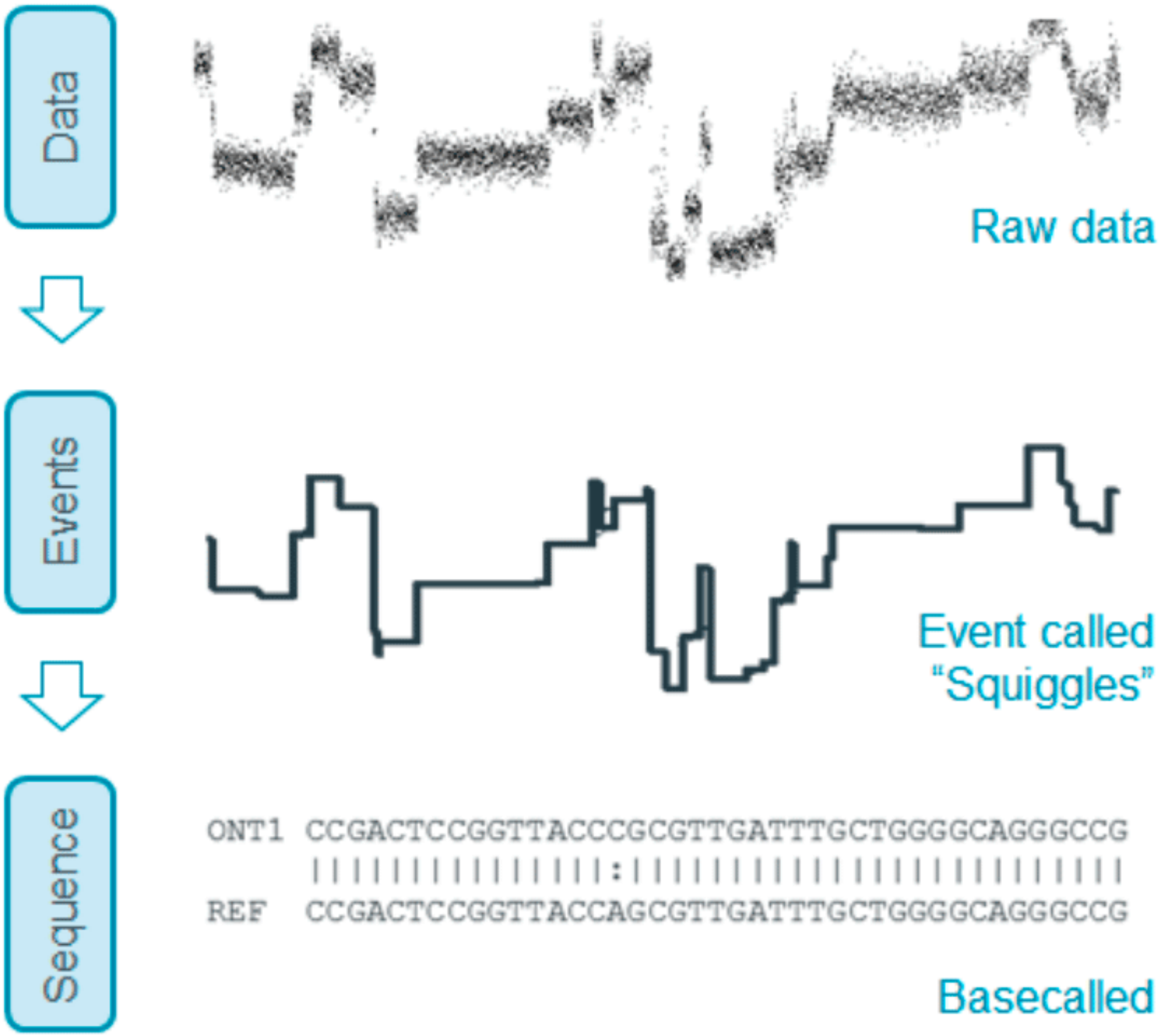
NANOPORE SEQUENCING BASECALLING

Wojciech Makałowski
Institute of Bioinformatics, University of Muenster, Germany
Department of Computational Biology, University of Tokyo, Japan
http://bioinformatics.uni-muenster.de

Data Structure

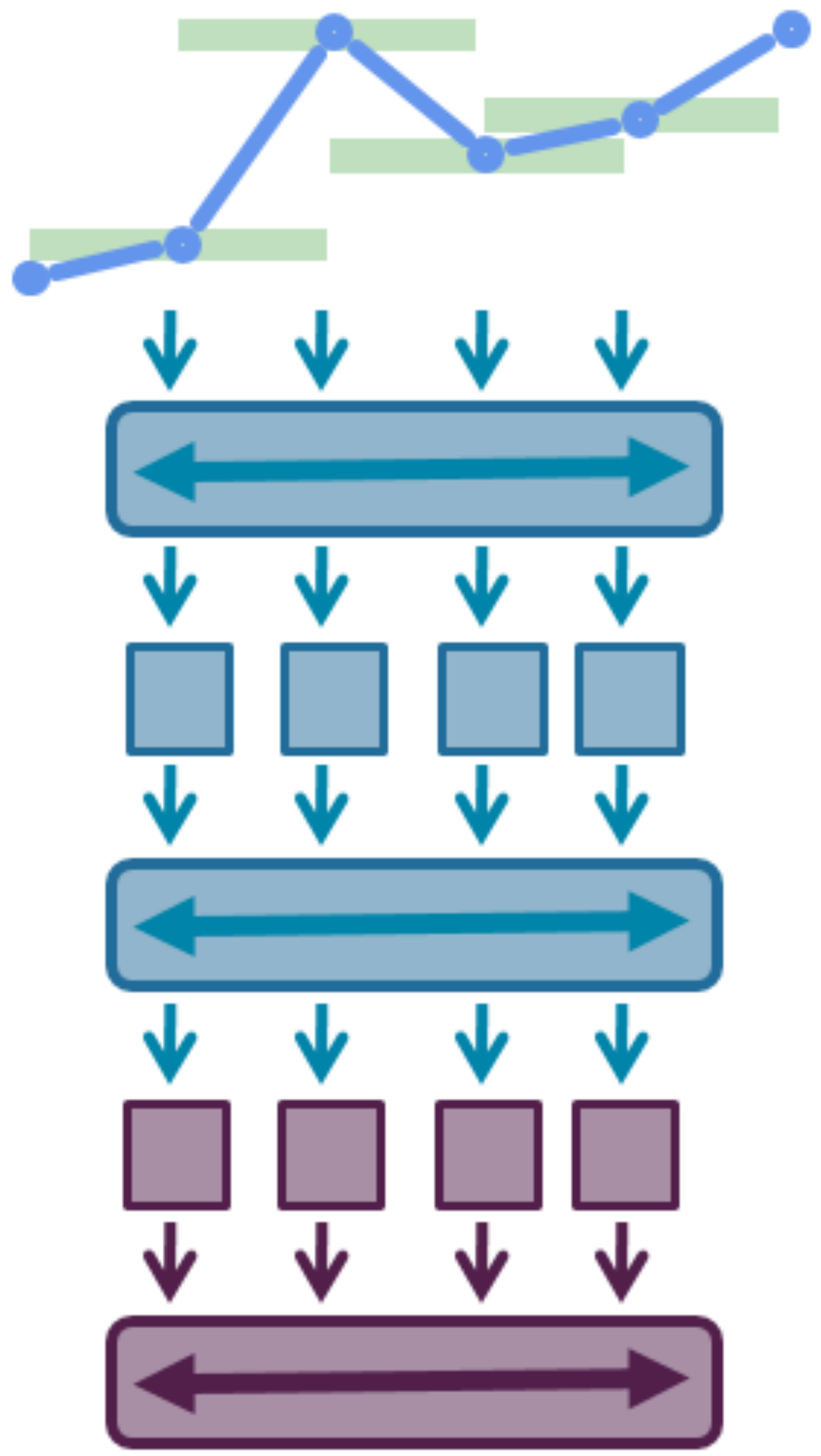


```
ONT1  CCGACTCCGGTTACCCGCGTTGATTTGCTGGGGCAGGGCCG
      |||||||||||||||:|||||||||||||||||||||||||
REF   CCGACTCCGGTTACCAGCGTTGATTTGCTGGGGCAGGGCCG
```

* Raw data - a direct measurement of the changes in ionic current as a DNA/RNA strand passes through the pore. These measurements are recorded by the MinKNOW software. MinKNOW also processes the signal into reads, each read corresponding to a single strand of DNA/RNA.

* Basecalling - the raw signal is processed into segments with information about current level, noise level and duration, which correspond to the movement of the DNA through the pore. These data are further processed by the basecalling algorithm to generate the base sequence of the read.

Basecalling - RNN



Distributions learned from squiggle training data

Bidirectional information flow (BLSTM layer)

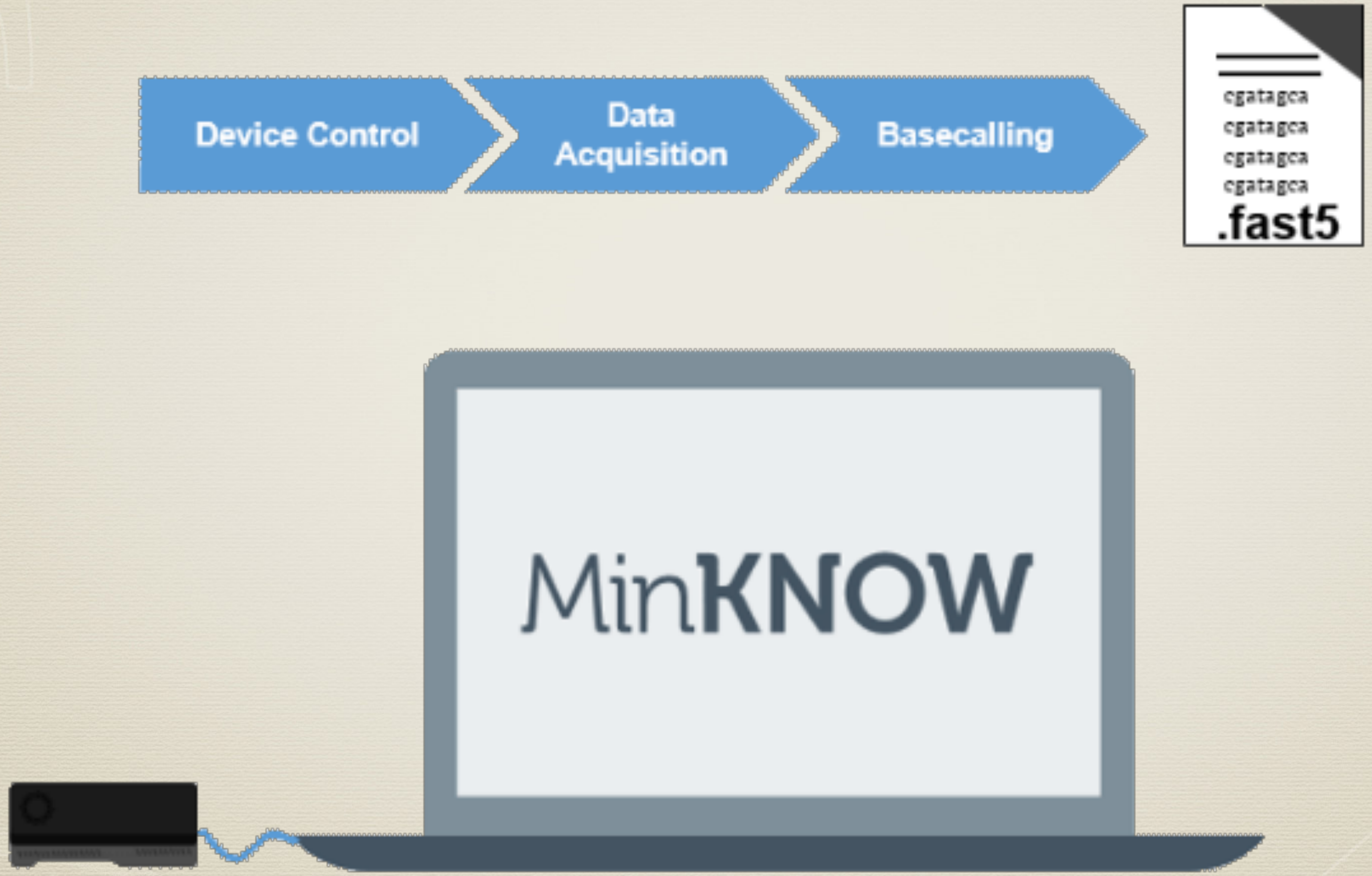
Processing layer

Bidirectional information flow (BLSTM layer)

Multi-base prediction

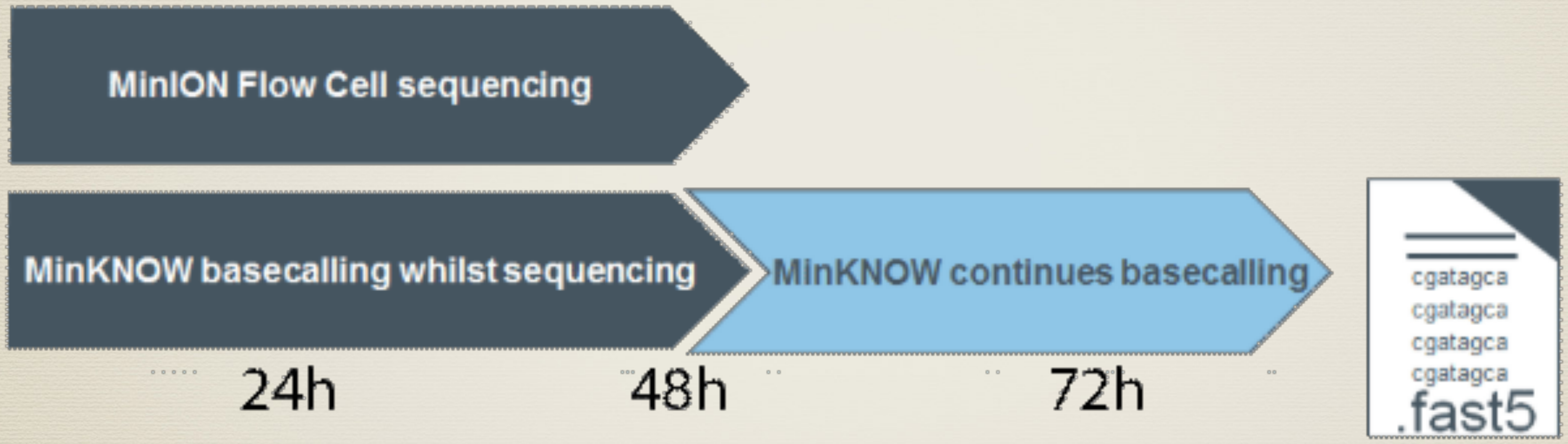
Decode to sequence

Live basecalling with MinKNOW



MinKNOW basecalling modes

Catch-up mode


MinION Flow Cell sequencing
MinKNOW basecalling whilst sequencing
MinKNOW continues basecalling
24h
48h
72h
cgatagca
cgatagca
cgatagca
cgatagca
.fast5

Keep-up mode - requires higher computational power usually not available with a laptop

Oxford Nanopore basecallers

Basecaller	Algorithm
MinKNOW basecaller	Production basecaller; uses a neural network. The algorithm is identical to the one used by Albacore, but may be a version behind
Albacore	Production basecaller; uses a neural network. Currently available as an executable, or as source code for members of the Developer group
Nanonet	Research basecaller; uses the latest research algorithm. Not actively supported
Scrappie	Research basecaller; uses a neural network with a 'transducer' model, which allows the basecaller to resolve long homopolymers. Not actively supported

Basecalling components

- Event detection
- Segmentation
- 1D basecalling

Read file location

* Windows
  * :\data\reads on the SSD
* Mac OS X
  * /Library/MinKNOW/data
* Linux
  * /var/lib/MinKNOW/data

File location and name

The reads are stored in data_folder/reads, where "data_folder" is the location that the user sets up during MinKNOW installation. This location does not change during autoupdates, but if the software is reinstalled, the installer will ask for the location.

The filename for a read follows a hardcoded pattern:

<output_reads_dir>/tmp/<batch_number>/
<data_set>_ch<channel_number>_read<read_number>.fast5.tmp

and if the partially completed read is not otherwise rejected, when it is completed:

<data_folder>/reads/<basecall_status>/<batch_number>/
<data_set>_ch<channel_number>_read<read_number>_<classification>.fast5

File content

```
/{attributes: file_version}
|-UniqueGlobalKey/
| |-tracking_id/{attributes: asic_id, asic_id_eeprom, asic_temp,
 device_id, exp_script_hash, exp_script_name, exp_script_purpose, exp_start_time,
 flow_cell_id, heatsink_temp, hostname, protocol_run_id, protocols_version_name,
 run_id, version, version_name}
| |-channel_id/{attributes: channel_number, digitisation, offset, range,
 sampling_rate}
| |-context_tags/{attributes: set when the experiment is configured}
|-Raw/
| |-Reads/
| |-Read_42/{attributes: start_time, duration, read_number, start_mux,
 read_id, median_before}
| |-Signal{samples}
|
```

Albacore

* Albacore is a software that provides an entry point to the Oxford Nanopore basecalling algorithms.
* It can be run from the command line on Windows and multiple unix-like platforms.
* A selection of configuration files allow basecalling DNA libraries made with our current range of sequencing kits and Flow Cells.

Albacore

System requirements

- 4 GB RAM plus 1 GB per worker thread for 1D basecalling
- 4 GB RAM plus 2 GB per worker thread for 1D2 and 2D basecalling
- Administrator access for installation
- ~100 Mb of drive space for installation, minimum 512 GB storage space for basecalled read files (1 TB recommended)
- When starting with a .fast5 file that only has raw data in it, the file size will increase approx. 5 times

Albacore

* read_fast5_basecaller.py
    * main script written in python
    * an entry gate to ONT basecalling

read_fast5_basecaller.py -l

Flow cell	Seq kit
FLO-MIN106	SQK-LSK2
FLO-MIN106	SQK-NSK0
FLO-MIN106	SQK-RAD0
FLO-MIN106	SQK-RLI0
FLO-MIN106	SQK-LWP0
FLO-MIN106	SQK-RAS2
FLO-MIN106	SQK-LSK1
FLO-MIN106	VSK-VBK0
FLO-MIN106	SQK-RBK0
FLO-MIN106	SQK-RLB0
FLO-MIN106	SQK-LWB0
FLO-MIN106	SQK-RAB2
FLO-MIN106	SQK-RNA0

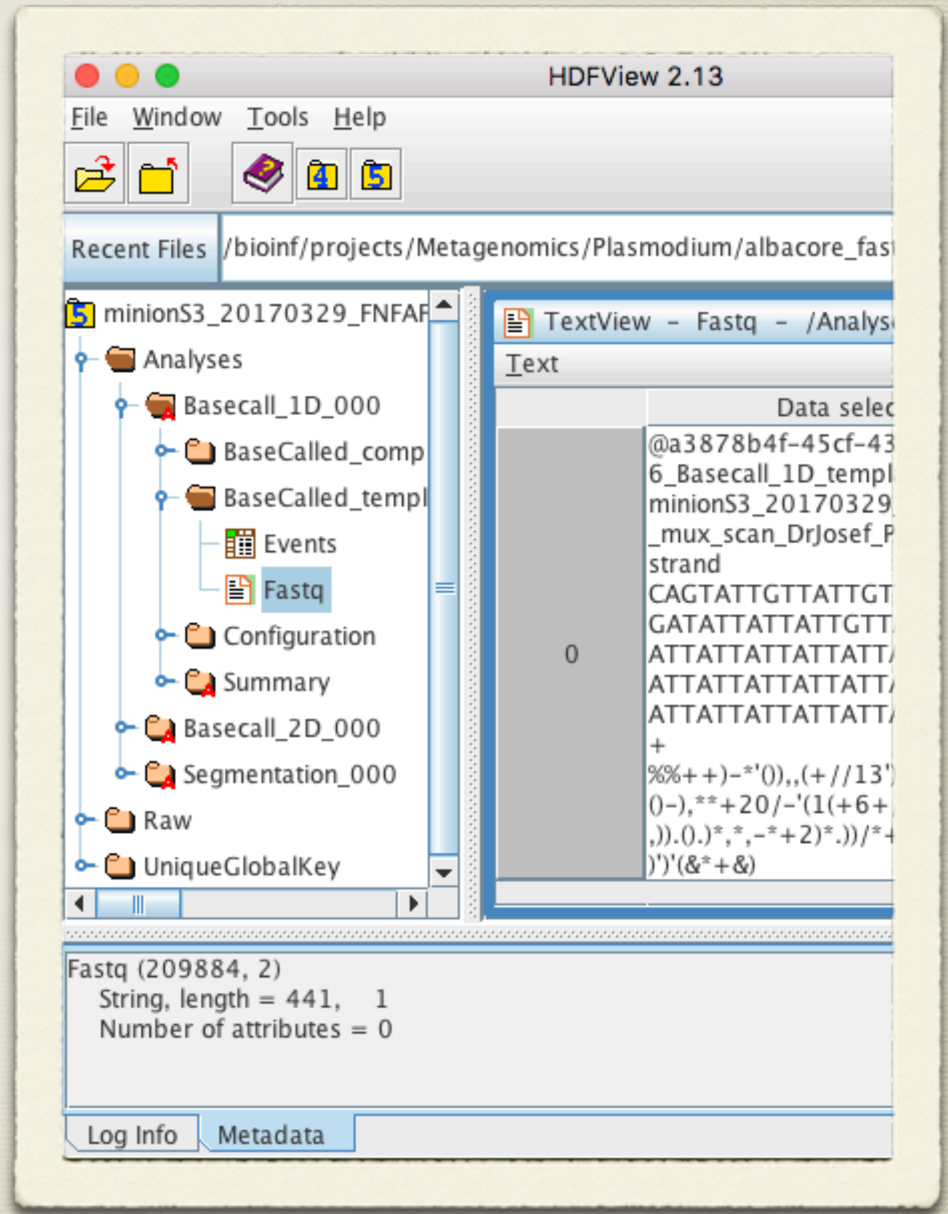
Flow cell	Seq kit
FLO-MIN107	SQK-RNA0
FLO-MIN107	SQK-RAD0
FLO-MIN107	SQK-RAD0
FLO-MIN107	SQK-RLI0
FLO-MIN107	SQK-LWP0
FLO-MIN107	SQK-RAS2
FLO-MIN107	SQK-LSK1
FLO-MIN107	VSK-VBK0
FLO-MIN107	SQK-DCS1
FLO-MIN107	SQK-PCS1
FLO-MIN107	SQK-RBK0
FLO-MIN107	SQK-RLB0
FLO-MIN107	SQK-LWB0
FLO-MIN107	SQK-RAB2
FLO-MIN107	SQK-LSK3

Basecalling example

```
read_fast5_basecaller.py -i 20170519_1639_170519_run -t 2
-s 20170519_1639_170519_run -o fastq,fast5 -r
-f FLO-MIN106 -k SQK-LSK208
```

MinION data formats FAST5

- The raw data is stored as binary files in HDF5 standard
- HDFView allows quick look at the raw data files
- https://support.hdfgroup.org/products/java/release/download.html#bin



/bioinf/projects/Metagenomics/Plasmodium/albacore_fas

```
minionS3_20170329_FNFAF
  Analyses
    Basecall_1D_000
      BaseCalled_comp
      BaseCalled_templ
        Events
        Fastq
      Configuration
      Summary
    Basecall_2D_000
    Segmentation_000
  Raw
  UniqueGlobalKey
```

```
0   @a3878b4f-45cf-43
    6_Basecall_1D_templ
    minionS3_20170329
    _mux_scan_DrJosef_F
    strand
    CAGTATTGTTATTGT
    GATATTATTATTGTT
    ATTATTATTATTATT
    ATTATTATTATTATT
    ATTATTATTATTATT
    +
    %%++)-*'()),,(+//13')
    ()-),**+20/-'(1(+6+
    ,)).().)*,*,-*+2)*.))/*+
    )')'(&*+&)
```

```
Fastq (209884, 2)
   String, length = 441,    1
   Number of attributes = 0
```

MinION data formats
FASTQ

```
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345 length=36
GGGTGATGGCCGCTGCCGATGGCGTCAAATCCCACC
+
IIIIIIIIIIIIIIIIIIIIIIIIIIIIII9IG9IC
```

Line 1 begins with a '@' character and is followed by a sequence identifier and an optional description

Line 2 is the raw sequence letters.

Line 3 begins with a '+' character and is optionally followed by the same sequence identifier (and any description) again.

Line 4 encodes the quality values for the sequence in Line 2, and must contain the same number of symbols as letters in the sequence.

MinION data formats FASTQ

$$Q = -10 \log_{10} p$$

p = probability that the corresponding base call is incorrect

ASCII	p	Q
!	1	0
)	0.1	10
3	0.01	20
=	0.001	30
H	0.0001	40
~		93

!"#$%&'()   *+,-./0123   456789:;<=   >?@ABCDEFGH   I

MinION data formats
FASTA

Very simple format but it may contain quite a bit in formation on the sequence.
Used by many software including BLAST and NanoPipe

```
>SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345 length=36
GGGTGATGGCCGCTGCCGATGGCGTCAAATCCCACC
```