# poretools Documentation

***Release 0.5.0***

**Nick Loman and Aaron Quinlan**

October 16, 2014

# Contents

The MinION (TM) from Oxford Nanopore Technologies (ONT) is the first nanopore sequencer to be commercialised and is now available to early-access users. The MinION (TM) is a USB-connected, portable nanopore sequencer which permits real-time analysis of streaming event data. Currently, the research community lacks a standardized toolkit for the analysis of nanopore datasets.

We have therefore develped `poretools`, a flexible toolkit for exploring datasets generated by nanopore sequencing devices from MinION for the purposes of quality control and downstream analysis. `Poretools` operates directly on the native FAST5 (a variant of the HDF5 standard) file format produced by ONT and provides a wealth of format conversion utilities and data exploration and visualization tools.

A preprint of the `poretools` manuscript is available on bioarxiv: http://biorxiv.org/content/early/2014/07/23/007401

Below are a few examples of common usage.

1. Extract sequences in FASTQ format from a set of FAST5 files.

```
poretools fastq fast5/
```

2. Make a collector's curve of the yield from a sequencing run.

```
poretools yield_plot --plot-type reads fast5/
```

3. Plot a histogram of read sizes from a set of FAST5 files.

```
poretools hist fast5/
```

CHAPTER 1

# Table of contents

## 1.1 Installation

### 1.1.1 Basic Installation

```
git clone https://github.com/arq5x/poretools
cd poretools
```

Install as root:

```
python setup.py install
```

Install as a plain old user who has root access:

```
sudo python setup.py install
```

Install as a plain old who lacks `sudo` priveleges:

```
# details: https://docs.python.org/2/install/index.html#alternate-installation-the-user-scheme
python setup.py install --user

# now update your PATH such that it includes the directory to which poretools was just copied.
# look for a line in the installation log like: Installing poretools script to /home/arq5x/.local/bi
# in this case, I would either add that path to the PATH environment variable for the current sessio
export PATH=$PATH:/home/arq5x/.local/bin

# or, better yet add it to your .bashrc file.
# at this point you should be able to run the poretools executable from anywhere on your system.
poretools --help
```

### 1.1.2 Installing on Windows with MinKNOW installed

MinKNOW installs the Anaconda distribution of Python, which means that h5py is already installed.

However, currently MinKNOW does not update the Windows registry to specify that Anaconda is the default version of Python, which makes installing packages tricky. To address this, some changes need to be made to the registry. This can be fixed by downloading the following file:

> <https://raw.githubusercontent.com/arq5x/poretools/master/dist/poretools.reg>

Ensure it is named 'poretools.reg' and then run it (by double-clicking). Windows will prompt you about making changes to the registry, which you should agree to.

The only additional dependency that is required is rpy2 and R.

Download rpy2 from the pre-built binary page at: <http://www.lfd.uci.edu/~gohlke/pythonlibs/>. You want the version for Python 2.7 on 64-bit Windows. Run the installer.

Then, to install poretools, simply download and run the Windows installer:

> <https://github.com/arq5x/poretools/blob/master/dist/poretools-0.3.1.win-amd64.exe?raw=true>

### 1.1.3 Plotting with R on Windows

If you wish to use the R plots (experimental, on Windows) you also need to:

Download R for Windows from: <http://cran.r-project.org/bin/windows/base/>

Run the installer, then start up R and install ggplot2:

```
install.packages("ggplot2")
```

You need to set two environment variables to run poretools currently:

```
set R_HOME=c:\Program Files\R\R-3.1.1
set R_USER=c:\Users\MY USER\Documents
```

You may also need to add the following directory to your PATH:

```
C:\Program Files\R\R-3.1.1\bin\x64
```

Instructions for updating your PATH on Windows can be found here: http://geekswithblogs.net/renso/archive/2009/10/21/how-to-set-the-windows-path-in-windows-7.aspx

### 1.1.4 Installing on OS X

First, you should install a proper package manager for OS X. In our experience, HomeBrew works extremely well.

To install HomeBrew, you run the following command (lifted from the HomeBrew site):

```
ruby -e "$(curl -fsSL https://raw.github.com/Homebrew/homebrew/go/install)"
```

Using HomeBrew, install HDF5 from the HomeBrew Science "tap";

```
brew tap homebrew/science
brew install hdf5
```

Now, you will need to install the R statistical analysis software (you may already have this...). The CRAN website houses automatic installation packages for different versions of OS X. Here are links to such packages for Snow Leopard and higher as well as Mavericks.

At this point, you can install poretools.

```
git clone https://github.com/arq5x/poretools
cd poretools
```

Install as an administrator of your machine:

```
sudo python setup.py install
```

Install as a plain old who lacks `sudo` priveleges:

```
# details: https://docs.python.org/2/install/index.html#alternate-installation-the-user-scheme
python setup.py install --user
```

### 1.1.5 Installing dependencies on Ubuntu

Package dependencies

```
sudo apt-get install git python-setuptools python-dev cython libhdf5-serial-dev
```

Then install R 3.0, this requires a bit of hacking. You need to replace 'precise' with the appropriate version if you are on a different Ubuntu version, see <http://cran.r-project.org/bin/linux/ubuntu/README> for more details.

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys E084DAB9
```

Open in a text editor (as sudo) the file `/etc/apt/sources.list` and add the following line to the bottom, for Ubuntu 12.04:

```
deb http://www.stats.bris.ac.uk/R/bin/linux/ubuntu precise/
```

Or, for Ubuntu 14.04:

```
deb http://www.stats.bris.ac.uk/R/bin/linux/ubuntu trusty/
```

Then, run the following commands to install R 3.0:

```
sudo apt-get update
sudo apt-get install r-base python-rpy2
```

Start R

```
R
```

Then run the following commands within the R programme, and follow any prompts:

```
options("repos" = c(CRAN = "http://cran.rstudio.com/"))
install.packages("codetools")
install.packages("MASS")
install.packages("ggplot2")
```

Then install poretools, finally:

```
git clone https://github.com/arq5x/poretools
cd poretools
sudo python setup.py install
poretools
```

### 1.1.6 In the cloud

Amazon Web Services machine image ID: ami-4c0ec424

### 1.1.7 Via docker

Build the docker container yourself (preferred):

Or use the pre-built image from Docker Hub:

```
docker pull stephenturner/poretools
docker run stephenturner/poretools --help
```

To run the poretools container on data residing on the host machine, run `docker run -h` and look at the help for the `-v` option.

## 1.2 Options

The following demonstrates the options available in `poretools`.

```
poretools --help
usage: poretools [-h] [-v]

                 {combine,fastq,fasta,stats,hist,events,readstats,tabular,nucdist,qualdist,winner,wig
                 ...

optional arguments:
  -h, --help            show this help message and exit
  -v, --version         Installed poretools version

[sub-commands]:
  {combine,fastq,fasta,stats,hist,events,readstats,tabular,nucdist,qualdist,winner,wiggle,times}
    combine             Combine a set of FAST5 files in a TAR achive
    fastq               Extract FASTQ sequences from a set of FAST5 files
    fasta               Extract FASTA sequences from a set of FAST5 files
    stats               Get read size stats for a set of FAST5 files
    hist                Plot read size histogram for a set of FAST5 files
    events              Extract each nanopore event for each read
    readstats           Extract signal information for each read over time.
    tabular             Extract the lengths and name/seq/quals from a set of
                        FAST5 files in TAB delimited format
    nucdist             Get the nucl. composition of a set of FAST5 files
    qualdist            Get the qual score composition of a set of FAST5 files
    winner              Get the longest read from a set of FAST5 files
    squiggle            Plot the observed signals for FAST5 reads
    times               Return the start times from a set of FAST5 files in
                        tabular format
    yield_plot          Plot the yield over time for a set of FAST5 files
```

## 1.3 IPython Notebook

An IPython notebook demonstrating the functionality and output of `poretools` is available in the repository. Use this link to view it via the `nbviewer` service: <http://nbviewer.ipython.org/github/arq5x/poretools/blob/master/poretools/ipynb/test_run_report.ipynb>

## 1.4 Usage examples

### 1.4.1 poretools `fastq`

Extract sequences in FASTQ format from a set of FAST5 files.

```
poretools fastq fast5/*.fast5
```

Or, if there are too many files for your OS to do the wildcard expansion, just provide a directory. `poreutils` will automatically find all of the FAST5 files in the directory.

```
poretools fastq fast5/
```

Extract sequences in FASTQ format from a set of FAST5 files.

```
poretools fastq fast5/
poretools fastq --min-length 5000 fast5/
poretools fastq --type all fast5/
poretools fastq --type fwd fast5/
poretools fastq --type rev fast5/
poretools fastq --type 2D fast5/
poretools fastq --type fwd,rev fast5/
```

Only extract sequence with more complement events than template. These are the so-called "high quality 2D reads" and are the most accurate sequences from a given run.

```
poretools fastq --type 2D --high-quality fast5/
```

### 1.4.2 poretools `fasta`

Extract sequences in FASTA format from a set of FAST5 files.

```
poretools fasta fast5/
poretools fasta --min-length 5000 fast5/
poretools fasta --type all fast5/
poretools fasta --type fwd fast5/
poretools fasta --type rev fast5/
poretools fasta --type 2D fast5/
poretools fasta --type fwd,rev fast5/
```

### 1.4.3 poretools `combine`

Create a tarball from a set of FAST5 (HDF5) files.

```
# plain tar (recommended for speed)
poretools combine -o foo.fast5.tar fast5/*.fast5

# gzip
poretools combine -o foo.fast5.tar.gz fast5/*.fast5

# bzip2
poretools combine -o foo.fast5.tar.bz2 fast5/*.fast5
```

### 1.4.4 poretools `yield_plot`

Create a collector's curve reflecting the sequencing yield over time for a set of reads. There are two types of plots. The first is the yield of reads over time:

```
poretools yield_plot --plot-type reads fast5/
```

The result should look something like:

The second is the yield of base pairs over time:

```
poretools yield_plot --plot-type basepairs fast5/
```

The result should look something like:

Of course, you can save to PDF or PNG with *–saveas*:

```
poretools yield_plot \
          --plot-type basepairs \
          --saveas foo.pdf\
          fast5/

poretools yield_plot \
          --plot-type basepairs \
          --saveas foo.png\
          fast5/
```

If you don't like the default aesthetics, try *–theme-bw*:

```
poretools yield_plot --theme-bw fast5/
```

### 1.4.5 poretools `squiggle`

Make a "squiggle" plot of the signal over time for a given read or set of reads

```
poretools squiggle fast5/foo.fast5
```

The result should look something like:

If you don't like the default aesthetics, try *–theme-bw*:

```
poretools squiggle --theme-bw fast5/
```

Other options:

```
# save as PNG
poretools squiggle --saveas png fast5/foo.fast5

# save as PDF
poretools squiggle --saveas pdf fast5/foo.fast5

# make a PNG for each FAST5 file in a directory
poretools squiggle --saveas png fast5/
```

### 1.4.6 poretools `winner`

Report the longest read among a set of FAST5 files.

```
poretools winner fast5/
poretools winner --type all fast5/
poretools winner --type fwd fast5/
poretools winner --type rev fast5/
poretools winner --type 2D fast5/
poretools winner --type fwd,rev fast5/
```

### 1.4.7 poretools `stats`

Collect read size statistics from a set of FAST5 files.

```
poretools stats fast5/
total reads 2286.000000
total base pairs    8983574.000000
mean    3929.822397
median  4011.500000
min 13.000000
max 6864.000000
```

### 1.4.8 poretools `hist`

Plot a histogram of read sizes from a set of FAST5 files.

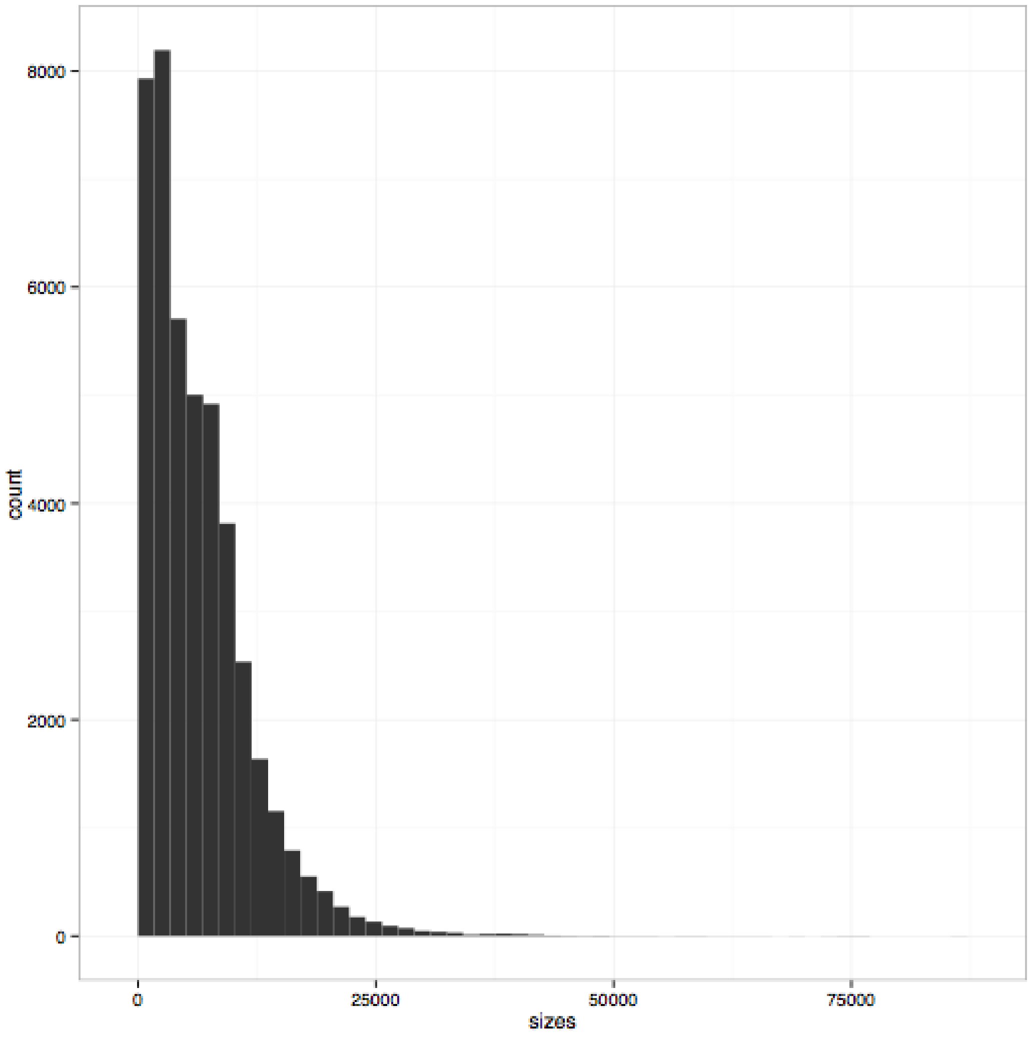
```
poretools hist fast5/
poretools hist --min-length 1000 --max-length 10000 fast5/

poretools hist --num-bins 20 --max-length 10000 fast5/
```

If you don’t like the default aesthetics, try *–theme-bw*:

```
poretools hist --theme-bw fast5/
```

The result should look something like:



### 1.4.9 poretools `nucdist`

Look at the nucleotide composition of a set of FAST5 files.

```
poretools nucdist fast5/
A   78287   335291  0.233489714904
C   75270   335291  0.224491561062
T   92575   335291  0.276103444471
G   84754   335291  0.252777438106
N   4405    335291  0.0131378414571
```

### 1.4.10 poretools `qualdist`

Look at the quality score composition of a set of FAST5 files.

```
poretools qualdist fast5/
!   0   83403   335291  0.248748102395
"   1   46151   335291  0.137644613187
#   2   47463   335291  0.141557632027
```

```
$   3   34471   335291  0.102809201559
%   4   24879   335291  0.0742012162569
&   5   20454   335291  0.0610037251224
'   6   16783   335291  0.0500550268274
(   7   13699   335291  0.0408570465655
)   8   11356   335291  0.0338690868529
*   9   9077    335291  0.0270720061081
+   10  6492    335291  0.0193622852984
,   11  4891    335291  0.014587328619
-   12  3643    335291  0.0108651887465
.   13  2585    335291  0.00770972080968
/   14  1969    335291  0.0058725107444
0   15  1475    335291  0.00439916371152
1   16  1146    335291  0.00341792651756
2   17  902 335291  0.00269020045274
3   18  790 335291  0.00235616225905
4   19  619 335291  0.0018461575169
5   20  532 335291  0.00158668142002
6   21  440 335291  0.00131229290378
7   22  397 335291  0.00118404609727
8   23  379 335291  0.00113036138757
9   24  313 335291  0.000933517452004
:   25  327 335291  0.000975272226215
;   26  138 335291  0.000411582774366
<   27  121 335291  0.000360880548538
=   28  96  335291  0.000286318451733
>   29  76  335291  0.000226668774289
?   30  69  335291  0.000205791387183
@   31  61  335291  0.000181931516205
A   32  48  335291  0.000143159225866
B   33  23  335291  6.8597129061e-05
C   34  14  335291  4.17547742111e-05
D   35  6   335291  1.78949032333e-05
F   37  3   335291  8.94745161666e-06
```

### 1.4.11 poretools `tabular`

Dump the length, name, seq, and qual of the sequence in one or a set of FAST5 files.

```
poretools tabular foo.fast5
length  name    sequence    quals
10    @channel_100_read_14_complement   GTCCCCAACAACAC    $%%'"$"%!)
```

### 1.4.12 poretools `events`

Extract the raw nanopore events from each FAST5 file.

```
poretools events burn-in-run-2 | head -5
file    strand  mean    start   stdv    length  model_state model_level move    p_model_state   mp_mo
burn-in-run-2/ch100_file15_strand.fast5  template    56.4648513559   6595.744    1.62598948551   0.02
burn-in-run-2/ch100_file15_strand.fast5  template    53.2614042745   6595.77 1.12361695715   0.0262
burn-in-run-2/ch100_file15_strand.fast5  template    51.0001271042   6595.7962   1.07380437991   0.14
burn-in-run-2/ch100_file15_strand.fast5  template    49.6976788934   6595.9384   1.03634357984   0.03
burn-in-run-2/ch100_file15_strand.fast5  template    51.7633085659   6595.9748   1.04743182078   0.04
```

### 1.4.13 poretools `times`

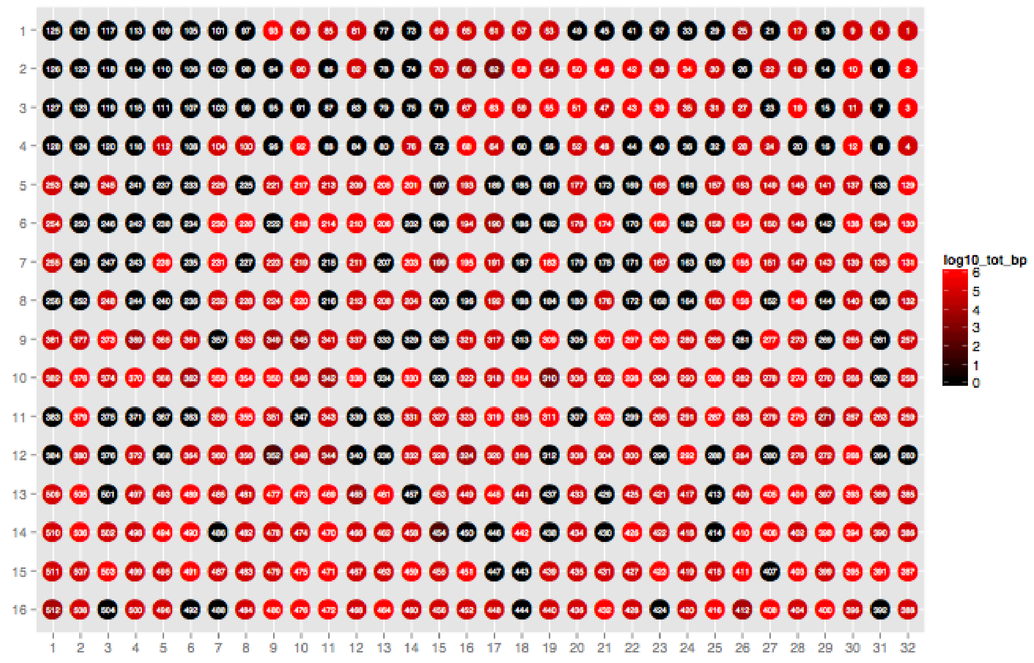
Extract the start time of each detected molecule into tabular format.

### 1.4.14 poretools `occupancy`

Plot the throughput performance of each pore on the flowcell during a given sequencing run.

```
poretools occupancy fast5/
```

The result should look something like:



# Requirements

- HDF5 >= 1.8.7 (http://www.hdfgroup.org/HDF5/)
- R >= 3.0.0
- Python >= 2.7
- rpy2 >= 2.4.2
- h5py >= 2.0.0

**Note:** Please note that Anaconda and Python(x,y) already have all these dependencies installed, other than R/Rpy2: Anaconda (Linux, Windows, OS X): https://store.continuum.io/cshop/anaconda/ Python(x,y) (Windows): https://code.google.com/p/pythonxy/