Methods in
Molecular Biology 2959

Springer Protocols

Sweta Rani
Lukasz Skalniak *Editors*

IMMUNO-model in Cancer

Methods and Protocols

cost
EUROPEAN COOPERATION
IN SCIENCE & TECHNOLOGY

Funded by
the European Union

OPEN ACCESS

Humana Press

METHODS IN MOLECULAR BIOLOGY

Series Editor
John M. Walker
School of Life and Medical Sciences
University of Hertfordshire
Hatfield, Hertfordshire, UK

For further volumes:
http://www.springer.com/series/7651

For over 35 years, biological scientists have come to rely on the research protocols and methodologies in the critically acclaimed *Methods in Molecular Biology* series. The series was the first to introduce the step-by-step protocols approach that has become the standard in all biomedical protocol publishing. Each protocol is provided in readily-reproducible step-by-step fashion, opening with an introductory overview, a list of the materials and reagents needed to complete the experiment, and followed by a detailed procedure that is supported with a helpful notes section offering tips and tricks of the trade as well as troubleshooting advice. These hallmark features were introduced by series editor Dr. John Walker and constitute the key ingredient in each and every volume of the *Methods in Molecular Biology* series. Tested and trusted, comprehensive and reliable, all protocols from the series are indexed in PubMed.

IMMUNO-model in Cancer

Methods and Protocols

Edited by

Sweta Rani

Department of Science, South East Technological University, Waterford, Cork, Ireland

Lukasz Skalniak

Department of Organic Chemistry, Jagiellonian University, Kraków, Poland

Humana Press

Editors
Sweta Rani
Department of Science
South East Technological University
Waterford, Cork, Ireland

Lukasz Skalniak
Department of Organic Chemistry
Jagiellonian University
Kraków, Poland

European Cooperation in Science and Technology

ISSN 1064-3745 ISSN 1940-6029 (electronic)
Methods in Molecular Biology
ISBN 978-1-0716-4733-2 ISBN 978-1-0716-4734-9 (eBook)
https://doi.org/10.1007/978-1-0716-4734-9

This Humana imprint is published by the registered company Springer Science+Business Media, LLC, part of Springer Nature.
The registered company address is: 1 New York Plaza, New York, NY 10004, U.S.A.

Preface

Immuno-model is a model invented to study the immune system. This book describes the computational and experimental models that help researchers understand the responses of the immune system in cancer and test experimental immuno-oncology approaches.

Macrophages can adapt to different phenotypes in response to signals from the microenvironment. This book on immuno-model describes methods to profile polarization in macrophages using ELISA. ELISA is widely used in immunology to detect proteins, antibodies, antigens, or hormones in a sample, including immune checkpoint inhibitors. Immune checkpoints are regulatory molecules that control the activation and intensity of immune responses. Glycosylation is the addition of carbohydrate groups to the proteins. Glycosylation of the immune checkpoint proteins not only promotes immune evasion in tumor cells but also holds therapeutic implications. New immune checkpoint inhibitors are warranted for better cancer treatment. One of the chapters evaluates immune checkpoint inhibitors. One of the most studied immune checkpoints is PD-1/PD-L1. There is one chapter detailing the interaction of PD-1/PD-L1 and T cells. Immune checkpoint inhibitors have significantly improved survival rates in cancers but there are patients who do not respond to these treatments. Induction of immunogenic cell death is another therapeutic option for cancer patients.

A lot of research is underway to study tumor immune microenvironment. There are several well-established in vitro models to study interaction between immune cells and cancer cells and these in vitro models are still evolving. Cells can be co-cultured using cell culture inserts or can be grown as 3D spheroids. 3D co-culture model can be used to study the interaction of immune cells and cancer cells to mimic the in vitro microenvironment. Cells can be grown in 3D using different techniques, and one of the techniques is using scaffolds derived from cancer patients. One of the chapters explores immunocompetent preclinical mouse models to study primary and metastatic brain cancer. CAR T-cell therapy is still evolving, and one of the chapters describes the method to generate CAR T-cells.

Bioinformatics has vast applications and plays a central role in immunology by enabling the analysis of large-scale datasets. Deconvolution analysis can be used to study the epigenetic dysregulation in human tumors and the tumor ecosystem. Computational methods can also be used to study the mutated peptides called neoantigens. Bioinformatics allows us to identify therapeutic targets and develop precision immunotherapies.

Waterford, Cork, Ireland — *Sweta Rani*
Kraków, Poland — *Lukasz Skalniak*

Chapter 15

Using Epigenetic Data to Deconvolute Immune Cells in Cancer from Blood Samples

Hatim Boughanem, Sotiris Ouzounis, Maurizio Callari, Rebeca Sanz-Pamplona, Manuel Macias-Gonzalez, and Theodora Katsila

Abstract

DNA methylation plays a crucial role in regulating gene expression and is a hallmark of epigenetic dysregulation in human tumors. High-throughput DNA methylation profiling can unravel intricate patterns in cancer. Moreover, understanding immune cell dynamics is essential for comprehending cancer progression and treatment response. Using DNA methylation data in immune cells, we can apply deconvolution algorithms estimate proportions of major immune cell types, providing insights into immune status and its implications in cancer. Functional analysis can identify specific overrepresented or underrepresented immune cell subsets, potentially uncovering novel biomarkers or therapeutic targets. This pipeline presents a detailed workflow in RStudio for DNA methylation studies and immune cell deconvolution, enhancing reproducibility and efficiency. The workflow integrates preprocessing, analysis, and visualization steps, facilitating robust inference of cell-type proportions from DNA methylation data.

Key words Immune cells, Blood, Cancer, Epigenetic, 450K, EPIC

1 Introduction

DNA methylation plays a crucial role in the regulation of gene expression. Epigenetic dysregulation is considered the hallmark of human tumors, offering valuable insights into disease mechanisms and potential therapeutic targets [1]. Utilizing high-throughput DNA methylation profiling platform holds promise for unraveling the intricacies of DNA methylation patterns in cancer studies. Moreover, understanding the composition and dynamics of immune cells within the tumor microenvironment and peripheral blood is essential for comprehending cancer progression and treatment response [2].

Characterizing immune cell proportions within blood samples is pivotal for understanding immune system dynamics in health and disease. Utilizing DNA methylation data, immune cell deconvolution algorithms enable estimation of major immune cell types,

Sweta Rani and Lukasz Skalniak (eds.), *IMMUNO-model in Cancer: Methods and Protocols*, Methods in Molecular Biology, vol. 2959, https://doi.org/10.1007/978-1-0716-4734-9_15,

including T cells, B cells, natural killer cells, and myeloid cells. Presentation of immune cell proportions provides valuable insights into immune status and its potential implications in cancer progression and treatment response. Furthermore, functional analysis offers a systematic approach to deciphering the biological significance of immune cell profiles. By comparing observed immune cell proportions with predefined immuno-profiling sets, enrichment analysis identifies overrepresented or underrepresented immune cell subsets within samples. Calculation of enrichment scores elucidates the functional relevance of immune cell composition in the context of cancer biology, potentially uncovering novel biomarkers or therapeutic targets [3].

Leveraging DNA methylation analysis and immune cell estimation in cancer studies holds immense potential for unraveling the complex interplay between epigenetic regulation and immune response. Through meticulous data preparation, utilization of relevant software packages, and integration of RStudio pipelines, comprehensive insights into DNA methylation patterns and immune cell dynamics can be attained. Here, we offer a detailed pipeline analysis workflow in Rstudio to enhance reproducibility and efficiency in the context of DNA methylation studies and immune cell deconvolution. This includes facilitate seamless integration of preprocessing, analysis, and visualization steps. This enhances robust inference of sample-specific cell-type proportions from DNA methylation data (Fig. 1).

Therefore, the objectives of this chapter are:

- Understanding the considerations when designing DNA methylation experiments.
- Discussing the steps involved in taking *idat*. files.
- Computing and assessing QC metrics at every step in the workflow.
- Identifying differentially methylated positions and regions.
- Analyzing immune cells within your blood sample.

2 Materials and Methods

2.1 EPIC and 450K Dataset

Our analysis will utilize the *idat*. files, and its related data sheet. This pipeline requires input data in the form of *idat*. files, which represent two distinct color channels before normalization. *idat*. files offer the most comprehensive dataset as they encompass measurements on control probes. While Genome Studio files can be utilized alongside this package, their functionality is limited due to the absence of control probe information. Moreover, Genome Studio output is typically normalized using methods within Genome Studio itself, which are often deemed less effective.

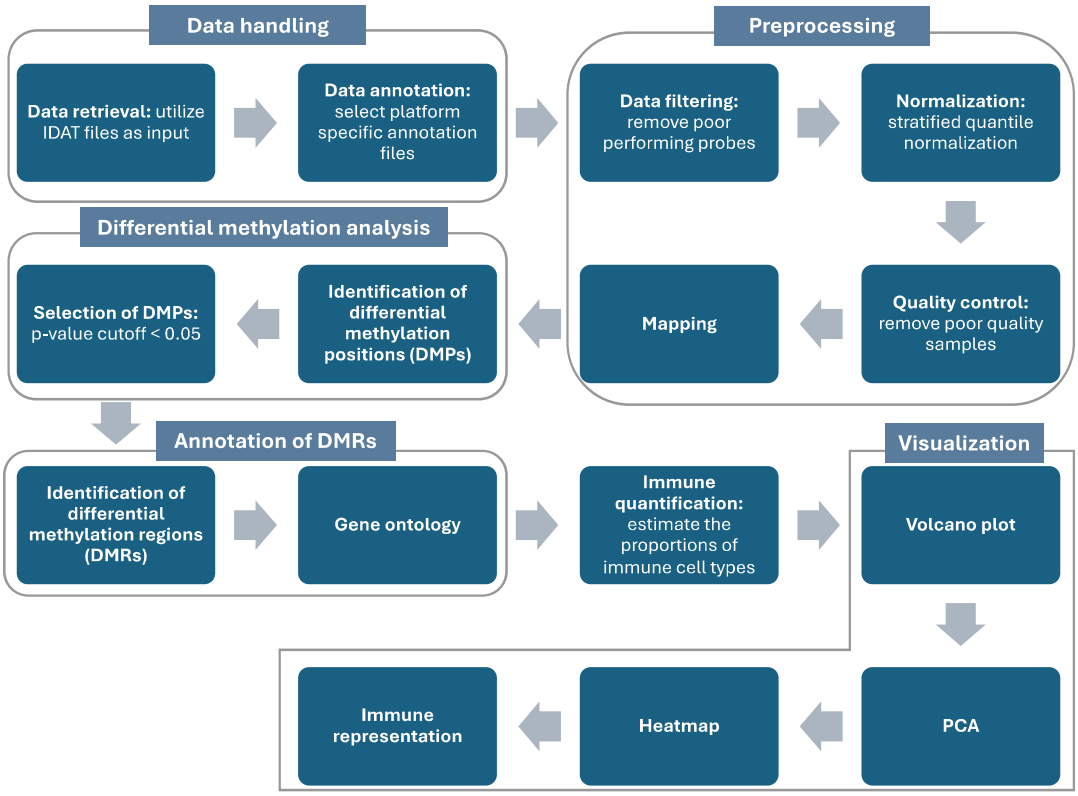

Fig. 1 Workflow of the pipeline. Visualization of a generalized pipeline for the deconvolution of methylation data to quantify immune cell proportions in cancer from blood samples. The pipeline consists of six major steps: (1) Data handling, (2) preprocessing, (3) differential methylation analysis, (4) annotation of differential methylation regions, (5) immune quantification, (6) visualization

For this purpose, we will use a public dataset containing as an example of genome-wide DNA methylation profiling of peripheral blood mononuclear cells (PBMCs) in both normal and breast cancer samples, provided by Wang T. et al. (2023) [4]. Of course, you can use your own dataset to run this example, with some adjustment that are clearly indicated along the pipeline. The Illumina Infinium 850k Human DNA Methylation BeadChip was utilized to obtain DNA methylation profiles across approximately 820,000 CpGs in PBMC samples. The samples included five newly diagnosed breast cancer (GSM7593324, GSM7593325, GSM7593326, GSM7593327, GSM7593328) patients and five normal controls (GSM7593399, GSM7593400, GSM7593401, GSM7593402, GSM7593403) to simplify the model. The users can access to this data by the following link: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE237036

Additionally, this chapter also covers the 450K platform in detail. It is designed to facilitate switching between platforms,

ensuring the use of both platforms depending on the samples available for analysis.

2.2 Computational Resources

RStudio can be installed from the RStudio website: http://www.rstudio.com/, or can be downloaded for all platforms using the link: https://rstudio.com/products/rstudio/download/. The whole analysis is conducted within the R/Bioconductor environment. R (version 4.0.0) should be downloaded and installed from the CRAN website (https://cran.r-project.org/). Bioconductor (version 3.11) can be installed from within the R console using the following commands:

```
if (!requireNamespace("BiocManager", quietly = TRUE)) {
install.packages("BiocManager")
}
BiocManager::install(version = "3.14")
```

2.3 Main Packages Needed

You can install all necessary packages for the analysis from CRAN and Bioconductor. To process *idat*. files, you need to install specific packages: "minfi" for raw data and metadata retrieval, and "IlluminaHumanMethylationEPICmanifest" for genomic annotation on the EPIC platform. For the 450 K platform, use the "IlluminaHumanMethylation450kmanifest" package [5].

```
# You can use each package, you need depending on the platform
you have
BiocManager::install(c('minfi', 'limma' 'IlluminaHumanMethy-
lationEPICmanifest', 'IlluminaHumanMethylation450kmanifest'))
BiocManager::install("IlluminaHumanMethylationEPICanno.
ilm10b4.hg19")

BiocManager::install("DMRcate")
BiocManager::install("FlowSorted.Blood.EPIC")
install.packages("ggfortify")
BiocManager::install("ComplexHeatmap")
BiocManager::install("clusterProfiler")
```

For the visualization of graphs and manipulation of data, it is required to install the packages "ggplot2" and "dplyr," among others.

```
BiocManager::install(c('ggplot2', 'dplyr', 'tibble', 'ggre-
pel'))
```

For regulatory factors enrichment analysis, the package "org.Hs.eg.db" is required for the conversion of the gene names from Entrez ID to Gene Symbols.

```
BiocManager::install('org.Hs.eg.db')
```

For extracting the data that we will use in this example, please install GEOquery and Biobase packages following the next script:

```
BiocManager::install(c('GEOquery', 'Biobase'))
```

For identifying and analyzing differentially methylated regions (DMRs) from Whole Genome Bisulfite Sequencing (WGBS) and Illumina Infinium Array (450K and EPIC) data, we utilize the DMRcate package:

```
BiocManager::install("DMRcate")
```

For deconvoluting the cell type composition in whole blood samples analyzed with the Illumina HumanMethylationEPIC, we will install the package FlowSorted.Blood.EPIC [6]:

```
BiocManager::install("FlowSorted.Blood.EPIC")
```

Finally, please check that all the packages were installed successfully by loading them one at a time using the library() function.

2.4 Creating Annotation File and Data Environment

Depending on the platform we are using, the following script is specially designed for EPIC. In case you are using the 450 K platform, you can replace "IlluminaHumanMethylationEPICanno.ilm10b4.hg19" with "annEPIC" and using "IlluminaHumanMethylation450kanno.ilmn12.hg19," with "ann450K." This modification ensures that the appropriate annotation files are used for the different platforms, allowing the script to accurately process the data. It is crucial to use the correct manifest and annotation files to avoid any discrepancies in the genomic analysis and ensure that the data interpretation is accurate and reliable for your specific platform.

```
# load package
library(minfi)
annEPIC <- getAnnotation("IlluminaHumanMethylationEPICanno.
ilm10b4.hg19")
```

2.5 Creating Data Information Sheet

We will use data from the paper published by Wang T. et al. (2023) https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE237036. Specifically, we will use DNA methylation data from peripheral blood mononuclear cells, including samples from five patients with breast cancer and five normal controls. The samples included five newly diagnosed breast cancer patients (GSM7593324, GSM7593325, GSM7593326, GSM7593327, GSM7593328)

and five normal controls (GSM7593399, GSM7593400, GSM7593401, GSM7593402, GSM7593403). To download these samples, you can manually access the dataset using the following link: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE237036. Alternatively, you can use the GEOquery package to access and download the data using the following script. We will also download the supplementary data from GSE47915, which contains all the raw data we need. This process may take some time.

```
# Load packages
library(GEOquery)
library(Biobase)
library(minfi)

# We will create a file in the desktop named GSE237036, in
which we will download the idat files. We create a function
checking if the directory exist or if it should be created
check_dir<-function(dir_path){
# Check if the directory exists
if (!dir.exists(dir_path)) {
# If the directory does not exist, create it
dir.create(dir_path)
cat("Directory created:", dir_path, "\n")
 } else {
cat("Directory already exists:", dir_path, "\n")
 }

}

# We initailize the working directory of our project
work_dir <- getwd()

# Utilizing the function created above to create the following
directories if they do not exist
dir_path1 <- paste0(work_dir,"/GSE237036") # Directory regard-
ing the data required
dir_path2 <- paste0(work_dir,"/GSE237036/Data") # for the idat
files
dir_path3 <- paste0(work_dir,"/GSE237036/Info") # for the
dataset's information

# Define the URL of the raw data file to be downloaded
raw_dat_url <- "https://ftp.ncbi.nlm.nih.gov/geo/series/
GSE237nnn/GSE237036/suppl//GSE237036_RAW.tar"
# Define the destination file path where the raw data file will
be saved
destfile <- paste0(work_dir,"/GSE237036/GSE237036_RAW.tar")
```

```
# Download the raw data file from the specified URL to the
destination file path
download.file(raw_dat_url, destfile, "curl", quiet = FALSE,
mode = "w",
 cacheOK = TRUE)
# Directory where the tar file will be extracted
out_dir <- paste0(work_dir,"/GSE237036/Data/")
# Extract the contents of the tar file into the specified
directory
untar(destfile, files = NULL, list = FALSE, exdir = out_dir)

# We download the sample sheet with information about the
samples. getGEO function retrieves GEO data corresponding to
the specified GEO accession number (GSE237036). The sample
sheet will be saved in the specified directory
targets <- getGEO(GEO = 'GSE237036', destdir =paste0(-
work_dir,"/GSE237036"))

# Define the URL of the processed matrix data file to be
downloaded
matrix_url <- "https://ftp.ncbi.nlm.nih.gov/geo/series/
GSE237nnn/GSE237036/suppl/GSE237036_matrix_processed.txt.gz"
# Destination file path where the processed matrix data file
will be saved
destfile <- paste0(work_dir,"/GSE237036/GSE237036_matrix_pro-
cessed.txt.gz")
# Download the processed matrix data file from the specified
URL to the destination file path
download.file(matrix_url, destfile, "curl", quiet = FALSE,
mode = "w",
 cacheOK = TRUE)
# File path where the uncompressed matrix data will be saved
out_dir2 <- paste0(work_dir,"/GSE237036/GSE237036_matrix_pro-
cessed.txt")
# Uncompress the downloaded gzipped matrix data file to the
specified destination file path
gunzip(destfile, destname = out_dir2)
```

The getGEOSuppFiles function creates a folder named GSE237036 in your working directory, containing the GSE47915_RAW.tar file, which we will extract in the next step. The getGEO function imports the metadata. We can extract the metadata from the GSE237036 object using the following command. Additionally, we will select the 10 samples that we will use for the analysis, using the package dplyr. if you do not want to preselect the files, just go to the next step.

```
# Select the object
targets <- pData(phenoData(targets[[1]]))

# We select the files that we will use
library(dplyr)
targets <- targets %>%
filter(geo_accession %in% c("GSM7593324", "GSM7593325",
"GSM7593326", "GSM7593327", "GSM7593328",
  "GSM7593399", "GSM7593400", "GSM7593401", "GSM7593402",
"GSM7593403"))
```

(Optional) If you have your own data stored on your computer or if you manually downloaded the data, please follow the next script to import and process your data accordingly.

```
# Load idat files and sample sheet, previously stores in a file
named GSE237036, located in your favorite file. Just place the
url of the file into the read.metharray.sheet
baseDir <- system.file("Your location", package = "minfiData")
targets <- read.metharray.sheet("Your location") # if your
file named EPIC containing idat. and sample sheet is located
in the Desktop, put "~/Desktop/EPIC"
# Preselect the samples we will use for the analysis
targets <- targets %>%
filter(geo_accession %in% c("GSM7593324", "GSM7593325",
"GSM7593326", "GSM7593327", "GSM7593328",
  "GSM7593399", "GSM7593400", "GSM7593401", "GSM7593402",
"GSM7593403"))
```

In the case that you have your own data, just replace location of your file by a directory or file that contain both *idat*. files and sample sheet. Now it is turn to load *idat*. From the file GSE237036 and match with the sample sheet. The initial step in minfi involves reading the *idat*. files using the built-in function read.450 k.exp. for 450K and read.metharray.exp. for EPIC (850 K). Users have various options available: you can specify the sample file names along with the directory path to read them, or you can specify the directory containing the files. In the latter scenario, all files with the *idat*. Extension within the directory will be imported into R. Additionally, users can import a sample sheet and utilize it to load the data into a RGChannelSet object. For more detailed guidance, please refer to the minfi vignette.

```
# Extract the supplementary file URLs from the targets object
tmp_targets <- targets$supplementary_file
# Extract the relevant part of the file names from the URLs,
starting from the 68th character to the end of the string
my_targets <- substr(tmp_targets,68,nchar(tmp_targets))
```

```
# Create a data frame with a single column named "Basename"
containing the extracted file names
my_targets2 <- data.frame("Basename"=my_targets)
# Read the methylation array data using the specified base
directory and targets data frame
RGset <- read.metharray.exp(
 base = out_dir, # Base directory where the raw data files are
located. If your file names EPIC contains another file namer
idat, in which contains idat. files, put: "~/Desktop/EPIC/
idat"
 targets = my_targets2, # Data frame containing the basenames
of the files to be read. Put only targets in you upload your
own data
 extended = TRUE, # Read extended format of the array data
 recursive = FALSE, # Do not search directories recursively
 verbose = FALSE, # Suppress verbose output
 force = TRUE # Force reading of the data even if some files
are missing
)

head(RGset)
```

2.6 Quality Control, Normalization, and Mapping

Poor-performing probes can obscure biological signals in the data and are typically filtered out before conducting differential methylation analysis. Since the signals from these probes are unreliable, removing them reduces the number of statistical tests performed, thereby lowering the multiple testing penalty. We filter out probes that have failed in one or more samples based on their detection *p* value. This ensures that only high-quality, reliable probes are used in the analysis, improving the accuracy and robustness of the results. After filtering, it is important to normalize the data to correct for technical variations and ensure comparability across samples. Various normalization methods, such as quantile normalization or functional normalization, can be applied depending on the specific characteristics of the dataset. Proper normalization is crucial for minimizing batch effects and other technical artifacts that could confound the biological interpretation of the data. Finally, quality control checks should be performed to assess the effectiveness of the filtering and normalization steps. Visualizations such as density plots, boxplots, and multidimensional scaling (MDS) plots can help to identify any remaining issues with the data and ensure that it is ready for downstream analysis. By rigorously preparing the data in this way, we can maximize the reliability and interpretability of the differential methylation analysis results.

```
detP <- detectionP(RGset)
failed <- detP > 0.01
```

```
keep <- colMeans(detP) < 0.05
RGset <- RGset[,keep]
targets <- targets[keep,]

rm(detP, failed, keep)
```

This function performs stratified quantile normalization preprocessing. The normalization procedure is applied separately to the methylated (Meth) and unmethylated (Unmeth) intensities. The type I and type II probe signals are aligned by first quantile normalizing the type II probes across samples, and then interpolating a reference distribution to normalize the type I probes. Because probe types and regions are confounded and DNA methylation (DNAm) distributions vary across regions, the probes are stratified by region before applying this interpolation. It is important to note that this algorithm relies on the assumptions necessary for quantile normalization and is not recommended for cases where global changes are expected, such as in cancer vs. normal comparisons. This normalization procedure is similar to one previously presented by Nizar Touleimat and Jörg Tost (2012). The different options for this function include:

- If fixMethOutlier is TRUE, the function corrects outliers in both the methylated and unmethylated channels when small intensities are close to zero.
- If removeBadSamples is TRUE, it removes poor quality samples using the previously discussed QC criteria.
- It performs stratified subset quantile normalization if quantileNormalize = TRUE and stratified = TRUE.
- It predicts the sex (if not provided in the sex argument) using the getSex function and normalizes males and females separately for the probes on the X and Y chromosomes.

```
GRset.quantile <- preprocessQuantile(RGset, fixOutliers =
TRUE,
 removeBadSamples = TRUE, badSampleCutoff = 10.5,
 quantileNormalize = TRUE, stratified = TRUE,
 mergeManifest = FALSE, sex = NULL)
```

In addition to filtering based on detection p values, it is also common to remove probes that map to multiple locations in the genome or overlap with known single nucleotide polymorphisms (SNPs) and sex chromosomes. These probes can introduce noise and potential biases into the analysis. By applying these additional filtering steps, we can further enhance the quality of the methylation data and increase the confidence in our findings.

```
keep <- !(featureNames(GRset.quantile) %in% annEPIC$Name[an-
nEPIC$chr %in%c("chrX","chrY")])
mSetSqFlt <- GRset.quantile[keep,]
mSetSqFlt <- dropLociWithSnps(mSetSqFlt)
rm(keep, GRset.quantile)
```

Another common metric for describing methylation levels is the *M* value, which is the log2 ratio of the intensities of the methylated probe to the unmethylated probe. An *M* value close to 0 indicates similar intensities between the methylated and unmethylated probes, suggesting the CpG site is approximately half-methylated, assuming the intensity data has been properly normalized. Positive *M* values indicate that more molecules are methylated than unmethylated, while negative *M* values indicate the opposite. While Beta- and *M* values are related, beta values are generally preferred for graphically representing methylation levels because percentage methylation has a more intuitive biological interpretation. However, due to their distributional properties, *M* values are more statistically valid for differential methylation analysis. A thorough comparison of both metrics can be found here.

```
mVals <- getM(mSetSqFlt)
bVals <- getBeta(mSetSqFlt)
```

3 Results

3.1 Differential Methylation Positions

After all this preprocessing and filtering, we can finally address the main biological question: which CpG sites are differentially methylated between the different cell types? To answer this, we will design a linear model using *limma*. We are interested in pairwise comparisons between the four cell types, accounting for variation between individuals. This analysis is conducted on the matrix of *M* values using limma, which provides t-statistics and associated *p* values for each CpG site. A convenient way to manage multiple comparisons is to use a contrasts matrix alongside the design matrix. The contrasts matrix allows for linear combinations of the design matrix columns corresponding to the comparisons of interest, effectively focusing the analysis on these comparisons. Next, these contrasts are fitted to the model, and the function eBayes is used to calculate the statistics and *p* values for differential expression. This function ranks genes based on the evidence for differential methylation. We would not go into detail about this statistical framework here, but more information can be found in the *limma* documentation. Using the topTable function in *limma*, you can extract differentially methylated genes for each comparison/contrast. To order these by *p* value, set sort.by = "p." The results for the first

comparison can be saved as a data.frame by setting coef = 1. The coef parameter explicitly refers to the column in the contrasts matrix that corresponds to the comparison of interest. Additionally, you can enhance the list of CpGs by including a genelist parameter in the topTable function. This helps retrieve the location of the CpG, the nearest gene or CpG island, and other relevant information.

```
library(limma)
# Create a new variable with groups if needed
targets$Sample_group <- ifelse(targets$source_name_ch1 ==
"normal PBMCs sample", "0", ifelse(targets$source_name_ch1
== "BC PBMCs sample", "1", NA))
table(targets$Sample_group)

# Here, you can add to the model such variables you consider to
adjust for
design <- model.matrix(~ targets$Sample_group, data=targets)

# Fit
fit <- lmFit(mVals, design)

# contrast.matrix <- makeContrasts(1-0,levels=design)
# fit2 <- contrasts.fit(fit, contrast.matrix)
fit2 <- eBayes(fit)

# Use a specific column for genome annotation
annEPICsub <- annEPIC[match(rownames(mVals),annEPIC$Name),
c(1:4,12:19,22:31,35:39:ncol(annEPIC))]

# Obtain Differentially methylated positions (DMP)
DMP <- topTable(fit2, num=Inf, coef=2, confint = TRUE,
genelist=annEPICsub)
DMP$UCSC_RefGene_Name <- sub(";.*", "", DMP$UCSC_RefGene_Name)

# Select the significant DMP
DMP <- subset(DMP, adj.P.Val < 0.05)

# Check the DMPs
sum(DMP$adj.P.Val < 0.05, na.rm=TRUE)
summary(decideTests(fit2))

# Write an Excel file with DMPs if needed
# install.packages("writexl")
# library(writexl)
# write_xlsx(DMP, "DMP.xlsx")
```

3.2 Differential Methylation Regions

Often, differential methylation at a single CpG site is not highly informative or can be difficult to detect. Therefore, identifying whether multiple nearby CpGs (or regions) are concordantly differentially methylated is often more insightful. Several Bioconductor packages provide functions for identifying differentially methylated regions (DMRs) from 450 k data. Some of the most popular options include the dmrFind function in the charm package, which has been somewhat replaced for 450 k arrays by the bumphunter function in minfi, and the dmrcate function in the DMRcate package. Each is based on different statistical methods, but we will use dmrcate here because it is based on limma, allowing us to use the design and contrast matrix we defined earlier. We will start again with our matrix of *M* values. For this type of analysis, the matrix must be annotated with the chromosomal positions of the CpGs and their gene annotations. Since the initial step involves running the limma differential methylation analysis for single CpGs, we need to specify the design matrix, contrast matrix, and the contrast of interest.

```
library(DMRcate)
myannotation <- DMRcate::cpg.annotate(object = mVals,
 datatype = "array",
 what = "M",
 analysis.type = "differential",
 design = design,
 coef = 2,
 arraytype = "EPIC")
str(myannotation)

# Extract info
DMR <- dmrcate(myannotation, lambda=1000, C=2)
results.ranges <- extractRanges(DMR, genome = "hg19")
head(results.ranges)
results.ranges <- data.frame(results.ranges)

# Write an Excel file with DMPs if needed
# write_xlsx(x = results.ranges1, path = "DMR1.xlsx", col_-
names = TRUE)
```

3.3 Gene Ontology

An alternative method to detect DMRs involves predefining the regions to be tested. Unlike the previous approach, which defines regions based on heuristic distance rules, this method defines regions based on shared functions. We will use the mCSEA package, which includes three types of regions for 450K and EPIC arrays: promoter regions, gene bodies, and CpG Islands. mCSEA is based on gene set enrichment analysis (GSEA), a popular methodology for functional analysis designed to address certain

limitations in the field of gene expression. In brief, CpG sites are ranked according to a metric (such as logFC or t-statistic), and an enrichment score (ES) is calculated for each region. This is done by traversing the entire ranked list of CpG sites, increasing the score when a CpG in the region is encountered and decreasing it when a CpG outside the region is encountered. A high ES indicates that the probes are found near the top of the ranked list, suggesting that the CpG sites in this region, on average, exhibit a higher methylation level. This approach is particularly effective for detecting smaller but consistent methylation differences.

In this case, we will apply this method to the output of the "naive-rTreg" comparison, ranking the CpGs by logFC differences. We will specify "promoters" as the type of region to be considered, although other options like CpG Islands or gene bodies can also be used. After obtaining a potentially extensive list of significantly differentially methylated CpG sites, one might question whether specific biological pathways are overrepresented in this list. In some cases, it is straightforward to link the top differentially methylated CpGs to genes that are biologically relevant to the cell types or samples being studied. However, with potentially thousands of significantly differentially methylated CpGs, gene set analysis (GSA) can be used to uncover meaningful biological patterns from these high-throughput data. The objective is typically to identify commonalities among the genes, using annotations from sources such as the Gene Ontology (GO) or Kyoto Encyclopedia of Genes and Genomes (KEGG).

This type of analysis can be performed using the gometh function in the missMethyl package. This function requires a character vector of the names (e.g., cg20832020) of the significant CpG sites and optionally, a character vector of all CpGs tested. Including all tested CpGs is recommended, especially if extensive filtering was done before the analysis, as it serves as the "background" from which any significant CpG could be chosen. For gene ontology testing, the user can set collection = "GO" (the default option). For testing KEGG pathways, collection = "KEGG" should be specified. In this tutorial, we will proceed with the results from the single-probe "naive vs rTreg" comparison and select all CpG sites with an adjusted p value of less than 0.05.

```
library(missMethyl)
# Get the significant CpG sites at less than 5% FDR
sigCpGs <- DMP$Name[DMP$P.Value<0.05]

# Get all the CpG sites used in the analysis to form the
background
all <- DMP$Name

# Run Gene Ontology Analysis, this may take a while
```

```
GO <- gometh(sig.cpg=sigCpGs,
 all.cpg=all,
 collection = c("GO"),
 array.type = c("EPIC"),
 plot.bias = FALSE,
 equiv.cpg = TRUE,
 anno = annEPIC,
 sig.genes = TRUE)

# Run Gene Set Enrichment Analysis
# Add ENTREZID reference
library(org.Hs.eg.db)
library(clusterProfiler)

DMP$entrez <- mapIds(org.Hs.eg.db,
 keys=DMP$UCSC_RefGene_Name,
 column="ENTREZID",
 keytype="SYMBOL",
 multiVals="first")

# Select genes
original_gene_list <- DMP$logFC
names(original_gene_list) <- DMP$entrez
gene_list <- na.omit(original_gene_list)
gene_list <-sort(gene_list, decreasing = TRUE)
table(duplicated(gene_list))

# Run
organism = "org.Hs.eg.db"
gse <- gseGO(geneList=gene_list,
 ont ="ALL",
 keyType = "ENTREZID",
 nPerm = 10000,
 minGSSize = 3,
 maxGSSize = 800,
 pvalueCutoff = 0.05,
 verbose = TRUE,
 OrgDb = organism,
 pAdjustMethod = "none")

# Run KEGG Analysis
KEGG <- gometh(sig.cpg=sigCpGs,
 all.cpg=all,
 collection = c("KEGG"),
 array.type = c("EPIC"),
 plot.bias = FALSE,
 equiv.cpg = TRUE,
 anno = annEPIC,
 sig.genes = TRUE)
```

3.4 Immune Quantification

Peripheral blood is commonly used for DNA methylation analyses due to its easy accessibility through minimally invasive procedures. Emerging evidence suggests that specific DNA methylation changes in blood may reflect pathological states in target organs that are not easily accessible by biopsy. Blood DNA methylation profiles can also capture information on systemic exposures or diseases where single organ assessment is not feasible. Epigenome-wide association studies (EWAS) have shown that some DNA methylation changes reflect induced epigenetic alterations within blood cells, while others indicate changes in leukocyte subtype proportions related to pathophysiology. To address cell heterogeneity and potential confounding, both reference-based and non-reference-based techniques are employed, with applications detailed in previous studies. Deconvolution techniques, such as constrained projection/quadratic programming (CP/QP), estimate the relative proportions of blood cell types using DNA methylation signatures.

Initially pipelines for estimating leukocyte subtypes in adult blood were based on six adult male samples purified by flow cytometry and profiled using the Illumina HumanMethylation450K array (450K array). With the advent of the Illumina HumanMethylationEPIC array (EPIC array), which interrogates over 860,000 CpG sites, there is a need to assess the accuracy of cell deconvolution using existing 450 K reference signatures. The EPIC array includes additional genomic content in enhancer regions and DNase hypersensitive sites (DHS), crucial for hematopoietic development and differentiation.

The FlowSorted.Blood.EPIC package extends the reference library for blood cell proportion deconvolution using the EPIC array, aiming to improve the accuracy of cell composition estimates and address potential platform differences. DNA methylation was measured using the EPIC array on neutrophils, B cells, monocytes, NK cells, CD4+ T cells, and CD8+ T cells sorted with antibody beads. An iterative algorithm for Identifying Optimal Libraries (IDOL) from leukocyte differentially methylated regions (L-DMR) was applied to enhance the accuracy and efficiency of cell composition estimates obtained through cell mixture deconvolution. The package contains Illumina HumanMethylationEPIC (EPIC) DNA methylation microarray data from immunomagnetic sorted adult blood cell populations. This dataset, includes 37 magnetically sorted blood cell references and 12 additional samples. The data is formatted as an RGChannelSet object, which allows for seamless integration and normalization using most existing Bioconductor packages.

The code provided below is designed to estimate the proportions of various immune cell types in a given methylation dataset. This process involves several steps. First, the reference data is loaded, including the FlowSorted.Blood.EPIC dataset, which

provides reference data for different blood cell types, and the IDOLOptimizedCpGs dataset, which contains optimized CpG probes for accurate cell type estimation. The target object, which includes sample sheet information, is utilized to extract relevant file names of the raw data files. A data frame named my_targets2 is created to hold these file names. The read.metharray.exp. function then reads the methylation data files specified in the my_targets2 data frame and stores the data in an object called RGset.

For estimating cell counts, the estimateCellCounts2 function is used with RGset as the input. This function estimates the proportions of different immune cell types by utilizing parameters such as the Noob preprocessing method (preprocessNoob), the IDOL optimization method for probe selection, and specifying cell types including CD8+ T cells, CD4+ T cells, NK cells, B cells, monocytes, and neutrophils. The FlowSorted.Blood.EPIC dataset is used as the reference set, and IDOLOptimizedCpGs are used for probe selection. The cell proportions are converted to percentages, rounding them to one decimal place for better readability. This framework provides a robust method for estimating immune cell proportions in blood samples using DNA methylation data.

```
library(FlowSorted.Blood.EPIC)

# Define the function below to retrieve the reference data from
ExperimentHub
libraryDataGet <- function(title) {
assign(title, ExperimentHub()[[query(
ExperimentHub(),
 title
 )$ah_id]])
}

# Load the FlowSorted.Blood.EPIC dataset, which provides
reference data for blood cell types
FlowSorted.Blood.EPIC <- libraryDataGet('FlowSorted.Blood.
EPIC')
# Load the IDOLOptimizedCpGs data, which contains optimized
CpG probes for cell type estimation
data("IDOLOptimizedCpGs")

# Estimate cell proportions in the given RGset using the
specified parameters
percEPIC <- estimateCellCounts2(
 RGset, # The methylation data set to be analyzed
  compositeCellType = "Blood", # Specify the composite cell
type as blood
 processMethod = "preprocessNoob", # Use the Noob preproces-
sing method
```

```
 probeSelect = "IDOL", # Select probes based on the IDOL
optimization method
 cellTypes = c("CD8T", "CD4T", "NK", "Bcell", "Mono", "Neu"),
 referencePlatform = "IlluminaHumanMethylationEPIC",
 referenceset = "FlowSorted.Blood.EPIC", # Use as reference
the FlowSorted.Blood.EPIC
 IDOLOptimizedCpGs = IDOLOptimizedCpGs # Provide the IDOL
optimized CpG probes
)

# Print the first few rows of the estimated cell proportions
print(head(percEPIC$counts))

# Convert the cell proportions to percentages and round to one
decimal place
percEPIC <- data.frame(round(percEPIC$prop * 100, 1))
```

4 Visualization

4.1 PCA

Principal Component Analysis (PCA) is a valuable tool for analyzing DNA methylation data from cancer and control patients, as it reduces the dimensionality of the dataset while retaining the most significant variation, thereby facilitating the identification of distinct methylation patterns that may differentiate cancerous tissues from normal tissues. In this case, we create a PCA from beta values, to use them as a valuable tool to separate two groups by DNA methylation (Fig. 2).

```
targets$Group <- ifelse(targets$source_name_ch1 == "normal
PBMCs sample", "Control",
ifelse(targets$source_name_ch1 == "BC PBMCs sample", "Can-
cer", NA))

# Create a data.frame for beta values to represent PCA
bVals_t <- t(bVals)
bVals_t <- data.frame(bVals_t)
pca_res <- prcomp(bVals_t, scale. = TRUE)

library(ggfortify)
autoplot(pca_res,
 data = targets,
 colour = 'Group',
 size = 5,
 addEllipses = TRUE,
 frame = TRUE, frame.type = 't')+
theme(title = element_text(size = 20))+
```

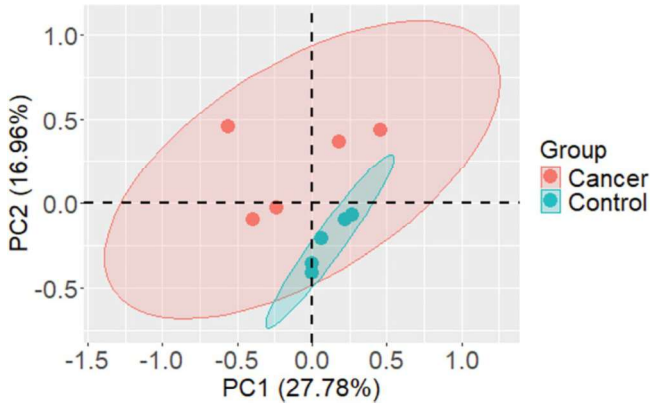

Fig. 2 PCA analysis using DNA methylation data. Tumor samples are in red circles and normal samples are in green circles

```
theme(axis.text = element_text(size = 20))+
theme(legend.text = element_text(size = 20))+
geom_hline(yintercept=0, linetype="dashed", color = "black",
size = 0.8) +
geom_vline(xintercept=0, linetype="dashed", colour= "black",
size = 0.8)
```

4.2 Volcano Plot

A volcano plot is a powerful visualization tool in the analysis of DNA methylation data from cancer and control patients. It allows researchers to quickly identify statistically significant differences in methylation levels between the two groups. By plotting the magnitude of change (e.g., fold change in methylation levels) on the X-axis and the statistical significance (e.g., p value) on the Y-axis, the volcano plot highlights individual CpG sites or regions that exhibit both large changes in methylation and strong statistical significance. This dual-axis approach helps in pinpointing potential epigenetic markers for cancer diagnosis or therapeutic targets, making the volcano plot an invaluable tool in cancer epigenetics research (Fig. 3).

```
library(tibble)
library(ggrepel)
DMP_adjusted <- subset(DMP, DMP$adj.P.Val <= 0.1)
DMP_adjusted <- DMP_adjusted %>%
mutate(
 Expression = case_when(logFC >= 0 & adj.P.Val <= 0.05 ~ "Up-
regulated",
 logFC <= 0 & adj.P.Val <= 0.05 ~ "Down-regulated",
 TRUE ~ "Unchanged")
 )
```

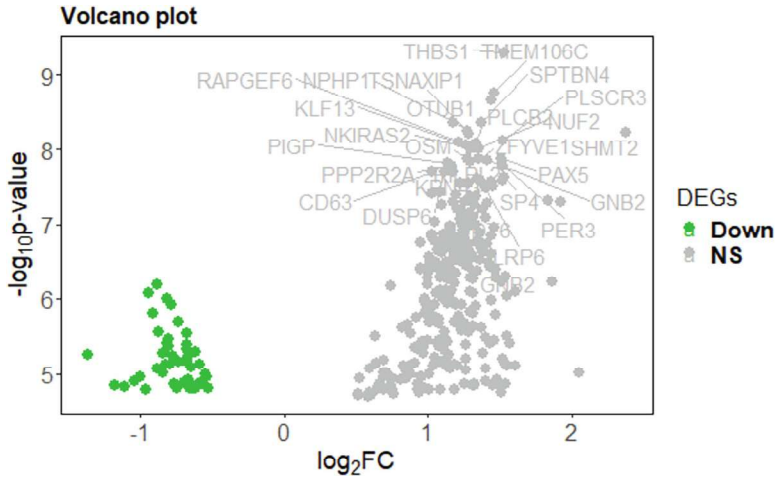

Fig. 3 Volcano plot for breast cancer vs normal, where the adjusted *p* value is less than 0.05

```
DMP_adjusted <- rownames_to_column(DMP_adjusted, var = "Header")

# Add a column to the data frame to specify if they are UP- or DOWN- regulated (log2fc respectively positive or negative)
DMP_adjusted$diffexpressed <- "NO"
# if log2Foldchange > 0.6 and pvalue < 0.05, set as "UP"
DMP_adjusted$diffexpressed[DMP_adjusted$logFC > 0 & DMP_adjusted$P.Value < 0.05] <- "UP"
# if log2Foldchange < -0.6 and pvalue < 0.05, set as "DOWN"
DMP_adjusted$diffexpressed[DMP_adjusted$logFC < 0 & DMP_adjusted$P.Value < 0.05] <- "DOWN"
# Explore a bit
head(DMP_adjusted[order(DMP_adjusted$P.Value) & DMP_adjusted$diffexpressed == 'DOWN', ])

# Create a new column "delabel" to de, that will contain the name of the top 30 differentially expressed genes (NA in case they are not)
DMP_adjusted$delabel1 <- ifelse(DMP_adjusted$UCSC_RefGene_Name %in% head(DMP_adjusted[order(DMP_adjusted$P.Value), "UCSC_RefGene_Name"], 30), DMP_adjusted$UCSC_RefGene_Name, NA)
DMP_adjusted$delabel2 <- ifelse(DMP_adjusted$Name %in% head(DMP_adjusted[order(DMP_adjusted$P.Value), "Name"], 30), DMP_adjusted$Name, NA)

ggplot(data = DMP_adjusted, aes(x = logFC, y = -log10(P.Value), col = Expression, label = delabel1)) +
```

```
theme_classic() +
# geom_vline(xintercept = c(-0.6, 0.6), col = "gray", linetype
= 'dashed') +
# geom_hline(yintercept = -log10(0.05), col = "gray", linetype
= 'dashed') +
geom_point(size = 3) +
scale_color_manual(values = c("green3", "grey", "red3"), # to
set the colours of our variable
 labels = c("Down", "NS", "Up")) +
labs(color = 'DEGs', #legend_title,
 x = expression("log"[2]*"FC"), y = expression("-log"[10]*"p-
value")) +
ggtitle('Volcano plot') +
geom_text_repel(max.overlaps = Inf, size = 5, face= "bold") +
theme(axis.text = element_text(size = 15)) +
theme(legend.text = element_text(size = 15, face= "bold")) +
theme(axis.title.x = element_text(size = 17, face = "bold")) +
theme(axis.title.y = element_text(size = 17, face = "bold")) +
theme(plot.title = element_text(size = 15, face = "bold")) +
theme(legend.title = element_text(size = 15)) +
theme(panel.border = element_rect(colour = "black", fill =
NA, size= 0.5),
 panel.grid.minor = element_blank(),
 panel.grid.major = element_blank())
```

4.3 Heatmap

Heatmaps are a crucial tool for visualizing DNA methylation data, as they provide an intuitive representation of methylation levels across numerous genomic regions and samples. This visualization technique can effectively highlight differences between cancer and control patients, making it easier to identify regions of the genome that are differentially methylated. By displaying complex data in a color-coded format, heatmaps facilitate the recognition of patterns and correlations that might be missed in traditional data tables. This ability to visually compare the methylation status across samples enables researchers to pinpoint specific genes or regions that could be involved in cancer development or progression, aiding in the discovery of potential biomarkers and therapeutic targets.

Moreover, heatmaps can be customized with annotations and clustering to further enhance data interpretation. For example, adding sample annotations for clinical variables or treatment groups can provide additional layers of insight, revealing how methylation patterns are associated with different clinical outcomes. The hierarchical clustering often applied in heatmap analyses groups similar samples and genomic regions together, which can uncover previously unrecognized subgroups within the data. This can lead to the identification of novel cancer subtypes or the

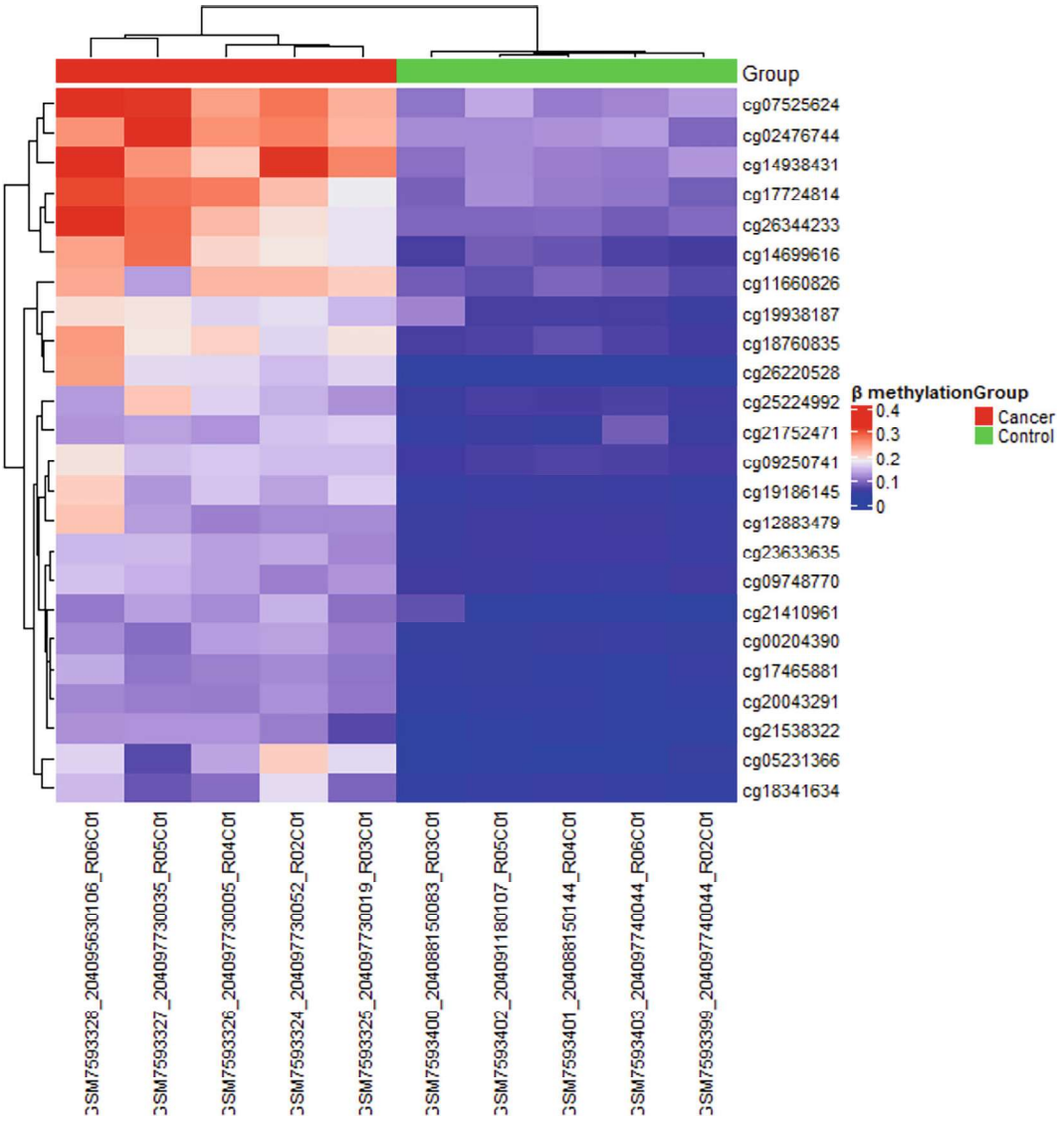

Fig. 4 Heatmap of DNA methylation positions, previously selected by adjusted *p* value less than 0.05 and an absolute log Fold Change greater than 1.5

elucidation of mechanisms underlying disease heterogeneity. In the context of integrative analyses, heatmaps can be used alongside other data types, such as gene expression or genetic mutations, to provide a comprehensive view of the epigenetic landscape and its interactions with other molecular alterations in cancer (Fig. 4).

```
library(ComplexHeatmap )
# Create data.frame for significant CpGs
DMP_subset <- subset(DMP, adj.P.Val<0.05 & abs(DMP$logFC) >
1.5)
```

```
idx = rownames(DMP_subset)

# Crear matrix of beta values from significant DMPs
bVals_significant <- data.frame(bVals[idx,])
bVals_significant <- as.matrix(bVals_significant)

# Create color of subsetting
col = list(Group = c("Control" = "green", "Cancer" = "red"))

# Create the heatmap annotation
ha <- HeatmapAnnotation(Group = as.factor(targets$Group), col
= col)

# Combine the heatmap and the annotation
library(ComplexHeatmap)
Heatmap(bVals_significant, name = "β methylation",
 top_annotation = ha,
 row_names_gp = gpar(fontsize = 10),
 column_names_gp = gpar(fontsize = 10))
```

4.4 GSEA and ORA

Gene Ontology (GO) and Kyoto Encyclopedia of Genes and Genomes (KEGG) analyses are indispensable tools in the realm of bioinformatics, providing valuable insights into the functional significance of genes and their involvement in various biological processes. These analyses offer a structured framework for annotating genes based on their molecular functions, biological processes, and cellular components, thereby unraveling the intricate interplay of genes within living organisms. The significance of results derived from GO and KEGG analyses lies in their ability to elucidate the underlying biological mechanisms driving complex biological phenomena. By associating genes with specific biological functions and pathways, these analyses facilitate the interpretation of high-throughput genomic data and enable researchers to discern meaningful patterns amidst vast datasets. This comprehension extends beyond individual genes, offering a holistic understanding of biological systems and their regulatory networks.

Furthermore, GO and KEGG analyses play a pivotal role in hypothesis generation and validation, guiding experimental studies aimed at deciphering the molecular basis of diseases, identifying therapeutic targets, and unraveling the intricacies of physiological processes. By pinpointing key pathways and biological functions enriched with relevant genes, these analyses provide valuable leads for further investigation, ultimately advancing our knowledge of disease mechanisms and therapeutic interventions. Moreover, the integration of GO and KEGG analyses with other omics data, such as transcriptomics, proteomics, and metabolomics, enables

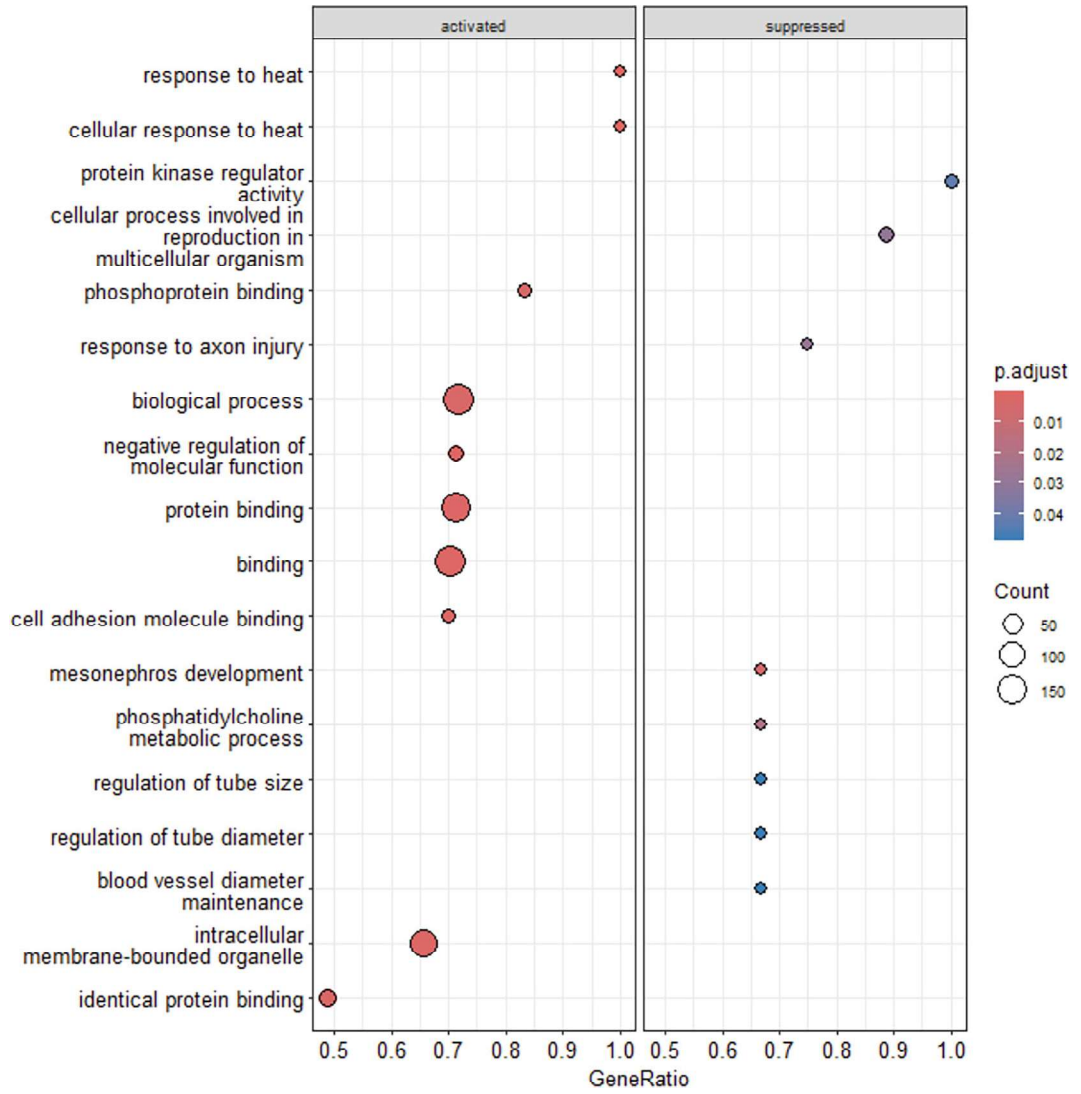

Fig. 5 Gene set enrichment analysis showing pathways activated and suppressed in cancer, by using DNA methylation analysis

comprehensive multi-omics investigations, facilitating a systems-level understanding of biological phenomena. This integrative approach fosters cross-disciplinary collaborations and accelerates discoveries in fields ranging from basic research to clinical applications. In conclusion, the results obtained from GO and KEGG analyses serve as a cornerstone in biological research, offering valuable insights into gene function, pathway regulation, and disease mechanisms. Their significance extends beyond mere data annotation, driving innovation and discovery in diverse areas of biomedical research.

```
library(multienrichjam)
library(enrichplot)

# GO
enr1 <- enrichDF2enrichResult(enrichDF = GO, keyColname =
"ONTOLOGY", geneColname = "SigGenesInSet", pvalueColname =
"P.DE", descriptionColname = "TERM", pvalueCutoff = 0.05)
 edox1 <- pairwise_termsim(enr1)
barplot(edox1, showCategory=30)

# KEGG
enr2 <- enrichDF2enrichResult(enrichDF = KEGG, keyColname =
"N", geneColname = "SigGenesInSet", pvalueColname = "P.DE",
descriptionColname = "Description", pvalueCutoff = 0.05)
edox2 <- pairwise_termsim(enr2)
barplot(edox2, showCategory=30)
```

The provided code utilizes the dotplot function from the enrichplot package to visualize the results of a gene enrichment analysis. Here's the detailed description of the code: dotplot(gse, showCategory = 10, split = ".sign"): This line generates a dot plot using the results of the gene enrichment analysis stored in the gse object. The showCategory parameter is set to 10, indicating that the top 10 categories will be displayed in the plot. The split parameter is set to ".sign", which splits the results into two panels based on the sign of the enrichment score. + facet_grid(. ~ .sign): This part of the code adds the facet to the plot, splitting the dots into two panels based on the sign of the enrichment score. In summary, this code generates a dot plot that displays the top categories of a gene enrichment analysis, with the results split into two panels based on the sign of the enrichment score. This provides a useful visualization of the gene enrichment analysis results, allowing for comparison of categories based on their enrichment sign (Fig. 5).

```
dotplot(gse, showCategory=10, split=".sign") + facet_grid(.~.
sign)
```

4.5 Immune Representation

For immune representation using estimateCellCounts function, you typically would not generate a dot plot directly from that function alone, as it returns estimated cell counts for various immune cell types. Instead, you might visualize the results in a bar plot or heatmap to represent the estimated counts of different immune cell types across samples. Customize the plot appearance, axis labels, titles, colors, and any other relevant aspects to make the visualization informative and visually appealing. Finally, interpret the generated plot to gain insights into the distribution of immune

cell types across your samples, and how they may vary under different conditions or experimental groups (Fig. 6).

```
library(tidyr)
library(ggpubr)
# Create a barplot
data_long <- percEPIC %>%
pivot_longer(cols = c(CD8T, CD4T, NK, Bcell, Mono, Neu),
names_to = "variable", values_to = "valor")

dodge <- position_dodge(width = 0.9)
limits <- aes(ymax = mean + SD, ymin = mean)

ggplot(data_long) +
geom_boxplot(aes(x = as.factor(Group), y = valor, fill = as.
factor(Group))) +
facet_wrap(~ variable, scales = "free_y") +
```

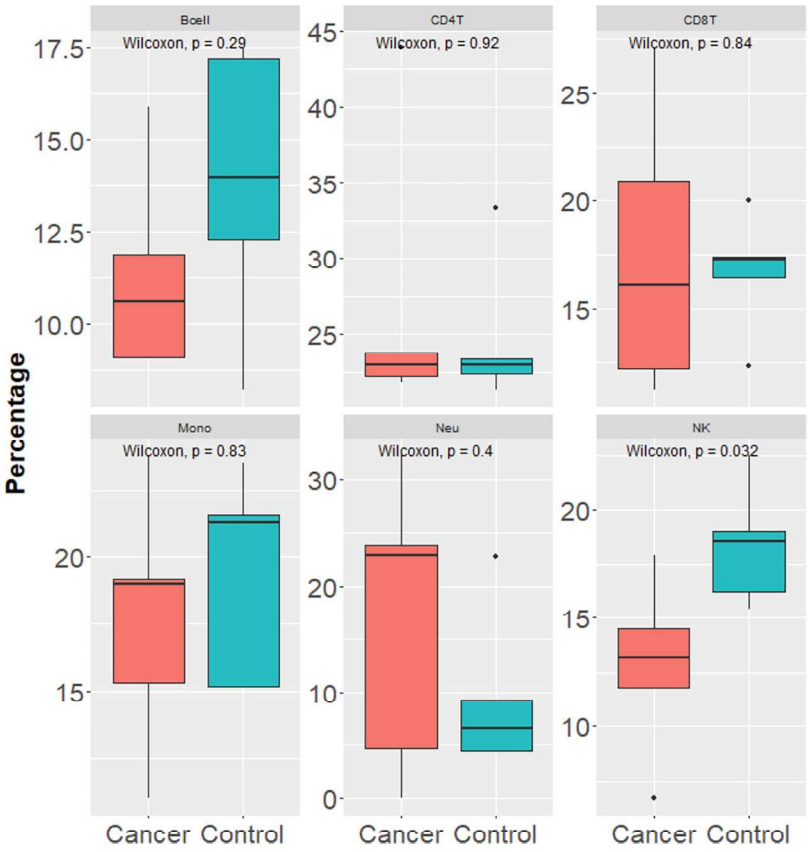

Fig. 6 Boxplot of DNA methylation deconvolution of immune cells located in the blood, including T cells, B cells, natural killers (NK), monocytes (mono), and neutrophils (neu). Wilcoxon test was used to calculate significant differences between groups

```
theme(legend.position = "none", text = element_text(size =
12)) +
labs(x = NULL, y = "Percentage") +
theme(axis.title = element_text(size = 20, face = "bold")) +
theme(axis.text = element_text(size = 20)) +
theme(legend.text = element_text(size = 20)) +
labs(color = "Participants") +
theme(legend.title = element_text(size = 20))+
scale_fill_manual(values = c("Cancer" = "#F8766D", "Control"
= "#00BFC4")) +
stat_compare_means(aes(x = as.factor(Group), y = valor, group
= as.factor(Group)))
```

5 Limitations

The RStudio workflow currently focuses on analyzing epigenetic data and estimating cell-type proportions specifically in blood samples. The current implementation of the pipeline has not been tested on tissue samples. However, as advancements in deconvolution methods continue and extended reference libraries are developed, the potential for its application broadens. Reference libraries containing purified cell subtypes, such as epithelial, mesenchymal, and progenitor cells, make deconvolution feasible also in solid tissue samples (PMID: 28977446). Thus, our pipeline could serve as a foundational framework, which with some fine-tuning, could be adapted to facilitate tissue deconvolution. Such efforts could integrate our workflow with tools like EpiSCORE (PMID: 32883324) to deconvolute bulk tissue samples of DNA methylomes. The working example of the workflow utilizes IDOL optimization algorithm (PMID: 26956433) and the FlowSorted. Blood.EPIC package which contains Illumina HumanMethylationEPIC DNA methylation microarray data (PMID: 29843789). This dataset includes information for the following six cell populations: T lymphocytes (CD4+ and CD8+), B lymphocytes (CD19+), monocytes (CD14+), natural killer (NK) cells (CD56+) and Neutrophils (Neu). Consequently, our pipeline is limited by the available cell types. To extend the analysis to include additional cell populations, users can utilize the extended version of the FlowSorted.Blood.EPIC package (PMID: 35140201). The extended version supports a total of 12 different cell types: neutrophils (Neu), eosinophils (Eos), basophils (Bas), monocytes (Mono),B lymphocytes naive (Bnv), B lymphocytes memory (Bmem), T helper lymphocytes naive (CD4nv), T helper lymphocytes memory (CD4mem), T regulatory cells (Treg), T cytotoxic lymphocytes naive (CD8nv), T cytotoxic lymphocytes memory (CD8mem),

and natural killer lymphocytes (NK). The usage of this package is restricted to research purposes and requires an academic license. Instructions for obtaining the license are provided in the readme file on their Github repository (https://github.com/immunomethylomics/FlowSorted.BloodExtended.EPIC).

The pipeline code has been tested for reproducibility on three major platforms: Windows, macOS, and Linux. An issue was found with the gene ontology analysis using KEGG, which produces an error on Linux. The entire pipeline relies on Bioconductor version 3.14 and has been tested with R versions 4.0.0 and 4.1.0. Users should be able to reproduce the pipeline using these versions. Additionally, it is expected that the pipeline can be reproduced with the latest version of R. Because the workflow depends on multiple packages, it is advised to set up an R project in an isolated environment using a package manager like "renv" to ensure reproducibility and minimize compatibility issues with package dependencies.

6 Notes

Code availability: https://github.com/SotirisOuzounis/ImmunoMethylation

Acknowledgments

This publication is based upon work from COST Action IMMUNO-model, CA21135, supported by COST (European Cooperation in Science and Technology). This study was supported by "Centro de Investigacion Biomédica en Red Fisiopatología de la Obesidad y Nutricion", which is an initiative of the "Instituto de Salud Carlos III" (ISCIII) of Spain, financed by the European Regional Development Fund, "A way to make Europe"/"Investing in your future" (CB06/03), a grant from ISCIII (PI18/01399, PI21/00633) and a grant from Consejeria Universidad, Investigacion e Innovacion Junta de Andalucia (PY20-01270). HB is supported by a "Sara Borrell" postdoctoral contract (CD22/00053) from the Instituto de Salud Carlos III—Madrid (Spain), "Financiado por la Unión Europea—NextGenerationEU" y mediante el Plan de Recuperación, Transformación y Resiliencia. HB also received funding from the project 'PI24/00061', funded by the 'Instituto de Salud Carlos III (ISCIII)' and co-funded by the European Union. This work has been supported by Fondazione Michelangelo (Milan, Italy) and Breast Cancer Research Foundation (grant BCRF 21-181) from MC.

References

1. Esteller M (2008) Epigenetics in cancer. N Engl J Med 358(11):1148–1159. https://doi.org/10.1056/nejmra072067
2. Izquierdo AG, Boughanem H, Diaz-Lagares A, Arranz-Salas I, Esteller M, Tinahones FJ et al (2022) DNA methylome in visceral adipose tissue can discriminate patients with and without colorectal cancer. Epigenetics 17:665–676
3. Tien FM, Lu HH, Lin SY, Tsai HC (2023) Epigenetic remodeling of the immune landscape in cancer: therapeutic hurdles and opportunities. J Biomed Sci 30(1):1–23. https://doi.org/10.1186/s12929-022-00893-0
4. Wang T, Li P, Qi Q, Zhang S, Xie Y, Wang J et al (2023) A multiplex blood-based assay targeting DNA methylation in PBMCs enables early detection of breast cancer. Nat Commun 14(1):4724. https://pubmed.ncbi.nlm.nih.gov/37550304/
5. Aryee MJ, Jaffe AE, Corrada-Bravo H, Ladd-Acosta C, Feinberg AP, Hansen KD et al (2014) Minfi: a flexible and comprehensive Bioconductor package for the analysis of Infinium DNA methylation microarrays. Bioinformatics 30:1363
6. Salas LA, Koestler DC, Butler RA, Hansen HM, Wiencke JK, Kelsey KT et al (2018) An optimized library for reference-based deconvolution of whole-blood biospecimens assayed using the Illumina HumanMethylationEPIC BeadArray. Genome Biol 19(1):64. Available from: /pmc/articles/PMC5975716/