Debarcer: De-Barcoding and Error Correction

Paul Krzyzanowski
Ontario Institute for Cancer Research
Toronto, Ontario, Canada
paul.krzyzanowski@oicr.on.ca

June 24, 2016

1 Location

You can download the latest version of Debarcer from GitHub:

```
https://github.com/oicr-gsi/debarcer/releases
```

Once the archive has been extracted to a suitable directory, you will need to copy the `config_files/debarcer.dotfile` to `~/.debarcer` as this configuration file is used throughout the pipeline.

The directory containing runDebarcer.sh must be in your PATH

2 Using Debarcer

2.1 Requirements

- Ensure you have 16G memory available
- Note that a bam file for the current alignment will be generated in your work directory
- Currently, you need to have access to 'qsub' through SGE or OGS

2.2 Running the package

Once `runDebarcer.sh` is in your `$PATH`, either manually or using modules, you can run it as follows:

```
$ runDebarcer.sh -r -f [FASTQFILE] -n [SAMPLENAME] -o [OUTPUTDIR]
```

Using modules is handy for running via qsub:

```
qsub -N "Debarcer" -b y -cwd -l h_vmem=16g "module load debarcer/dev; \
runDebarcer.sh -r -f [FASTQFILE] -n [SAMPLENAME] -o [OUTPUTDIR]"
```

There is a small test set of data included with the distribution:

```
runDebarcer.sh -r -f [DebarcerRoot]/demodata/Sample_Test.R1.fastq.gz \
-n Sample_Test -o ./testresults
```

This test data was extracted from a sample that only contained the TP146 amplicon (Specifically, `Sample 9297` from an internal dataset, to be posted somewhere in the future). This file only contains reads which were from UID families of depth 20-30, inclusive.

2.2.1 Timings

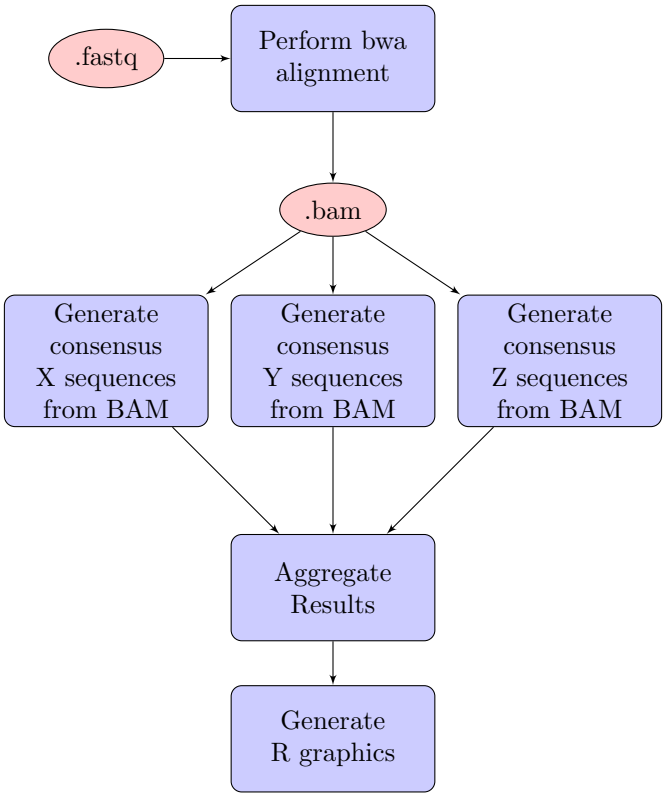
- The included test file should run in under 5 minutes.
- 2 million mapped reads: Under 15 minutes.
- 20 million mapped reads: 200 minutes (3h)

2.3 Automating Analyses

To help automate this whole process, the
`populateWorkDirectory.pl` script in the debarcer 'src' directory can create a results hierarchy based on a directory of fastq files and write the appropriate run scripts:

```
mkdir -p ResultsRoot\fastqs
cd ResultsRoot\fastqs
find [location of the the fastq files] -name "*fastq.gz" -exec ln -s {} \;
cd ..
perl $DEBARCERHOME\src\populateWorkDirectory.pl ./fastqs
```

3 Workflow



4 Output

The output contains several different types of files.

Package results files with a tar command that wraps up the files found by:

```
find . -name *.pdf -o -name "*bamPositionComposition*" -o \
-name "*UID*" -o -name "*SummaryStatistics.txt" -o -name "*log"
```

i.e.

```
tar cvfz results_file_name.tar.gz `[find command goes in backticks]`
```

This can also be done using the `packageResults.sh` file in the Debarcer `tools` subdirectory. Simply execute that script from the root of your results hierarchy.

4.1 Revisions

- 0.3.0: April 7, 2016.
- 0.2.0: July 27, 2015.

5 Miscellany

Requirements: Consensus calling script needs at least 16G memory.

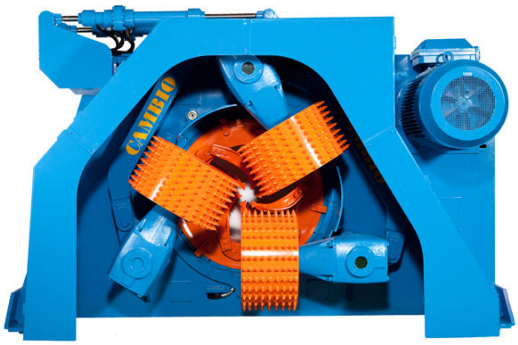


Figure 1: An actual debarker.

Figure 1 shows a tree debarker, which is similar to a giant pencil sharpener. Debarkers are used to remove the rough surface layers of a tree, which usually exhibit errors generated in the tree growth process $nicks, scratches, breaks, and other injuries$. Simiarly to PCR, errors generated early in tree growth have a bigger effect at the endpoint (tree cutting), some of which debarkers can't repair.