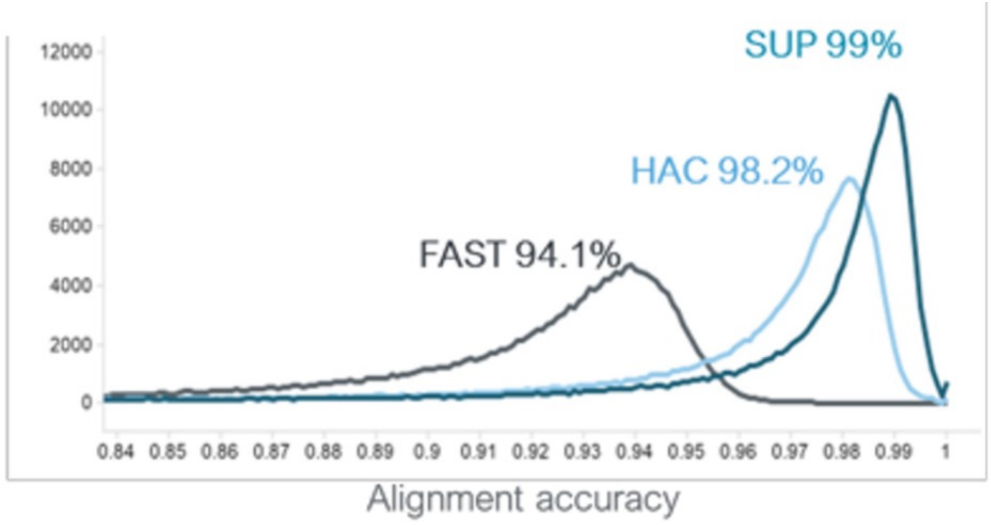
Guppy uses neural networks for basecalling.

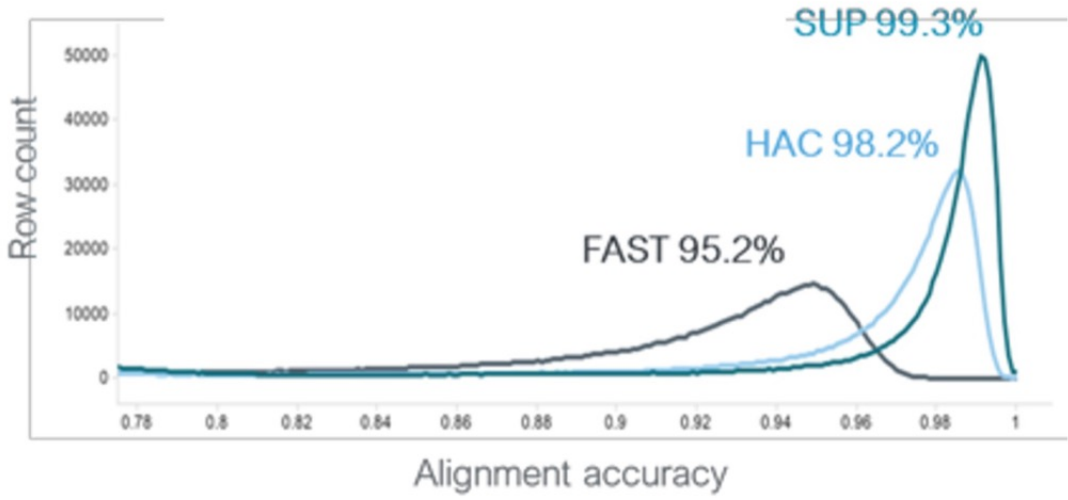




Guppy has different basecalling modes

- Fast, High accuracy (HAC), and Super accurate (SUP)



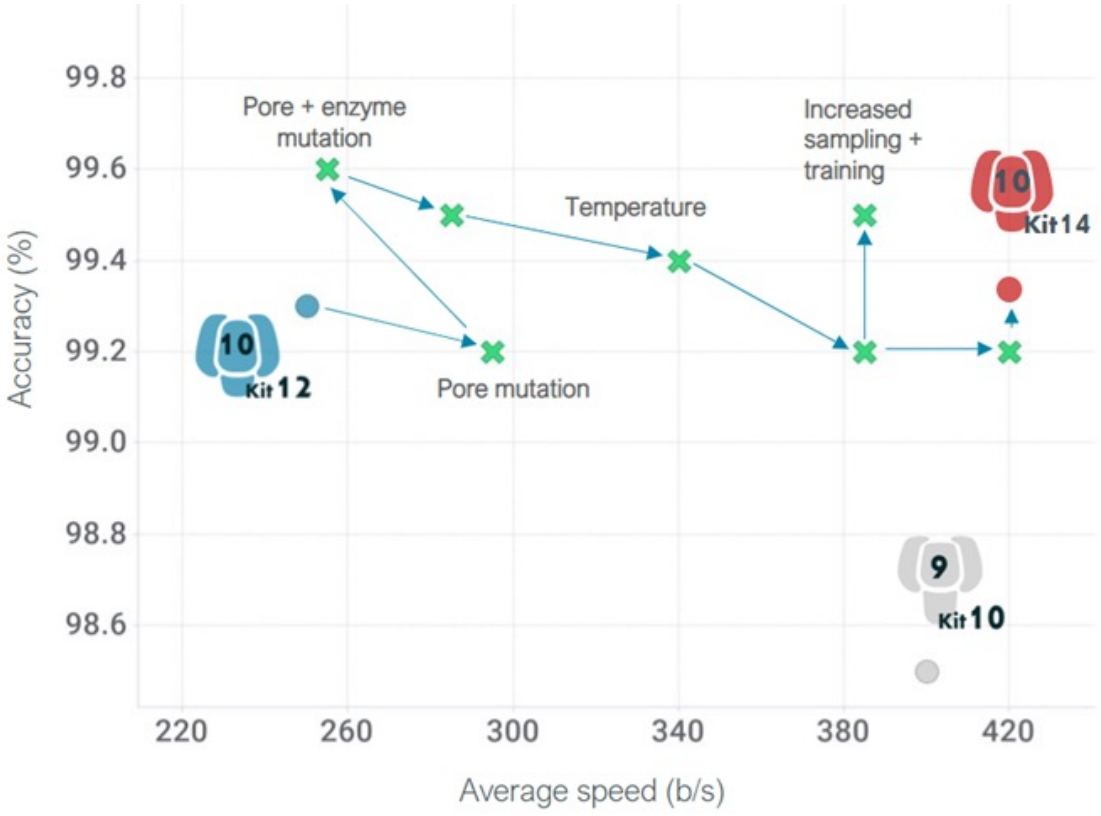


Sequencing Speed:

Default (400 bps)

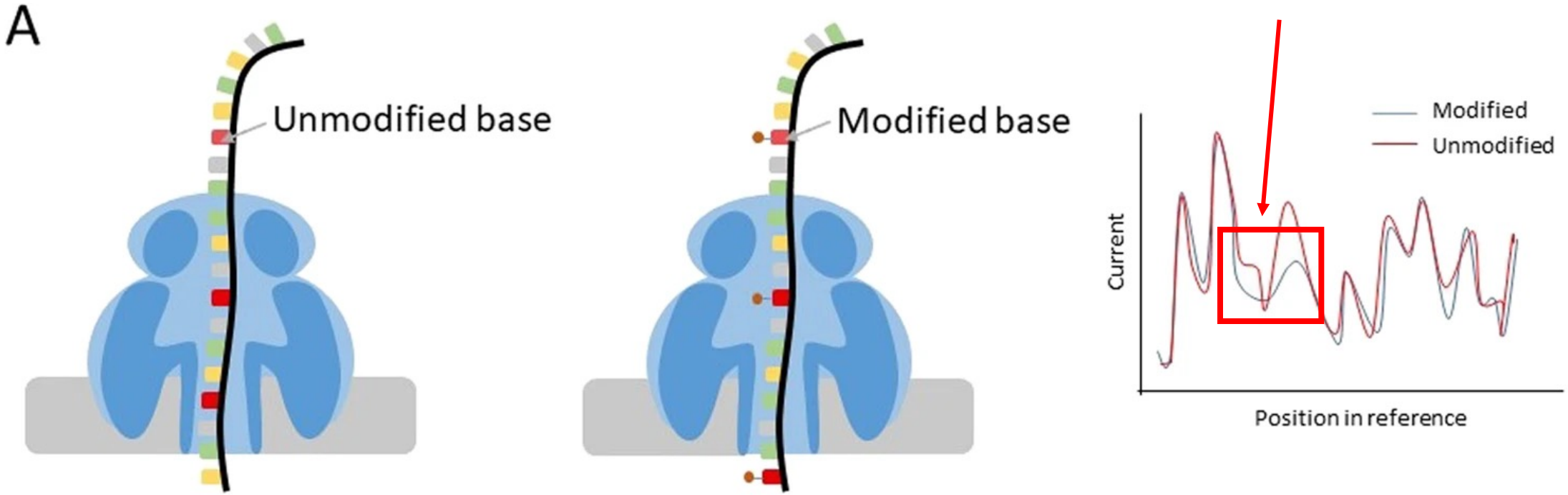
Accurate (260 bps)

Configuration files and calling accuracy during analysis depends on sequencing chemistries



Flow cell	Kit	Raw read accuracy	Analysis tools
R10.4.1	SQK-LSK114 260bps	99.6% modal (Q24, simplex)	"Super accuracy" basecaller in MinKNOW
R10.4.1	SQK-LSK114 260 bps	99.92% modal (Q31, duplex)	Duplex basecaller in Guppy
R10.4	SQK-LSK112	>99.3% modal	"Super accuracy" basecaller in MinKNOW
R9.4.1 *	SQK-LSK110	98.3% modal	"Super accuracy" basecaller in MinKNOW

Directly sequencing DNA enables modification detection



Xu & Seki, 2018

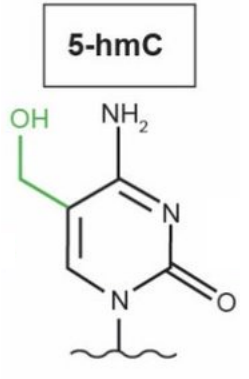
Calling modified bases with Remora

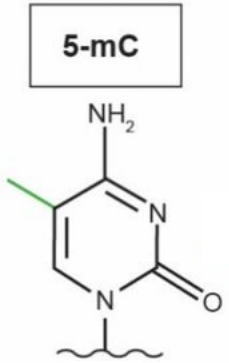
- Remora is designated basecalling model that is trained to identify base modifications

- Guppy uses Remora model to call methylation.

Current methylation detection capabilities with Remora

Flow cell/Pore type	Kit	Available Remora models
R9.4.1 (FLO-MIN106, FLO-PRO002, FLO-FLG001)	E8 (Kit 9, 10, 11)	5mc, 5hmC_5mC
R9.4.1 (FLO-MIN106, FLO-PRO002, FLO-FLG001)	E8.1 (Kit 12)	5mC
R10.4 (FLO-MIN112, FLO-PRO112)	E8.1 (Kit 12)	5mC, 5hmC_5mC
R10.4.1 (FLO-MIN114, FLO-PRO114)	E8.2 (Kit 14)	5mC





Reference-anchored methylation calling with Megalodon

- **Megalodon** is a command line tool from Oxford Nanopore that extracts modified base and sequence variant calls from raw nanopore reads by anchoring the information rich basecalling neural network output to a reference genome/transcriptome.

- **Outputs include**: basecalls (FASTA/Q), reference mappings (SAM/BAM/CRAM), modified base calls (per-read and bedgraph/bedmethyl/modVCF), sequence variant calls (per-read and VCF)

- **Requirements:** Guppy and Remora

GPU vs CPU

- Basecalling is computationally intensive
- Both Guppy and Megalodon run optimally by using GPU, which is highly recommended
- CPU usage is also possible
- Important consideration for HPC3 users

Megalodon usage example

```
megalodon ./fast5 \
--guppy-server-path /usr/bin/guppy_basecall_server \
--guppy-config dna_r9.4.1_450bps_modbases_5mc_cg_sup.cfg \
--guppy-params "--num_callers 4 --gpu_runners_per_device 2 --chunks_per_runner 120" \
--remora-modified-bases dna_r9.4.1_e8 sup 0.0.0 5mc CG 0 \
--outputs mod_mappings \
--sort-mappings \
--mod-map-emulate-bisulfite \
--mod-map-base-conv C T \
--mod-map-base-conv m C \
--reference /references/hg38.fa \
--output-directory ./megalodon_output \
--devices cuda:0,1
```

Megalodon usage example

```
megalodon ./fast5 \
--guppy-server-path /usr/bin/guppy_basecall_server \
--guppy-config dna_r9.4.1_450bps_modbases_5mc_cg_sup.cfg \
--guppy-params "--num_callers 4 --gpu_runners_per_device 2 --chunks_per_runner 120" \
--remora-modified-bases dna_r9.4.1_e8 sup 0.0.0 5mc CG 0 \
--outputs mod_mappings \
--sort-mappings \
--mod-map-emulate-bisulfite \
--mod-map-base-conv C T \
--mod-map-base-conv m C \
--reference /references/hg38.fa \
--output-directory ./megalodon_output \
--devices cuda:0,1
```

Megalodon usage example

```
megalodon ./fast5 \
--guppy-server-path /usr/bin/guppy_basecall_server \
--guppy-config dna_r9.4.1_450bps_modbases_5mc_cg_sup.cfg \
--guppy-params "--num_callers 4 --gpu_runners_per_device 2 --chunks_per_runner 120" \
--remora-modified-bases dna_r9.4.1_e8 sup 0.0.0 5mc CG 0 \
--outputs mod_mappings \
--sort-mappings \
--mod-map-emulate-bisulfite \
--mod-map-base-conv C T \
--mod-map-base-conv m C \
--reference /references/hg38.fa \
--output-directory ./megalodon_output \
--devices cuda:0,1
```

Megalodon usage example

```
megalodon ./fast5 \
--guppy-server-path /usr/bin/guppy_basecall_server \
--guppy-config dna_r9.4.1_450bps_modbases_5mc_cg_sup.cfg \
--guppy-params "--num_callers 4 --gpu_runners_per_device 2 --chunks_per_runner 120" \
--remora-modified-bases dna_r9.4.1_e8 sup 0.0.0 5mc CG 0 \
--outputs mod_mappings \
--sort-mappings \
--mod-map-emulate-bisulfite \
--mod-map-base-conv C T \
--mod-map-base-conv m C \
--reference /references/hg38.fa \
--output-directory ./megalodon_output \
--devices cuda:0,1
```

Megalodon usage example

```
megalodon ./fast5 \
--guppy-server-path /usr/bin/guppy_basecall_server \
--guppy-config dna_r9.4.1_450bps_modbases_5mc_cg_sup.cfg \
--guppy-params "--num_callers 4 --gpu_runners_per_device 2 --chunks_per_runner 120" \
--remora-modified-bases dna_r9.4.1_e8 sup 0.0.0 5mc CG 0 \
--outputs mod_mappings \
--sort-mappings \
--mod-map-emulate-bisulfite \
--mod-map-base-conv C T \
--mod-map-base-conv m C \
--reference /references/hg38.fa \
--output-directory ./megalodon_output \
--devices cuda:0,1
```

Megalodon usage example

```
megalodon ./fast5 \
--guppy-server-path /usr/bin/guppy_basecall_server \
--guppy-config dna_r9.4.1_450bps_modbases_5mc_cg_sup.cfg \
--guppy-params "--num_callers 4 --gpu_runners_per_device 2 --chunks_per_runner 120" \
--remora-modified-bases dna_r9.4.1_e8 sup 0.0.0 5mc CG 0 \
--outputs mod_mappings \
--sort-mappings \
--mod-map-emulate-bisulfite \
--mod-map-base-conv C T \
--mod-map-base-conv m C \
--reference /references/hg38.fa \
--output-directory ./megalodon_output \
--devices cuda:0,1
```

Megalodon usage example

```
megalodon ./fast5 \
--guppy-server-path /usr/bin/guppy_basecall_server \
--guppy-config dna_r9.4.1_450bps_modbases_5mc_cg_sup.cfg \
--guppy-params "--num_callers 4 --gpu_runners_per_device 2 --chunks_per_runner 120" \
--remora-modified-bases dna_r9.4.1_e8 sup 0.0.0 5mc CG 0 \
--outputs mod_mappings \
--sort-mappings \
--mod-map-emulate-bisulfite \
--mod-map-base-conv C T \
--mod-map-base-conv m C \
--reference /references/hg38.fa \
--output-directory ./megalodon_output \
--devices cuda:0,1
```

Some other Megalodon output options

- **--mod-output-formats bedmthyl**: bed9+2 file containing the number of reads and the percent methylation (as specified by the ENCODE consortium)

- **--write-mod-log-probs:** per-read modified base log probabilities out in non-standard VCF field.

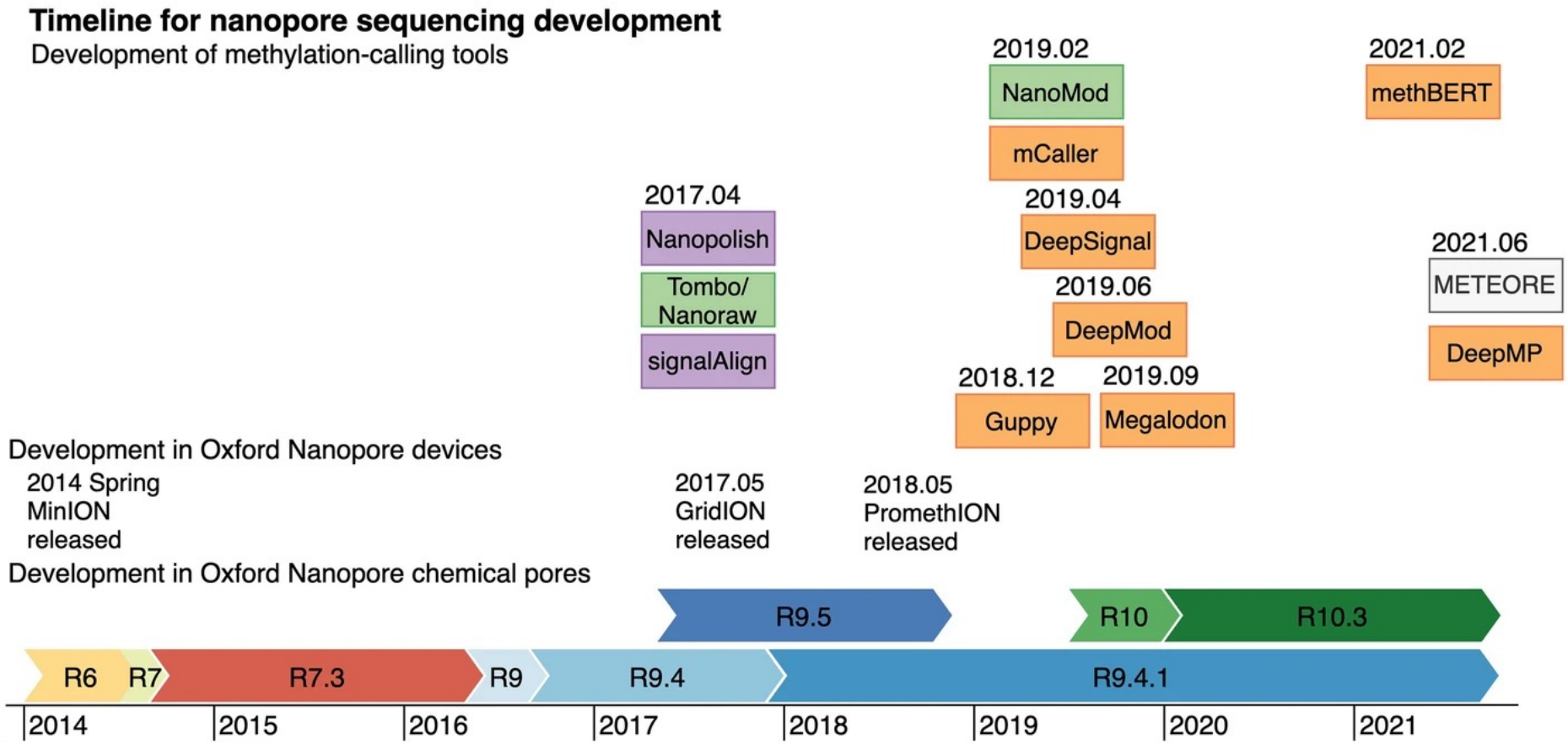
- **--write-mods-text:** per-read modified bases in text format.

Running Megalodon on command line

- This is an example output of running Megalodon in the terminal (NOT as a bash script/slurm job)

```
(base) jsakr@watson:/media/backup_disk/nanopore/data/fshd/fshd005$ megalodon ./  --guppy-server-path /usr/bin/guppy_basecall_server  --gu
ppy-config dna_r9.4.1_450bps_modbases_5mc_cg_sup.cfg  --guppy-params "--num_callers 4 --gpu_runners_per_device 2 --chunks_per_runner 120"
  --remora-modified-bases dna_r9.4.1_e8 sup 0.0.0 5mc CG 0  --outputs mod_mappings  --sort-mappings  --mod-map-emulate-bisulfite  --mod-m
ap-base-conv C T  --mod-map-base-conv m C  --reference /home/jsakr/references/chm13_v2.0-t2t.fa  --output-directory ./megalodon_output  -
-devices cuda:0,1
[09:48:27] Running Megalodon version 2.5.0
[09:48:27] Loading guppy basecalling backend
[09:48:31] Loading reference
[09:49:38] Loaded Remora model calls modified bases: m=5mC (alt to C)
[09:49:38] Preparing workers to process reads
[09:49:39] Processing reads
Full output or empty input queues indicate I/O bottleneck
3 most common unsuccessful processing stages:
     3.5% (    153 reads) : No alignment
    -----
    -----
Read Processing:  17%|████████                                          | 4472/26712 [20:32<1:42:10,  3.63reads/s, samples/s=4.59e+5]
 input queue capacity extract_signal      : 100%|████████████████████████████████████████████████████████████| 10000/10000
output queue capacity per_read_mods       :   0%|                                                             | 14/10000
```

Other methylation callers



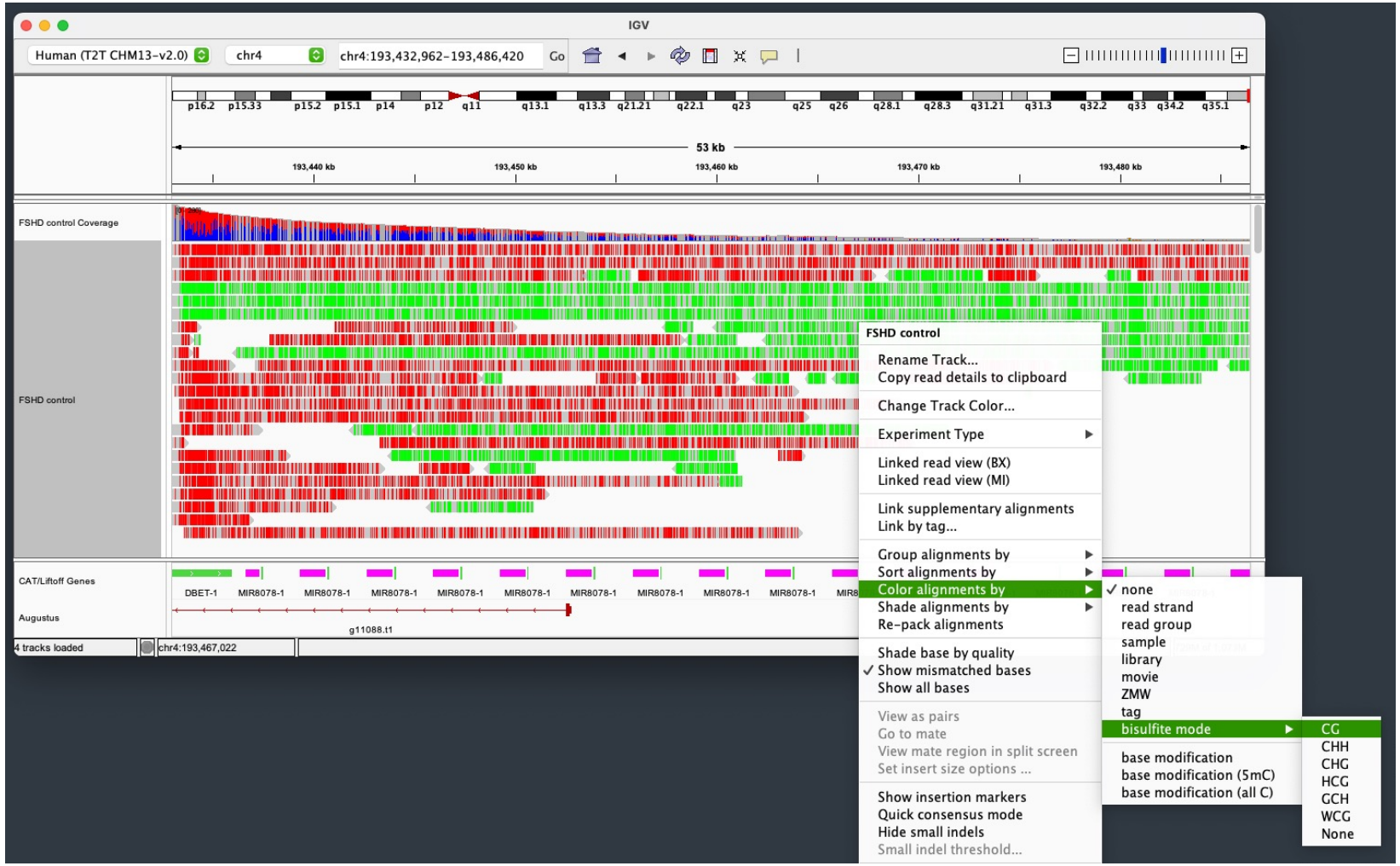
Liu et al., 2021

Visualization with Integrative Genomics Viewer (IGV)

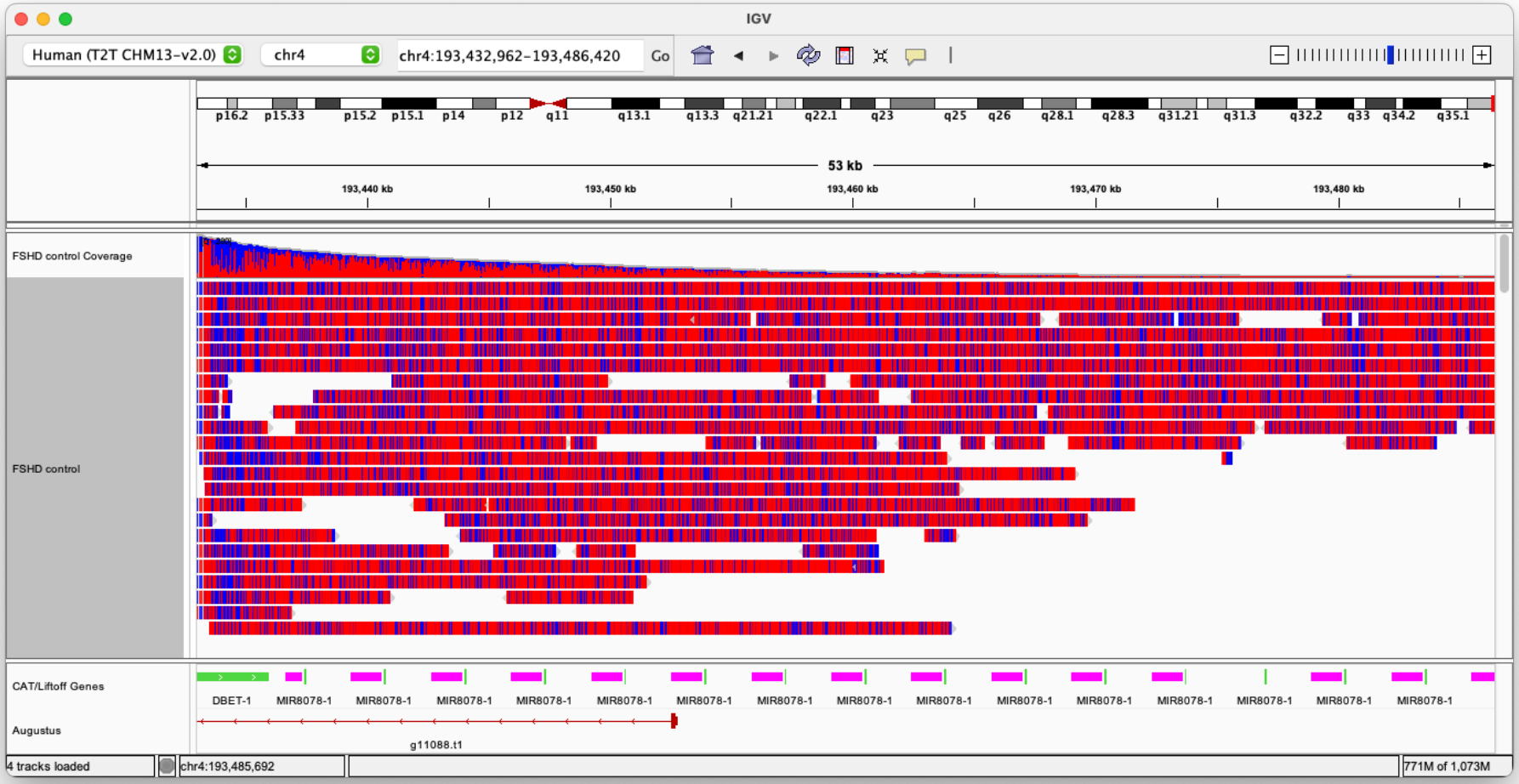
- IGV is a genome browser that allows you to visualize read mapping (like the UCSC Genome browser)

- IGV has a "**bisulfite mode**" that color codes hypomethylated regions in blue and hypermethylated regions in red.

Enabling bisulfite mode in IGV

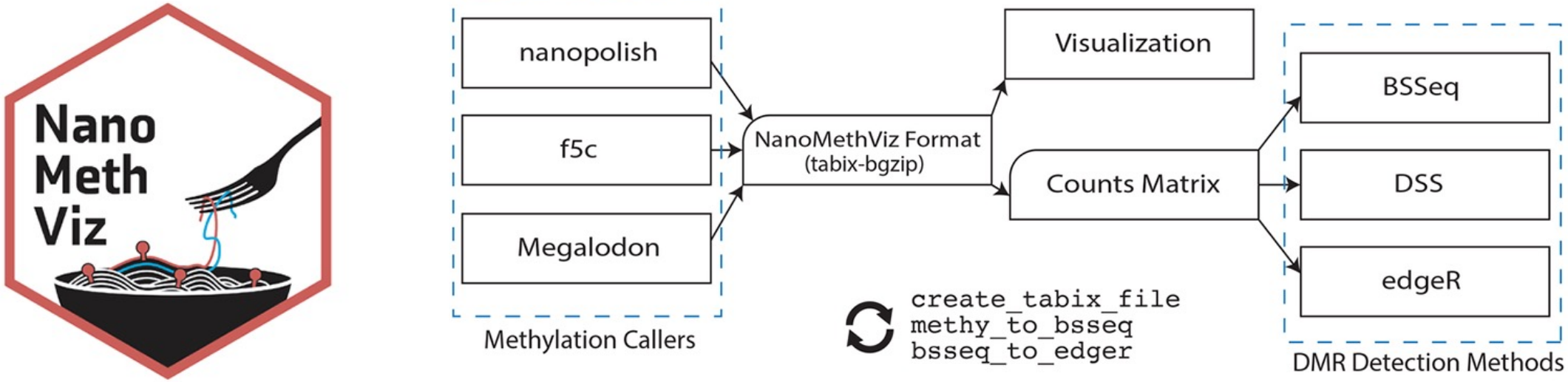


Enabling bisulfite mode in IGV



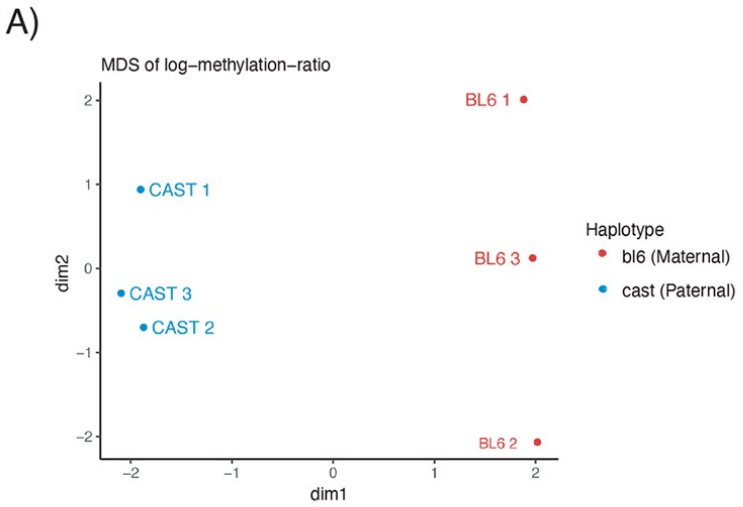
Using **NanoMethViz** for differentially methylated regions (DMRs) analysis post Megalodon

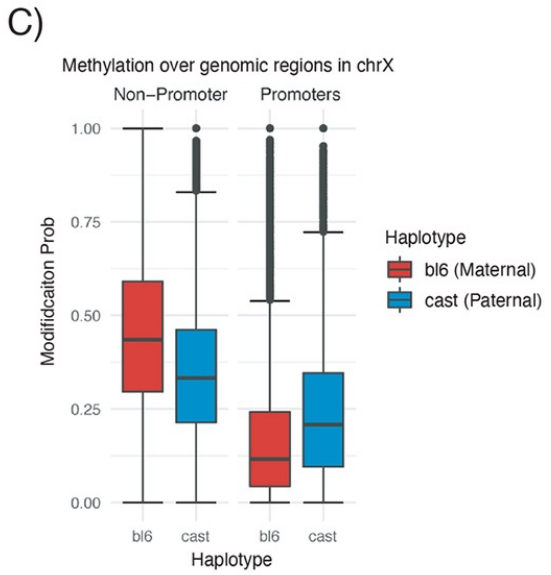
- provides conversion of data formats output by popular methylation callers into formats compatible with Bioconductor packages for DMR analysis

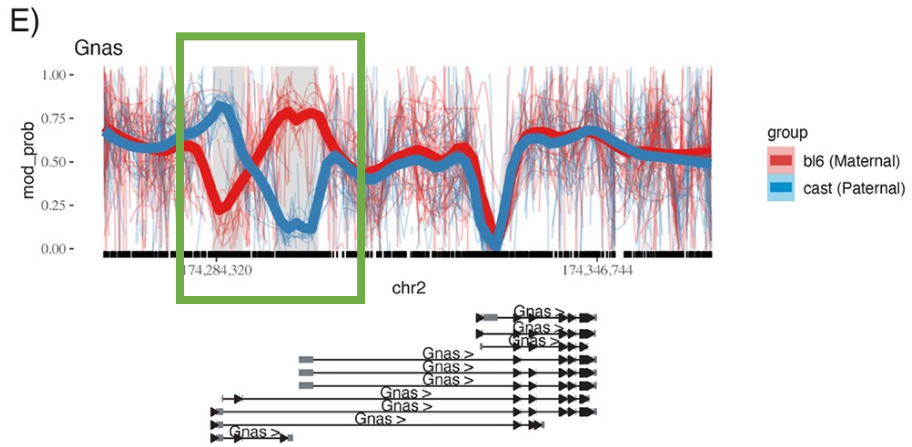


Example NanoMethViz plots

- dataset generated from triplicate female mouse placental tissues from F1 crosses between homozygous C57BL/6J mothers and CAST/EiJ fathers.







Su et al., 2021

Other DMR packages

- Methplotlib
  - Paper and GitHub

  - Paper and GitHub

Further reading

- Nanopore
  - Nanopore sequencing technology, bioinformatics and applications (Wang et al., 2021)
- Methylation tools
  - DNA methylation-calling tools for Oxford Nanopore sequencing: a survey and human epigenome-wide evaluation (Liu et al., 2021)
  - Systematic benchmarking of tools for CpG methylation detection from nanopore sequencing (Yuen et al., 2021)

Links and Resources

- Nanopore
  - “Epigenetics and methylation analysis” page on Nanopore website
- Megalodon
  - Documentation and GitHub
  - List of common arguments and modified base arguments
- IGV
  - User guide and downloads
  - Visualizing reads in bisulfite mode