

Using TagDust in the CASAVA pipeline

by Timo Lassmann

January 6, 2019

Copyright ©2013 - 2019 Timo Lassmann (timolassmann@gmail.com)

This document is free; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This document is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with TagDust.

If not, see (<http://www.gnu.org/licenses/>).

Introduction

Replacing the default CASAVA de-multiplexing with TagDust is trivial. Simply follow the steps below:

1. Install bcl2fastq ¹
2. Create a sample sheet to turn off the default multi-plexing:

```
FCID,Lane,SampleID,SampleRef,Index,Description,Control,Recipe,Operator,SampleProject
Noname,1,not_demultiplexed,,,N,,BCF,tagdust
Noname,2,not_demultiplexed,,,N,,BCF,tagdust
```

Make sure that the second column matches the actual lanes present.

3. run configureBclToFastq.pl with the sample sheet generated above to configure CASAVA:

```
bash-3.1$ bin/configureBclToFastq.pl --sample-sheet samplesheet.csv
--input-dir <>
--output-dir <> --use-bases-mask 'y76n*,y6n,y76n*'
```

It is important to set the length of the reads and barcode accurately. In the example above we expect a 76t paired end data indexed by a 6nt barcode. In case of a 8nt barcode we would use: *-use-bases-mask 'y76n*,y8n,y76n*'*.

Have a look at the CASAVA documentation for more information on this topic.

4. create a TagDust architecture file indicating which indices were used. E.g:

```
bash-3.2$ more casava_6nt_arch.txt
tagdust -1 B:ATCACG,CGATGT,TTAGGC,TGACCA,ACAGTG,GCCAAT,
CAGATC,ACTTGA,GATCAG,TAGCTT,GGCTAC,CTTGTA
tagdust -1 R:N
```

The first line tells TagDust2 to look for reads composed entirely of one of the 12, 6nt long index sequences. This line has to be changed to reflect the actual indices used in a particular experiment. It is also possible to add additional lines and let TagDust2 figure out which combination of indices was used. The second architecture is used for both the other actual reads.

5. run TagDust on the output files generated by CASAVA:

```
bash-3.1$ tagdust -arch casava_6nt_arch.txt \
not_demultiplexed_NoIndex_L001_R1_001.fastq.gz \
not_demultiplexed_NoIndex_L001_R2_001.fastq.gz \
not_demultiplexed_NoIndex_L001_R3_001.fastq.gz \
-o demultiplexed
```

¹<http://support.illumina.com/downloads/bcl2fastq-conversion-software.184.html>