CytoMeth

CytoMeth tool compiles a set of open source software named in the Roche pipeline guidelines to perform SeqCap Epi data analysis. The pipe includes read quality assessment, read filtering, mapping to a reference genome, removal of PCR duplicates, assessment of coverage statistics, analyse methylation and variant calling and filtering as well as some additional functionalities added to improve the process and facilitate obtaining the processed results. Here, to obtain methylomes for brain tumor samples we used SeqCap Epi CpGiant Methylation panel and performed bisulphite conversion followed by Illumina NGS sequencing and CytoMeth tool analysis.

Table of contents

Installation

CytoMeth is implemented as a set of R scripts that run various tools in a specific sequence with specific set of parameters. It can be installed (and used) in two ways:

- as a set of scripts and third party required tools installed directly in your Linux OS
- as a docker image that can be run under any OS

If you prefer the docker installation please skip the section below and go to the section The Docker. If you prefer to install it directly on your Linux environment please go through all below steps.

Environment Preparation

Notice that below steps refer to Linux OS. However more experienced user can also use this manual to install CytoMeth on OSX but it requires some slight adaptations that are not provided in the description.

To complete the installation process CytoMeth requires the following components installed on your OS:

- R and Rscript
- Conda - an open source package management system
- Python 2.x
- Java 8 (1.8) or above
- wget tool

If you are sure all of the above is working correctly (*R*, *conda*, *Java*, *python 2.x*) on your system you can skip the next section and go to the section Installation of CytoMeth Components.

Hardware requirements

Hardware requirements highly depend on type and size of the input data. However for human data size of the memory should be at least 4GB reserved for the CytoMeth. Tools that CytoMeth runs also take the

advantage of multicore CPU architecture.

Recommended OS packages

Make sure that your Linux OS does not lack any of the following system packages:

```
sudo apt-get update && apt-get install -y --install-recommends
locales \
wget \
zip \
unzip \
curl \
libcurl4-openssl-dev \
libssl-dev \
libssl1.1 \
libncurses5-dev
```

R and Rscript

Check if there is R installed on your machine. Type in a terminal window:

```
R --version
Rscript --version
```

If you dont have R or Rscript please install them. If missing R:

```
sudo apt install r-base
sudo apt install r-base-dev
```

If still missing Rscript try to install:

```
sudo apt-get install littler
```

Conda

Check if there is conda installed on your machine:

```
conda info
```

If conda *command not found* please install it. Download anaconda from the web:

```
wget https://repo.anaconda.com/archive/Anaconda3-2020.11-Linux-x86_64.sh
```

Install it in the following directory: '*/opt/anaconda*' and remove the installation file.

```
sudo bash Anaconda3-2020.11-Linux-x86_64.sh -b -p /opt/anaconda
rm Anaconda3-2020.11-Linux-x86_64.sh
```

Create new users group 'anaconda' and give all required priviliges to that group.

```
sudo groupadd anaconda
sudo chgrp -R anaconda /opt/anaconda
sudo chmod 770 -R /opt/anaconda
```

Add all CytoMeth users to the anaconda group. Please replace *_user_* with your username. These users will have an access to all tools installed by conda.

```
sudo adduser _user_ anaconda
```

All CytoMeth users need to add path to anaconda to the *PATH* system variable in '*.bashrc*' file:

```
echo 'export PATH="/opt/anaconda/bin:$PATH"' >> ~/.bashrc
source ~/.bashrc
```

Now, conda should be available from a terminal window. Open the new one and type again:

```
conda info
```

You should see all information about conda environment.

Python 2.x

Check your python version. It is recommended to use python 2.x with CytoMeth.

```
python2 --version
```

If your OS lacks of python 2.x and pip please install it with the following commnads:

```
sudo apt-get update
sudo apt-get install python2
sudo curl https://bootstrap.pypa.io/pip/2.7/get-pip.py --output get-pip.py
sudo python2 get-pip.py
```

Java

Check Java version. It is recommended to use Java >=8 (1.8) with CytoMeth.

```
java -version
```

If your OS lacks of Java please install it with the following commnads:

```
sudo apt-get update
sudo apt-get install openjdk-8-jre-headless
```

Installation of CytoMeth Components

To get CytoMeth from the github repository you may download it as a zip file or clone the project:

```
git clone https://github.com/mdraminski/CytoMeth.git
cd CytoMeth
```

Installation script

To install or update all required *R* and conda packages, download required reference files and set up CytoMeth, run '*install.sh*' and '*install.data.sh*' scripts both located in CytoMeth directory. The first one installs all required R and conda packages and the second one downloads all required reference files and basic example data. For the first time select '*y*' option to install all required by CytoMeth components. All required packages and files should be installed or updated automatically and if that succeeded there is no need of any manual installation presented later below. If there is any missing component and CytoMeth stops with appropriate warning you may take a look at a specific section 'Required Tools' or 'Reference Files'. In that case please also try to rerun the script in a terminal. Please notice that size of reference files is several GB

and it can take a few minutes to download them, however the downloading time strongly depends on your internet connection speed.

```
./install.sh
./install.data.sh
```

Required Tools

Required R packages

The script '*install.sh*' file should install the following *R* packages:

- data.table (CRAN package)
- ggplot2 (CRAN package)
- rjson (CRAN package)
- RColorBrewer (CRAN package)
- yaml (CRAN package)
- stringr (CRAN package)
- methylKit (Bioconductor package)
- GenomicRanges (Bioconductor package)
- genomation (Bioconductor package)

These packages can be also manually installed by typing the command in a terminal window:

```
Rscript R/install.packages.R
```

Required conda packages

The follwing *conda* packages are required by CytoMeth and these packages are automatically installed or updated by '*./install.sh*' command. The current version of CytoMeth was succesfully tested on versions presented below:

- bsmap (ver. 2.90)
- bamtools (ver. 2.5.1)
- bamutil (ver. 1.0.14)
- bedtools (ver. 2.27)
- seqtk (ver. 1.3)
- fastqc (ver. 0.11.8)
- samtools (ver. 1.9)

These tools can be also manually installed by typing the command in the a terminal window:

```
conda update conda
conda update conda-build
conda install -y -c bioconda bsmap
conda install -y -c bioconda bamtools
conda install -y -c bioconda bamutil
conda install -y -c bioconda bedtools
conda install -y -c bioconda seqtk
conda install -y -c bioconda fastqc
conda install -y -c bioconda samtools
```

Required Java Tools

Java tools (in CytoMeth '*/tools/*' directory) are provided with CytoMeth with the following versions:

- Trimmomatic (ver. 0.36)

- GATK (ver. 3.8.1)
- picard (ver. 1.141) or picard2 (ver. 2.20.6)

Required 'conda.info' file

CytoMeth also requires in its main directory '*conda.info*' file that can be manually created by typing the following command in a terminal window:

```
conda info --json > conda.info
```

This file is also automatically created during installation process.

Reference Files

Reference files required by CytoMeth are automatically installed by '*install.data.sh*' script. This data contains human reference genome (hg38), SeqCap_EPI_CpGiant_hg38 Roche methylome panel, Ensemble gene annotation data and CpG Island coordinates data. If you would like to download them manually plese run the following commands in a terminal window:

```
wget -c -O ./referenceData/CytoMethRefData.zip http://zbo.ipipan.waw.pl/tools/CytoMeth/referenceData/Cy
unzip ./referenceData/CytoMethRefData.zip;
```

Set of optional reference files that is also available to download contains contains human reference genome (hg38) inclding additional reference genome NC_001416 phage widely used for conversion efficiency evaluation.

```
wget -c -O ./referenceData/CytoMethRefDataNC_001416.zip http://zbo.ipipan.waw.pl/tools/CytoMeth/referen
unzip ./referenceData/CytoMethRefDataNC_001416.zip;
```

All reference files are located in */referenceData/* directory by default.

Basic Example Data

Basic example data may be downloaded by wget command:

```
wget -c -O ./input/small_FAKE03_R1.fastq http://zbo.ipipan.waw.pl/tools/CytoMeth/input/small_FAKE03_R1.
wget -c -O ./input/small_FAKE03_R2.fastq http://zbo.ipipan.waw.pl/tools/CytoMeth/input/small_FAKE03_R2.
```

The Docker

CytoMeth project is also available as a docker. The CytoMeth docker is a virtual machine that contains all the environment (apps and libraries) ready to run CytoMeth. To download and run CytoMeth docker please install Docker app from https://www.docker.com/. After successfull instalation of Docker app you may build your own CytoMeth docker from the sources or download ready to use CytoMeth docker from Docker Hub Downloading the docker from Docker Hub.

Building your own docker locally

To build your own docker use *build* command and after the successful creation the docker is ready to run. Pleaese notice building of the docker may take tens of minutes because the proper environment must be created from the scratch, however it must be done only once.

To get CytoMeth from the github repository you may download it as a zip file or clone the project:

```
git clone https://github.com/mdraminski/CytoMeth.git
cd CytoMeth
```

To build your own docker run the command in the CytoMeth directory:

```
docker build -t cytometh .
```

Downloading the docker from Docker Hub

The docker ready to go is also publicly available on Docker Hub and can be pulled to your system by the command below. The second command adds a new docker tag so the name of the local docker image is cytometh instead of inconvenient e.g. mdraminski/cytometh:2.

```
docker pull mdraminski/cytometh:3
docker tag mdraminski/cytometh:3 cytometh
```

Running the docker

To run the docker that is already built in your system or pulled from Docker Hub type the command below. Please notice that you are running fresh virtual machine session and this image does not yet contain '*input*' and '*referenceData*' folders. However they may be created and filled by '*install.data.sh*' script (See below '*Reference data*' section).

```
docker run -it cytometh /bin/bash
```

Notice all data that you download or create (e.g. results) within the docker session is available until its shut down. Therefore it is highly recomennded to share the folder between the docker and the host system (for data and results transfer). To run the docker that shares the folder between host system and the docker it is needed to specify it right after '-v' parameter e.g. to share Desktop folder in your home folder run the command below:

```
docker run -it -v ~/Desktop:/Desktop cytometh /bin/bash
```

Please remember if you want to share the folder '*Desktop*' between your docker and local system you need to modify the following '*config.yml*' paths accordingly:

```
input_path: "/Desktop/input/"
results_path: "/Desktop/results/"
ref_data_path: "/Desktop/referenceData/"
```

In this case please make sure your input folder has full access to all users [777].

```
chmod -R 777 /Desktop/input/
```

Quit from the docker

To shut down the virtual machine type command 'exit'. It is similar as quiting from ssh session.

Reference data

Reference data is several Gigabytes big therefore it is not included in the parent docker. However after successful running of the docker on your machine you can download the data by running '*install.data.sh*' script in the CytoMeth main directory.

```
./install.data.sh
```

After successful installation of the reference data CytoMeth docker is ready to use. All reference files are located in */referenceData/* directory by default. However they disappear after turning off the docker. Therefore it is recommended to use shared folder (see the section '*Running the docker*'). In this case it is recommended to copy and run the script '*install.data.sh*' directly in the shared folder. Please remember to set up all the paths to the new shared folder in your '*config.yml*' file.

CytoMeth Usage

Configuration

File 'config.yml'

The file '*config.yml*' contains CytoMeth input parameters and before use of CytoMeth please define your processing. The default settings look like below:

```yaml
#General Params
verbose: TRUE
threads: 8
memory: 16G
overwrite_results: FALSE
clean_tmp_files: TRUE
remove_clipped_bam: FALSE
plot_format: "pdf"

#in/out paths
input_path: "./input/"
results_path: "./results/"
#anaconda_bin_path: "/opt/anaconda/bin/"

### Reference Data - Path
ref_data_path: "./referenceData/"
### Reference Data - Files
ref_data_sequence_file: "hg38_phage.fa"
ref_data_intervals_file: "SeqCap_EPI_CpGiant_hg38_custom_liftOver_phage.bed"
ref_control_sequence_name: "NC_001416"

### Reference Data - Remaining Files
ref_data_trimmomatic_adapter: "Trimmomatic/adapters/TruSeq3-PE-2.fa"
ref_data_CpgIslandAnnotation: "cpgIslandExt.hg38.bed"
ref_data_CpGGenomAnnotation: "geneAnnotationEnsemble.hg38.bed"

# Specific Tools params
trimmomatic_MINLEN: 50
sqtk_run: FALSE
sqtk_subset: 10000000
min_depth: 1
#meth_tool: ['methratio', 'bssnper']
meth_tool: 'methratio'
### methratio_processing: ["allCHR","batchCHR"]
methratio_processing: "batchCHR"
```

Input parameters:

- verbose - prints additional info and commands on the screen
- threads - defines number of threads used by tools. Most of the tools does not gain any processing speed for more than 10-12 threads.
- memory - amount of memory dedicated to Java and other tools. If you see Java 'out of memory' error or any sudden stop of the program please increase the parameter. The minimum amount that is recommended for human genome analysis is 6GB. Plese use one of the following sufixes: 'M', 'G', 'T' (case sensitive: mega, giga, tera).
- overwrite_results - if TRUE then all result files from the sample processed again will be overwritten. If FALSE CytoMeth will skip phases that related phase result file exists in apriopriate results_path.
- clean_tmp_files - if TRUE all useless temporary files will be removed after the processing of the sample.
- remove_clipped_bam - whether to remove final clipped.bam file or not
- plot_format - set up plot format of report files. Available formats: 'pdf','png', 'eps', 'tiff', 'jpg'.
- input_path - defines path to input fastq R1/R2 files, all samples existing in this directory will be processed in batch process.
- results_path - the path to keep all temporary and result files.
- anaconda_bin_path - path to conda and conda packages. This parameter is retrieved from 'conda.info' file and it is commented out by default. If you want to specify specific path to conda/bin directory uncommend it and define. This parameter overwrites the setting from 'conda.info' file.
- ref_data_path - defines path to the reference data
- ref_data_sequence_file - additional control sequence file (see Input files section) by default it is set on 'hg38_phage.fa'.
- ref_data_intervals_file - panel file (see Input files section) by default it is set on SeqCap_EPI_CpGiant_hg38_custom_liftOver_phage.bed'
- ref_control_sequence_name - name of control sequence (by default phage sequence)
- trimmomatic_MINLEN - MINLEN parameter of the trimmomatic tool.
- sqtk_run - if TRUE initial sqtk sampling is processed.
- sqtk_subset - size of the subset to select by the sqtk tool.
- min_depth - the minimum coverage to call variants/methylomes
- meth_tool - defines the tool for final calculation of beta values (Methratio or BS-Snper). BS-Snper tool also provides SNPs in additional vcf file. Notice that it is possible to run both tools to obtain results from both eg. meth_tool: ['methratio', 'bssnper'].
- methratio_processing - determines if methratio process should be run gene by gene (better for big input samples) or all genes at once (faster for small input data samples). It is set on 'batchCHR' by default.

File 'tools.conf.yml'

The file '*tools.conf.yml*' contains CytoMeth tools parameters and it is located in tools directory. The settings in the file configure paths and names of all tools needed by CytoMeth processing default values are highly recommended. The file by default is defined as below:

```
### TOOLS - path and tools cfg file
tools_path: "./tools/"
tools_config: "tools.conf.csv"

### TOOLS Definition
bedtools: "bedtools"
samtools: "samtools"
bamtools: "bamtools"
bamUtil: "bam"
bsmap: "bsmap"
methratio: "methratio/methratio.py"
trimmomatic: "Trimmomatic/trimmomatic-0.36.jar"
```

```
picard: "Picard/picard.jar"
picard_ver: 1
gatk: "GATK/GenomeAnalysisTK.jar"
fastqc: "fastqc"
seqtk: "seqtk"
bisSNP:
bssnper: "BS-Snper/BS-Snper.pl"

### python2 path
python2: "python2"
```

Input files

Before you run the processing you need to:

- Copy your nucleotide sequences FASTA R1/R2 files (both in *.fastq* format) named: 'SAMPLENAME_R1.fastq' and 'SAMPLENAME_R2.fastq' (where 'SAMPLENAME' is a unique name of your sequenced sample) to the '*./input/*' directory.
- If your files are compressed (*.gz* format) please decompress before use:

```
gunzip -c SAMPLENAME_R1.fastq.gz > SAMPLENAME_R1.fastq
gunzip -c SAMPLENAME_R2.fastq.gz > SAMPLENAME_R2.fastq
```

- Prepare reference FASTA (in .fa or .fasta format) file with additional control sequence. CytoMeth comes with 'hg38_phage.fa' reference file with an additional sequence used as control (phage DNA sequence) and the file 'hg38.fa' without that additional seqence. Any new reference '.fa' file requires corresponding '.fai' and '.dict' files that should be generated. The control is used to estimate bisulfite conversion efficiency. Remember to add the sequence of your control e.g. enterobacteria phage lambda genome to the reference genome file so that captured controls can be mapped to the lambda genome. Reassuming, the reference genome file must be extended with a control sequence. Notice that all reference files are located in */referenceData/* directory.
- Prepare panel file (in .bed format) with panel coordinates and control coordinates '*SeqCap_EPI_CpGiant_hg38_custom_liftOver_phage.bed*'. If you used different panel or performed whole genome analysis, please prepare the '.bed' file defining genomic regions covered by your design, to compute not biased statistics. **Important**: Check if your panel file (bed format) control sequence coordinates has the same name (header ID) as in reference fasta file.

Running the CytoMeth Processing

To run entire CytoMeth processing and summary reporting please run '*CytoMethRun.sh*' bash script in a terminal window:

```
./CytoMethRun.sh
```

It is also possible to run the batch processing separately for all samples located in *'/input/'* directory. If it is required please type in a terminal window:

```
Rscript R/CytoMeth.R
```

When above processing is finished create summary quality report on all results files located in *'/results/QC_report'* directory:

```
Rscript R/CytoMethQC.R
```

The script above creates summary csv file that aggregates quality measures values for all processed samples. It also creates two barplots: overall coverage report plot, CpG vs nonCpG frequency report. The methylation results can be also visualised in respect to specific genomic regions. We annotate the level of methylation to CpG islands, promoters, intergenic regions, introns and exons and provide proper plots in '*results/QC_report*' directory.

It is also possible to define your own processing chain and run multiple experiments on different input and output folders or different set of input parameters. To set up the CytoMeth process manually plese edit '*CytoMeth.R*' file.

```
source("./R/main.R")
#read default config from the config.yml file
conf <- readConfig()
#set up required parameters e.g. input path
conf$input_path <- "./myinputfolder/"
conf$overwrite_results <- F

# run batch processing of all files located in conf$input_path folder
CytoMeth(conf)

# For single sample processing it is required to define R1 and R2 files
CytoMeth(conf, file.path(conf$input_path,"small_FAKE03_R1.fastq"),
  file.path(conf$input_path,"small_FAKE03_R2.fastq"))
```

Output files

Methylation beta values

Methratio results:

The result files are located in *'/results/methyl_results'* directory - for each input sample there are two types of output files:

- 'SAMPLENAME.methylation_results.bed' - text file in bed format
- 'SAMPLENAME.methylation_results.rds' - binary file easy to read in R by *readRDS* function

The example output file is presented below:

```
    chr  start    end context betaVal strand coverage numCs numTs  posCs
100 chr1 135081 135082     CHH   0.000      +       18     0    18 135081
101 chr1 135082 135083     CHG   0.000      +       18     0    18 135082
102 chr1 135083 135084     CHG   0.000      -       14     0    14 135084
103 chr1 135085 135086     CHG   0.000      +       19     0    19 135085
104 chr1 135086 135087      CG   0.947      +       19    18    19 135086
105 chr1 135091 135092     CHH   0.000      -       20     0    20 135092
```

BS-Snper results:

The result files are located in *'/results/bssnper'* directory - for each input sample there are three types of output files:

- 'SAMPLENAME.bssnper.vcf' - the standard VCF file
- 'SAMPLENAME.methylation_results.bed' - text file in bed format
- 'SAMPLENAME.methylation_results.rds' - binary file easy to read in R by *readRDS* function

The 'SAMPLENAME.methylation_results.bed' file is formatted the same way like Methratio file

FastQC report files

The result files are located in *'/results/QC/FastQC'* directory. For each sample there are four output files:

- 'SAMPLENAME_R1_fastqc.zip'
- 'SAMPLENAME_R2_fastqc.zip'
- 'SAMPLENAME_R2_fastqc.html'
- 'SAMPLENAME_R2_fastqc.html'

There is no need to unzip these files, FastQC report is available by opening html file in the browser.

CytoMeth Quality Control report files

After the processing of each single sample CytoMeth creates a summary file associated with that sample. The default location for this summary file is '*results/QC_report/SAMPLENAME_QC_summary.yml*'. Example of that *yml* file is below (created for fake data):

```
Sample_ID: small_FAKE03
Input_read_pairs: 289087.0
Read_Pairs_Surviving_trimming: 289086.0
Prc_Read_Pairs_Survaving_trimming: 100.0
Prc_duplicated_reads_top: 7.5427
Prc_duplicated_reads_bottom: 7.7892
Number_of_reads_after_removing_duplicates: 533815.0
Number_of_reads_after_filtering: 474034.0
Prc_passed_filtering_step: 88.8011764
Number_of_on_target_reads: 533906.0
Prc_of_on_target_reads: 100.0170471
Mean_coverage: 0.41
Number_of_Cs_in_control: 0
Conversion_eff: NaN
Number_of_Cs_in_panel: 295448
Number_of_Cs_in_panel_CpG: 42450
Number_of_Cs_in_panel_non_CpG: 252998
Number_of_Cs_in_panel_non_CpG_cov_min10: 189937
Number_of_Cs_in_panel_CpG_cov_min10: 33856
Number_of_Cs_in_panel_CpG_cov_max9: 8594
Prc_of_Cs_in_panel_CpG_cov_min10: 79.7550059
Prc_of_Cs_in_panel_CpG_cov_max9: 20.2449941
processing_time: 3.3 hours
```

Notice that in the result file you may find separate conversion stats from both tools (Methratio and BS-Snper) if parameter 'meth_tool' was set respectively.

CytoMeth Quality Control summary file

The file '*results/QC_report/SummaryQC.csv*' contains all QC report files results for all samples.

CytoMeth Quality Control plots

The directory '*results/QC_report/*' contains set of plot files:

- 'SAMPLENAME_histCpGStats.pdf'
- 'BetaValuesSummary.pdf'

- 'SummaryCntCommonCpG.pdf'
- 'SummaryCpGCoverage.pdf'
- 'SummaryHypermethylatedCpGGenomAnnotation.pdf'
- 'SummaryHypermethylatedCpGIslandsAnnotation.pdf'
- 'SummaryHypomethylatedCpGGenomAnnotation.pdf'
- 'SummaryHypomethylatedCpGIslandsAnnotation.pdf'
- 'SummaryMethylationLevel.pdf'
- 'SummarySitesCovBy10CpGnonCpG.pdf'
- 'SummarySitesCovBy10.pdf'

Version

For more information please see CHANGES.md

- Version: 1.0.0
- Date: 20.05.2021

Authors

This tool has been created and implemented by:

- Michal Draminski [1] (author, developer, maintainer)
- Agata Dziedzic [1] (author, developer)
- Rafal Guzik [3] (author)
- Bartosz Wojtas [2] (author)
- Damian Loska [1] (developer)
- Michal J. Dabrowski [1] (author)

1. Computational Biology Lab, Polish Academy of Science, Warsaw, Poland
2. Neurobiology Center, Nencki Institute of Experimental Biology, Warsaw, Poland
3. Andrzej Frycz Modrzewski Krakow University, Faculty of Medicine and Health Sciences, Department of Biochemistry, Poland.

License

The set of tools, used by or provided with CytoMeth is under following licenses:

- fastqc - GNU General public license https://www.bioinformatics.babraham.ac.uk/projects/download.html
- trimmomatic - GNU GENERAL PUBLIC LICENSE https://github.com/timflutre/trimmomatic/blob/master/distSrc/LICENSE
- bsmap - GNU Public License (GPL) https://github.com/genome-vendor/bsmap/blob/master/README.txt
- picard - The Picard toolkit is open-source under the MIT license and free for all uses. https://broadinstitute.github.io/picard/
- bamtools - The MIT License https://github.com/pezmaster31/bamtools/blob/master/LICENSE
- bamUtil - GNU General Public License https://github.com/statgen/bamUtil/blob/master/src/Validate.h

- GATK - BSD 3-Clause “New” or “Revised” License https://github.com/broadinstitute/gatk/blob/master/LICENSE.TXT
- BisSNP - GNU General Public License https://github.com/dnaase/Bis-tools/blob/master/Bis-SNP/src/main/java/edu/usc/epigenome/uecgatk/bissnp/BisSNP.java
- samtools - The MIT/Expat License https://github.com/samtools/htslib-plugins/blob/master/LICENSE
- BS-Snper - https://github.com/hellbelly/BS-Snper

Acknowledgments

- We would like to thank to everyone who helped and contributed to this work.
- This work is supported by a grant from the Polish National Science Centre [DEC-2015/16/W/NZ2/00314].