UNIVERSITÉ DE NANTES

UNIVERSITÉ DE NANTES

UFR SCIENCES ET TECHNIQUES

Manuel d'utilisation :
PipelineERRBS

Jennifer RONDINEAU
Master 2 Bioinformatique

Équipe 11 "Oncogénomique intégrative de la genèse et de la progression du myélome multiple",
Center for Cancer Research Nantes-Angers
UMR 892 Inserm - 6299 CNRS / Université de Nantes

UNIVERSITÉ DE NANTES

Instituts thématiques

Inserm

Institut national de la santé et de la recherche médicale

CRCNA

Centre de Recherche en Cancérologie Nantes Angers

14 mars 2016 — 16 septembre 2016

Sommaire

1 Présentation **1**

2 Installation de la PipelineERRBS **1**

2.1 Les installations pré-requises 1

2.1.1 Installation de Trim_galore 1

2.1.2 Installation de bowtie2 2

2.1.3 Installation de Bismark 2

2.1.4 Installation de Samtools 2

2.1.5 Installation de fastQC 3

2.1.6 Installation de R et des packages nécessaires 3

2.2 Téléchargement de l'outil 4

2.3 Installation de l'outil 4

3 Guide d'utilisation **4**

3.1 Étape 1 : Filtration des données et alignement des reads 4

3.2 Étape 2 : Extraction des CpGs 5

3.3 Étape 3 : Détermination des DMCs et DMRs chez un patient entre deux conditions 5

3.4 Étape 4 : Annotation 6

3.5 Options supplémentaires 7

3.5.1 Détermination des DMCs et DMRs entre deux groupes(cas/témoins) 7

3.5.2 Obtention de la couverture des CpGs pour un patient 7

4 Informations sur les séquences de références fournies **7**

5 Contacts **7**

1 Présentation

PipelineERRBS est un package python crée dans le but de rassembler tous les outils nécessaires afin de pouvoir traiter des données ERRBS. Pour cela, ce pipeline fait appel à un ensemble d'outils : bismark, samtools, methylKit, eDMR, HOMER... L'innovation qu'apporte cet outil réside en la fluidité des étapes d'analyses des données. En effet, l'installation de ce package permet d'avoir, dans un même outil, toutes les fonctions nécessaires :

1. au filtrage des données,
2. à l'alignement des reads contre un génome de référence (hg19),
3. à l'extraction des CpGs,
4. au calcul des cytosines et régions différentiellement méthylées entre deux conditions,
5. à l'annotation de ces régions différentiellement méthylées, et à l'étude des voies biologiques dans lesquelles elles sont impliquées.

Le code de PipelineERRBS est disponible en ligne :

https://github.com/JenniferRondineau/PipelineERRBS

2 Installation de la PipelineERRBS

Cet outil python a été créé sous Linux, et il nécessite l'installation au préalable de plusieurs outils décrits ci dessous.

2.1 Les installations pré-requises

2.1.1 Installation de Trim_galore

Trim_galore est un outil permettant de filtrer des contaminations par des adaptateurs illumina, il utilise par défaut les 13 premières paires de bases de l'adaptateur 'AGATCGGAAGAGC'.

Son site web pour plus d'informations :

http://www.bioinformatics.babraham.ac.uk/projects/trim_galore/

Installation :

```
wget "http://www.bioinformatics.babraham.ac.uk/projects/trim_galore/trim_galore_v0.4.1.
    zip"

unzip trim_galore_v0.4.1.zip
cd trim_galore_zip/
gedit ~/.bashrc
```

Puis ajout du programme dans le PATH, (ajout à la fin du fichier bashrc la ligne suivante) :

```
export PATH=$PATH:<YOUR_PATH>/trim_galore/
```

2.1.2 Installation de bowtie2

Bowtie2 est un outil permettant l'alignement de reads sur un gènome de référence, il est notamment utilisé par un autre outil nécessaire pour l'analyse des données ERRBS : Bismark.

Son site web pour plus d'informations :

http://bowtie-bio.sourceforge.net/bowtie2/index.shtml

Installation :

```
wget http://freefr.dl.sourceforge.net/project/bowtie-bio/bowtie/0.12.8/bowtie-0.12.8-src
    .zip
unzip bowtie-0.12.8-src.zip
cd bowtie-0.12.8
make
```

2.1.3 Installation de Bismark

Bismark a été spécialement conçu pour aligner des séquences traitées au bisulfite de sodium contre un génome d'intérêt.

Son site web pour plus d'informations :

http://www.bioinformatics.babraham.ac.uk/projects/bismark/

Installation :

```
wget "http://www.bioinformatics.babraham.ac.uk/projects/bismark/bismark_v0.16.1.tar.gz"

tar -xvzf bismark_v0.16.1.tar.gz
cd bismark_v0.16.1/
gedit ~/.bashrc
```

Puis ajout du programme dans le PATH, (ajout dans le fichier bashrc à la fin cette ligne) :

```
export PATH=$PATH:<YOUR_PATH>/bismark_v0.16.1/
```

2.1.4 Installation de Samtools

Samtools est un outil permettant de manipuler facilement les fichiers de type BAM et SAM.

Son site web pour plus d'informations :

http://samtools.sourceforge.net/

Installation :

```
sudo apt-get install git
git clone git://github.com/samtools/samtools.git
cd samtools
./configure
make
make install
```

2.1.5 Installation de fastQC

FastQC est un outil permettant de vérifier la qualité des données de séquençage.

Son site web pour plus d'informations :

http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

Installation :

```
wget "http://www.bioinformatics.babraham.ac.uk/projects/fastqc/fastqc_v0.11.5.zip"
unzip fastqc_v0.11.5.zip
cd FastQC
chmod 755 fastqc
sudo ln -s /path/to/FastQC/fastqc /usr/local/bin/fastqc
```

2.1.6 Installation de R et des packages nécessaires

Pour le calcul des DMRs et des DMCs, deux packages R sont nécessaires : methylKit et eDMR.

Leur site web pour plus d'informations :

https://github.com/al2na/methylKit
https://github.com/ShengLi/edmr

Le package eDMR nécessite une version de R actuellement en développement, la version 3.4.0, vérifier au préalable que vous disposé bien de cette version.

Installation des packages dans R :

```
R Under development (unstable) (2016-05-17 r70628) -- "Unsuffered␣Consequences"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R est un logiciel libre livre sans AUCUNE GARANTIE.
Vous pouvez le redistribuer sous certaines conditions.
Tapez 'license()' ou 'licence()' pour plus de details.

# installation du package methylKit
> library(devtools)
> install_github("al2na/methylKit", build_vignettes=FALSE,
  repos=BiocInstaller::biocinstallRepos(),
  dependencies=TRUE)
> library(methylKit)

# installation du package eDMR
> install.packages( c("data.table", "mixtools", "devtools"))
> source("http://bioconductor.org/biocLite.R")
> biocLite(c("GenomicRanges","IRanges"))
> library(devtools)
> install_github("edmr", username = "ShengLi",build_vignettes=FALSE)
> library(edmr)
```

2.2 Téléchargement de l'outil

PipelineERRBS est disponible en ligne sur Github :

https://github.com/JenniferRondineau/PipelineERRBS

Pour le télécharger, il suffit de taper en ligne de commande :

```
git clone "https://github.com/JenniferRondineau/PipelineERRBS.git"
```

2.3 Installation de l'outil

Une fois téléchargé, il ne reste plus qu'à l'installer :

```
cd PipelineERRBS
sudo python setup install
gedit ~/.bashrc
```

Puis rajouter à la fin de votre bashrc ces deux lignes, en remplaçant "<PATH>" par le chemin absolu de la localisation de votre installation :

```
export PipelineERRBS_PATH=<PATH>/PipelineERRBS/PipelineERRBS/scriptR
export PipelineERRBSdata_PATH=<PATH>/PipelineERRBS/PipelineERRBS/data
```

3 Guide d'utilisation

PipelineERRBS présente 4 principales options, correspondant aux grandes étapes du traitement des données :

1. Filtration des données et alignements des reads contre un gènome de référence (hg19)
2. Extraction des CpGs
3. Détermination des cytosines et régions différentiellement méthylées (DMC et DMR) entre deux conditions
4. Annotation des DMCs et des DMRs par HOMER (analyse des voies biologiques dans lesquels ils sont impliqués également possible en même temps)

Deux options supplémentaires sont également disponibles pour l'instant.

3.1 Étape 1 : Filtration des données et alignement des reads

Pour cette étape, PipelineERRBS proposent deux alternatives, soit les données sont paired-end :

```
PipelineERRBS align --paired -1 <fastqR1> -2 <fastqR2> -g <genomefolder>  -o <outputdir>
```

Soit les données sont single-end :

```
PipelineERRBS align --single <fastq> -g <genomefolder> -o <outputdir>
```

Les fichiers donnés en entrée doivent être sous la forme de "fastq" ou "fastq.gz". Pour l'option "-g" il faut donner le PATH où se situe le génome de référence sous forme de fasta (".fa"). Et enfin l'option -o indique le dossier dans lequel vous désirez retrouver les résultats.

Grâce à ces lignes de commandes, les données sont filtrées par Trim_galore, et les reads sont alignés par bismark sur le génome de référence de votre choix.

3.2 Étape 2 : Extraction des CpGs

L'option "extractionCpG" permet d'extraire à partir des fichiers de séquençage (".sam") toutes les cytosines présentant une couverture supérieure à 10X, associées à leur niveau de méthylation :

```
PipelineERRBS extractionCpG -f <SAM FILE> -o <outputdir>
```

Le fichier d'entrée doit être un fichier ".SAM" trié. En sortie, on obtient un fichier "_CpG.txt", exemple :

chrBase	chr	base	strand	coverage	freqC	freqT
chr1.10563	chr1	10563	F	513	97.66	2.34
chr1.10542	chr1	10542	F	514	97.28	2.72
chr1.10564	chr1	10564	R	632	100.00	0.00
chr1.10577	chr1	10577	F	258	16.67	83.33
chr1.10571	chr1	10571	F	513	98.44	1.56
chr1.10590	chr1	10590	R	314	0.00	100.00
chr1.10526	chr1	10526	R	639	99.37	0.63

"FreqC" donne le pourcentage de fois, pour cette base, qu'un C est retrouvé (donc le pourcentage de cytosines méthylées puisque toutes les cytosines non méthylés ont été transformés en Thymine grâce à la technique RRBS).

3.3 Étape 3 : Détermination des DMCs et DMRs chez un patient entre deux conditions

Pour la détermination des DMCs, ce pipeline utilise methylKit, qui caractérise une cytosine comme étant différentiellement méthylées dès lors qu'elle présente une différence de méthylation supérieure à 25% entre les deux conditions, ainsi qu'une qvalue<0.01.

Pour le calcul des DMRs, il utilise eDMR qui regarde la distribution de la distance entre les CpGs dont la couverture est supérieure à 10x, il utilise une distribution normale bimodale pour identifier la coupure entre deux DMRs. Une région est caractérisée comme étant une DMR si elle contient au moins un DMC, au moins 3 CpGs inclus et que la différence absolue de méthylation soit supérieure à 20%.

```
PipelineERRBS methylDiffbyPatient --control <sam> OR <_CpG.txt> --case <sam> OR <_CpG.
    txt> --name <str> -o <outputdir>
```

Les 4 arguments (–control, –case, –name et -o) sont obligatoires. Pour les arguments –control et –case, il est nécessaire de fournir soit un fichier SAM trié, soit un fichier "_CpG.txt" (fichier de sortie de l'option extractionCpG).

L'option –name permet de donner un identifiant à cette analyse différentielle.

L'option -o permet d'indiquer le dossier de sortie pour les résultats.

En sortie, on obtient 2 fichiers textes, un contenant les DMCs l'autre contenant les DMRs, ainsi que quelques illustrations concernant la disposition des régions.

Exemple de fichier DMCs:

chr	start	end	strand	pvalue	qvalue	meth.diff
chr1	135028	135028	+	5,48911393654451e-05	0,000307364059412726	-31,9736842105263
chr1	135031	135031	+	1,77115235188938e-05	0,000109650587363739	-32,7745466985973
chr1	135150	135150	+	1,03802714052854e-08	1,10526839861084e-07	-55,8677098150782
chr1	135161	135161	+	4,26825137560199e-09	4,80404208575384e-08	-44,6126126126126
chr1	135172	135172	+	2,69918132874183e-05	0,000161064925394973	-41,71826625387
chr1	135190	135190	+	3,74562877891635e-05	0,000217118873158736	-34,4197138314785

Exemple de fichier DMRs :

chr	starts	ends	strands	mean.meth.diff	DMR.pvalue	DMR.qvalue
chr1	879382	879510	+	-27.7376709840854	0.000461603787312931	0.000980710531908333
chr1	917726	917820	-	-38.3941207117346	5.63656051901432e-05	0.000169090892947209
chr1	918341	918430	+	-54.6546846265812	0.000405874565175383	0.000880458843230543
chr1	941271	941334	+	-30.511762905449	0.000175332180678109	0.000438445630056934
chr1	944199	944364	-	-21.4768637328734	0.000161951816997515	0.000410450454973729
chr1	986848	987050	+	-21.6039453334345	0.000394628178359929	0.0008616269877773
chr1	1052599	1052656	-	36.4372634523558	4.62311600320429e-08	3.90502845832789e-07

Exemples d'illustrations :

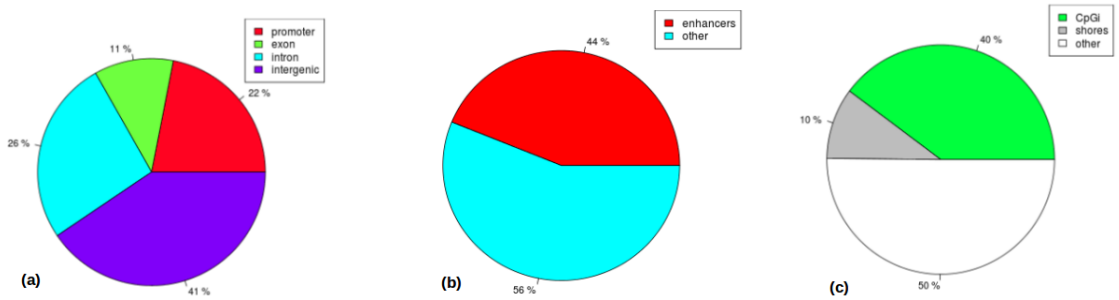

Figure 1: a) Annotation différentielle de la méthylation sur le gène b) Annotation de la méthylation différentielle par rapport aux enhancers c) Annotation de la méthylation différentielle par rapport aux ilôts CpGs

3.4 Étape 4 : Annotation

Pour l'annotation des DMCs et des DMRs, ce pipeline utilise HOMER, qui est une suite d'outils pour la découverte de motifs et pour l'analyse de données issues du séquençage à haut débit, et plus particulièrement, il utilise sa fonction "annotatePeaks.pl" qui permet d'associer des régions génomiques au RefSeq (génome hg19).

```
PipelineERRBS annotate -b <BEDFILE> -s <FILE> --go <output directory> -o <OUTPUTFILE>
```

Les options obligatoires sont "-b" et "-o", le fichier d'entré contenant les positions à annoter (chr, start, end) et le fichier de sortie qui va contenir l'annotation.

L'option "-s" est facultative et permet d'obtenir des statistiques sur l'annotation.

L'option "–go" est facultative également et permet d'obtenir un dossier complet contenant les "biological process", les pathways etc... dans lesquels sont impliquées ces régions annotées.

3.5 Options supplémentaires

3.5.1 Détermination des DMCs et DMRs entre deux groupes (cas/témoins)

Un script est disponible dans le dossier "PipelineERRBS/scriptR/" permettant d'effectuer une analyse différentielle de profil de méthylation entre deux groupes (cas/témoins), pour cela, il suffit de modifier comme dans l'exemple du fichier 'methylDiffbyGroup.R', les paths des fichiers "_CpG.txt" des cas et des témoins.

```
Rscript $PipelineERRBS_PATH/methylDiffbyGroup.R
```

On obtient en sortie, les mêmes types de fichiers que pour une analyse 'methylDiffbyPatient'.

3.5.2 Obtention de la couverture des CpGs pour un patient

Cette option permet d'obtenir tous les CpGs qui sont couverts dans les deux conditions (par exemple tous les CpGs d'un patient qui sont couverts par la technique au diagnostic de sa maladie et à sa rechute).

```
PipelineERRBS coveredCpG --control <_CpG.txt> --case  <_CpG.txt> --name <str> -o <
    outputdir>
```

4 Informations sur les séquences de références fournies

Pour l'instant, PipelineERRBS est disponible uniquement pour l'annotation avec le génome de référence hg19. Les fichiers d'annotations fournis avec le packages ("refseq.hg19" et "cpgi.hg19") proviennent de l'UCSC genome browser (https://genome.ucsc.edu/).
Le fichier "all_enhancers" a été crée en regroupant les enhancers décrit dans les projet ENCODE (https://sites.google.com/site/anshulkundaje/projects/epigenomeroadmap) et FANTOM(http://enhancer.binf.ku.dk/presets/), ces données proviennent de plus de 100 lignées cellulaires différentes.

5 Contacts

Si vous avez des questions concernant ce pipeline d'analyse, contactez Jennifer Rondineau :

jennifer.rondineau@etu.univ-nantes.fr