Using TENET

Rhie Lab at the University of Southern California

2024-11-27

Abstract

TENET identifies key transcription factors (TFs) and regulatory elements (REs) linked to a specific cell type by finding significantly correlated differences in gene expression and RE methylation between case and control input datasets, and identifying the top genes by number of significant RE DNA methylation site links. It also includes many additional tools to aid in visualization and analysis of the results, including plots displaying and comparing methylation and expression data and RE DNA methylation site link counts, survival analysis, TF motif searching in the vicinity of linked RE DNA methylation sites, custom TAD and peak overlap analysis, and UCSC Genome Browser track file generation. A utility function is also provided to download methylation, expression, and patient survival data from The Cancer Genome Atlas (TCGA) for use in TENET or other analyses.

Contents

Introduction

There is a lack of publicly available bioinformatic tools to identify transcription factors (TFs) that regulate cell type-specific regulatory elements (REs). To address this, we developed the Tracing regulatory Element Networks using Epigenetic Traits (TENET) R package. TENET uses histone mark and open chromatin datasets, along with matched DNA methylation and gene expression data, to identify dysregulated REs and the TFs bound to them in a particular cell or tissue type in comparison with another. To assist in identifying TFs and REs linked to a particular cell type, we collected hundreds of epigenomic datasets from a variety of cell lines, primary cells, and tissues and developed methods to interrogate findings using motif databases, clinical information, and other genomic datasets from 10 cancer types. Additionally, many downstream analysis functions have been included to aid users in analyzing and visualizing results generated by the TENET workflow.

This vignette provides basic instructions to run the workflow of TENET to identify key TFs and linked RE DNA methylation sites. We will cover how to install the package, an overview of the necessary input data to use functions included in the TENET package, and the use of the TENET step 1-7 functions and the `TCGADownloader` utility function.

Acquiring and installing TENET and associated packages

To use TENET, users will need to install the base package as well as its associated example experiment data package, TENET.ExperimentHub. **Note:** TENET.ExperimentHub will install automatically when TENET is installed from Bioconductor.

TENET also uses annotation datasets hosted in the Bioconductor AnnotationHub database. These datasets will be automatically loaded from AnnotationHub when necessary. They are also available separately via the TENET.AnnotationHub package. It is not necessary to install the TENET.AnnotationHub package to use TENET's functions.

R 4.4 or a newer version is required.

On Ubuntu 22.04, installation was successful in a fresh R environment after adding the official R Ubuntu repository using the instructions at https://cran.r-project.org/bin/linux/ubuntu/ and running:

```
sudo apt-get install r-base-core r-base-dev libcurl4-openssl-dev libfreetype6-dev
libfribidi-dev libfontconfig1-dev libharfbuzz-dev libtiff5-dev libxml2-dev
```

No dependencies other than R are required on macOS or Windows.

Two versions of this package are available.

To install the stable version from Bioconductor, start R and run:

```r
## Install BiocManager, which is required to install packages from Bioconductor
if (!requireNamespace("BiocManager", quietly = TRUE)) {
    install.packages("BiocManager")
}

BiocManager::install(version = "devel")
BiocManager::install("TENET")
```

The development version containing the most recent updates is available from our GitHub repository (https://github.com/rhielab/TENET).

To install the development version from GitHub, start R and run:

```r
## Install prerequisite packages to install the development version from GitHub
if (!requireNamespace("BiocManager", quietly = TRUE)) {
    install.packages("BiocManager")
}
if (!requireNamespace("remotes", quietly = TRUE)) {
    install.packages("remotes")
}

BiocManager::install(version = "devel")
BiocManager::install("rhielab/TENET.ExperimentHub")
BiocManager::install("rhielab/TENET")
```

Loading TENET

To load the TENET package, start R and run:

```
library(TENET)
```

To load the TENET.ExperimentHub package, start R and run:

```
library(TENET.ExperimentHub)
```

To load the TENET.AnnotationHub package if it has been installed, start R and run:

```
library(TENET.AnnotationHub)
```

Running TENET without internet access

Some TENET features and examples download datasets from the internet if they have not already been cached. You must run `TENETCacheAllData()` once while connected to the internet before using these TENET features or examples without internet access (for example, on HPC cluster nodes). See the documentation for `TENETCacheAllData` for more information.

Input data

TENET primarily makes use of a MultiAssayExperiment object containing the following data:

Expression SummarizedExperiment object

A SummarizedExperiment object named "expression" containing gene expression data for genes in the human genome. Although gene expression values can be given in a variety of forms, TENET has been primarily tested using log2-transformed, upper-quartile normalized fragments per kilobase of transcript per million mapped reads (FPKM-UQ) values. Gene expression values for each gene should be given in the rows, while expression values from each sample should be included in the columns of the assay object in the "expression" SummarizedExperiment object. Additionally, IDs for each gene should be included in the rownames of this object and sample IDs should be included in the colnames.

The samples within the "expression" SummarizedExperiment object should be matched with those in the "methylation" SummarizedExperiment object.

```
## Load the MultiAssayExperiment package. This is not strictly necessary, but
## allows the user to avoid typing MultiAssayExperiment:: before its functions.
library(MultiAssayExperiment)
```

```
## Load in the example MultiAssayExperiment dataset from the TENET.ExperimentHub
## package
exampleTENETMultiAssayExperiment <-
    TENET.ExperimentHub::exampleTENETMultiAssayExperiment()
#> see ?TENET.ExperimentHub and browseVignettes('TENET.ExperimentHub') for documentation
#> loading from cache

## Examine the SummarizedExperiments that should be contained in a
## MultiAssayExperiment object appropriate for use in TENET analyses, including
## the "expression" object
experiments(exampleTENETMultiAssayExperiment)
#> ExperimentList class object of length 2:
#>  [1] expression: RangedSummarizedExperiment with 11637 rows and 242 columns
#>  [2] methylation: RangedSummarizedExperiment with 20000 rows and 242 columns

class(
    experiments(
        exampleTENETMultiAssayExperiment
    )[["expression"]]
)
#> [1] "RangedSummarizedExperiment"
#> attr(,"package")
#> [1] "SummarizedExperiment"

## Examine data for the first 6 genes and samples in the assay object of the
## "expression" SummarizedExperiment object
assays(
    experiments(
        exampleTENETMultiAssayExperiment
    )[["expression"]]
)[[1]][
    seq_len(6), seq_len(6)
]
#>                 TCGA-5L-AAT0-01A-12R-A41B-07 TCGA-A1-A0SK-01A-12R-A084-07
#> ENSG00000000457                     15.76383                     15.33703
#> ENSG00000001036                     17.74616                     17.52045
#> ENSG00000001167                     17.71227                     18.51419
#> ENSG00000001461                     16.54327                     17.14562
#> ENSG00000001630                     13.15326                     12.82170
#> ENSG00000002016                     15.08772                     17.33838
#>                 TCGA-A2-A0CO-01A-13R-A22K-07 TCGA-A2-A0CR-01A-11R-A22K-07
#> ENSG00000000457                     16.13222                    15.080922
#> ENSG00000001036                     17.61294                    17.961833
#> ENSG00000001167                     17.45729                    16.940356
#> ENSG00000001461                     16.79942                    16.447146
#> ENSG00000001630                     13.99033                     8.953481
#> ENSG00000002016                     15.47675                    14.826567
#>                 TCGA-A2-A0SU-01A-11R-A084-07 TCGA-A2-A0SX-01A-12R-A084-07
#> ENSG00000000457                     16.32653                     16.63090
#> ENSG00000001036                     18.56908                     18.13695
#> ENSG00000001167                     17.92969                     18.12988
#> ENSG00000001461                     17.64260                     16.99802
#> ENSG00000001630                     13.11417                     13.13048
#> ENSG00000002016                     14.61574                     15.46229
```

Additionally, genomic coordinates for genes can be included in a GRanges object as the rowRanges of the "expression" SummarizedExperiment object. To properly use TENET functions, this GRanges object should include gene IDs as names, which match with the IDs used as the rownames of the expression assay object. Additionally, it should at least include the chromosome, 1-indexed coordinates, and strand of each gene, and a metadata column named "geneName" which maps the IDs for genes to the gene names (as TENET generally assumes the user has used Ensembl IDs for genes, this allows TENET to map these back to the gene names for data summary and plots). Additional columns can be included, but are not used by TENET.

If this rowRanges dataset is not included in the "expression" SummarizedExperiment object, TENET can still be used, but the user will have to specify a `geneAnnotationDataset` in subsequent functions from which to pull information for the genes included in the "expression" SummarizedExperiment object. This dataset should contain gene and transcript information, and must be supplied as a GRanges object (such as one imported by `rtracklayer::import`) or a path to a GFF3 or GTF file. **Note:** TENET has only been tested with GENCODE and Ensembl gene annotation datasets. If you are using another dataset, please ensure that it uses the values "gene" and "transcript" for feature types, which must be stored in a column named "type". In GFF3 files, feature types may alternatively be stored in the first colon-separated field of the "ID" column, the second field of which must be the ID itself. Types stored there will only be used if the "type" column does not contain the required types. Gene names must be stored in a column named "geneName" or "Name". GTF files must contain a "geneId" column, and GFF3 files must contain an "ID" column. An annotation dataset specified as a GRanges object will be assumed to be derived from a GFF3 file if it contains an "ID" column, and from a GTF file otherwise. Ensembl GTF files older than release 75 are not supported.

It is not necessary to include a colData object in the "expression" SummarizedExperiment object, though a colData object in the outer MultiAssayExperiment object is required.

```r
## Examine the rowRanges of the "expression" SummarizedExperiment object for the
## first genes.
## Note: Additional columns are included here, but only the chromosome
## (seqnames), coordinates (rowRanges), strand, and geneName columns are used.
head(
    rowRanges(
        experiments(
            exampleTENETMultiAssayExperiment
        )[["expression"]]
    )
)
#> GRanges object with 6 ranges and 10 metadata columns:
#>                   seqnames              ranges strand |   source     type
#>                      <Rle>           <IRanges>  <Rle> | <factor> <factor>
#>   ENSG00000000457     chr1 169849631-169894267      - |   HAVANA     gene
#>   ENSG00000001036     chr6 143494812-143511720      - |   HAVANA     gene
#>   ENSG00000001167     chr6   41072945-41099976      + |   HAVANA     gene
#>   ENSG00000001461     chr1   24415802-24472976      + |   HAVANA     gene
#>   ENSG00000001630     chr7   92112153-92134803      - |   HAVANA     gene
#>   ENSG00000002016    chr12       911736-990053      - |   HAVANA     gene
#>                       score     phase            gene_id      gene_type
#>                   <numeric> <integer>        <character>    <character>
#>   ENSG00000000457        NA      <NA> ENSG00000000457.14 protein_coding
#>   ENSG00000001036        NA      <NA> ENSG00000001036.14 protein_coding
#>   ENSG00000001167        NA      <NA> ENSG00000001167.14 protein_coding
#>   ENSG00000001461        NA      <NA> ENSG00000001461.17 protein_coding
#>   ENSG00000001630        NA      <NA> ENSG00000001630.17 protein_coding
#>   ENSG00000002016        NA      <NA> ENSG00000002016.18 protein_coding
#>                     gene_name       level     hgnc_id          havana_gene
#>                   <character> <character> <character>          <character>
#>   ENSG00000000457       SCYL3           2  HGNC:19285 OTTHUMG00000035941.6
#>   ENSG00000001036       FUCA2           2   HGNC:4008 OTTHUMG00000015728.3
#>   ENSG00000001167        NFYA           2   HGNC:7804 OTTHUMG00000014669.1
#>   ENSG00000001461      NIPAL3           2  HGNC:25233 OTTHUMG00000003299.4
#>   ENSG00000001630     CYP51A1           1   HGNC:2649 OTTHUMG00000193420.2
#>   ENSG00000002016       RAD52           2   HGNC:9824 OTTHUMG00000090361.6
#>   -------
#>   seqinfo: 25 sequences from an unspecified genome; no seqlengths

## The names of this object are the gene IDs
```

```
head(
    names(
        rowRanges(
            experiments(
                exampleTENETMultiAssayExperiment
            )[["expression"]]
        )
    )
)
#> [1] "ENSG00000000457" "ENSG00000001036" "ENSG00000001167" "ENSG00000001461"
#> [5] "ENSG00000001630" "ENSG00000002016"
```

Methylation SummarizedExperiment object

A SummarizedExperiment object named “methylation” containing DNA methylation data for methylation sites. Methylation values should be given in the form of beta (β) values ranging from 0 (low methylation) to 1 (high methylation) with values for each RE DNA methylation site in the rows and data from each individual sample in the columns of the assay object in the “expression” SummarizedExperiment object. An ID for each RE DNA methylation site (often the ID of the corresponding probe in a methylation array) should be included in the rownames of this assay object. As stated above, the samples within the “methylation” SummarizedExperiment object should be matched with those in the “expression” SummarizedExperiment object.

```
## Again, examine the SummarizedExperiments included in the MultiAssayExperiment
## object, focusing on the "methylation" object here
experiments(exampleTENETMultiAssayExperiment)
#> ExperimentList class object of length 2:
#>  [1] expression: RangedSummarizedExperiment with 11637 rows and 242 columns
#>  [2] methylation: RangedSummarizedExperiment with 20000 rows and 242 columns

class(
    experiments(
        exampleTENETMultiAssayExperiment
    )[["methylation"]]
)
#> [1] "RangedSummarizedExperiment"
#> attr(,"package")
#> [1] "SummarizedExperiment"

## Examine data for the first six RE DNA methylation sites and samples in the
## assay object of the "methylation" SummarizedExperiment object
assays(
    experiments(
        exampleTENETMultiAssayExperiment
    )[["methylation"]]
)[[1]][
    seq_len(6), seq_len(6)
]
#>            TCGA-5L-AAT0-01A-12D-A41Q-05 TCGA-A1-A0SK-01A-12D-A10P-05
#> cg00002190                    0.8082020                   0.96617713
#> cg00002809                    0.9696186                   0.93638926
#> cg00002930                    0.1154501                   0.11167377
#> cg00008621                    0.9216933                   0.94481440
#> cg00010932                    0.2438511                   0.05579969
#> cg00011754                    0.8469034                   0.61155778
#>            TCGA-A2-A0CO-01A-13D-A22B-05 TCGA-A2-A0CR-01A-11D-A22B-05
#> cg00002190                   0.81227729                   0.86014089
#> cg00002809                   0.94525595                   0.92231070
#> cg00002930                   0.06936009                   0.06800885
#> cg00008621                   0.85299922                   0.91724358
#> cg00010932                   0.25145250                   0.43795638
#> cg00011754                   0.76882205                   0.84580930
#>            TCGA-A2-A0SU-01A-11D-A10P-05 TCGA-A2-A0SX-01A-12D-A10P-05
#> cg00002190                   0.90100287                   0.92015136
#> cg00002809                   0.96101718                   0.94437306
#> cg00002930                   0.08515684                   0.08049043
#> cg00008621                   0.94498701                   0.89035594
#> cg00010932                   0.18278488                   0.33361396
#> cg00011754                   0.89295317                   0.92339088
```

Like the “expression” object, the “methylation” SummarizedExperiment object can include a GRanges object in its rowRanges with information on the genomic coordinates for each RE DNA methylation site included in the assay of the “methylation” SummarizedExperiment. This GRanges object should include the methylation site IDs as names, and these should match with the IDs used as the rownames of the methylation assay object. Otherwise, this GRanges object only needs to include the chromosome and 1-indexed coordinates of each RE DNA methylation site. In contrast with the “expression” SummarizedExperiment object, strand information and names are not used. Additional columns can be included, but are not used by TENET.

If this rowRanges dataset is not included in the “methylation” SummarizedExperiment object, TENET can still be used, but the user will have to specify a `DNAMethylationArray` in subsequent functions from which to pull information for the RE DNA methylation sites included in the “methylation” SummarizedExperiment object. If this is the case, it is assumed the user has provided methylation beta values from probes in one of the Illumina methylation arrays supported by the sesameData package (see `?sesameData::sesameData_getManifestGRanges`).

It is not necessary to include a colData object in the “methylation” SummarizedExperiment object, though a colData object in the outer MultiAssayExperiment object is required.

```
## Examine the rowRanges of the "methylation" SummarizedExperiment object for
## the first six RE DNA methylation sites.
## Note: Additional columns are included here, but only the chromosome
## (seqnames) and coordinates (ranges) are used.
head(
    rowRanges(
        experiments(
            exampleTENETMultiAssayExperiment
        )[["methylation"]]
    )[, seq_len(6)]
)
#> GRanges object with 6 ranges and 6 metadata columns:
#>              seqnames              ranges strand | address_A address_B
#>                 <Rle>           <IRanges>  <Rle> | <integer> <integer>
#>   cg00002190     chr8   19697522-19697523      - |  62631497      <NA>
#>   cg00002809    chr17   78486271-78486272      + |  34718366  46624378
#>   cg00002930     chr6   31547621-31547622      - |  56793371      <NA>
#>   cg00008621    chr14   61713265-61713266      - |  24724419      <NA>
#>   cg00010932     chr2 169824154-169824155      - |  48671370      <NA>
#>   cg00011754     chr3   14543281-14543282      - |  63690419      <NA>
#>                  channel  designType    nextBase nextBaseRef
#>              <character> <character> <character> <character>
#>   cg00002190        Both          II         G/A           C
#>   cg00002809         Red           I           A           T
#>   cg00002930        Both          II         G/A           C
#>   cg00008621        Both          II         G/A           C
#>   cg00010932        Both          II         G/A           C
#>   cg00011754        Both          II         G/A           C
#>   -------
#>   seqinfo: 26 sequences from an unspecified genome; no seqlengths

## The names of this object are the RE DNA methylation site IDs (usually probe
## IDs)
head(
    names(
        rowRanges(
            experiments(
                exampleTENETMultiAssayExperiment
            )[["methylation"]]
        )
    )
)
#> [1] "cg00002190" "cg00002809" "cg00002930" "cg00008621" "cg00010932"
#> [6] "cg00011754"
```

MultiAssayExperiment colData object

The colData object should contain information for each of the patients from which samples in the "expression" and "methylation" SummarizedExperiment objects have been derived. The rownames of this object should be the patient IDs, which can be matched to samples using the MultiAssayExperiment object's sampleMap discussed subsequently. Columns of data can be included in this dataset to use some downstream step 7 TENET functions but are not essential for running most TENET functions. These include "vital_status" and "time" columns, with information on each sample's survival status and survival time, used in the `step7TopGenesSurvival` function, as well as a "purity" column and columns with gene copy number ("…_CNV") and somatic mutation ("…_SM") status used in the `step7ExpressionVsDNAMethylationScatterplots` function. See documentation for these functions for more information on how these data should be formatted. Additional columns of information can be included, but are not used by TENET.

```
## Examine the number of rows in the colData of the MultiAssayExperiment object
## compared to the number of samples (columns) in the "expression" and
## "methylation" summarized experiment objects. The number of patient entries
## does not need to match the number of samples included in the "expression" or
## "methylation" objects, as a single "Control" and "Case" sample could be
## derived from the same patient (though ideally, no more than one of each)
nrow(colData(exampleTENETMultiAssayExperiment))
#> [1] 231

experiments(exampleTENETMultiAssayExperiment)
#> ExperimentList class object of length 2:
#>  [1] expression: RangedSummarizedExperiment with 11637 rows and 242 columns
#>  [2] methylation: RangedSummarizedExperiment with 20000 rows and 242 columns
```

```
## Examine some of the rownames, which should contain a unique identifier for
## each patient. These will be used in the MultiAssayExperiment's sampleMap
## object to match them to the samples included in the "expression" and
## "methylation" objects
rownames(colData(exampleTENETMultiAssayExperiment))[seq_len(6)]
#> [1] "TCGA-5L-AAT0" "TCGA-A1-A0SK" "TCGA-A2-A0CO" "TCGA-A2-A0CR" "TCGA-A2-A0SU"
#> [6] "TCGA-A2-A0SX"
```

MultiAssayExperiment sampleMap object

The final essential component which should be included in the MultiAssayExperiment object for use in TENET analyses is the sampleMap object. This object is used to match the samples in the “expression” and “methylation” data objects to each other and match each of the samples in these datasets to the data contained in the MultiAssayExperiment colData, as well as to note which samples are “Control” samples, and which are “Case” samples. This object should have the following format:

```
## The sampleMap object should contain a row for each of the samples included
## in both the "expression" and "methylation" objects
nrow(sampleMap(exampleTENETMultiAssayExperiment))
#> [1] 484

experiments(exampleTENETMultiAssayExperiment)
#> ExperimentList class object of length 2:
#>  [1] expression: RangedSummarizedExperiment with 11637 rows and 242 columns
#>  [2] methylation: RangedSummarizedExperiment with 20000 rows and 242 columns

## 4 columns of data should be included in the sampleMap, "assay", "primary",
## "colname", and "sampleType"
colnames(sampleMap(exampleTENETMultiAssayExperiment))
#> [1] "assay"      "primary"    "colname"    "sampleType"

## The "assay" column should be a factor which lists which data object each
## sample is from ("expression", or "methylation)
sampleMap(exampleTENETMultiAssayExperiment)$assay[seq_len(6)]
#> [1] expression expression expression expression expression expression
#> Levels: expression methylation

levels(sampleMap(exampleTENETMultiAssayExperiment)$assay)
#> [1] "expression"  "methylation"

table(sampleMap(exampleTENETMultiAssayExperiment)$assay)
#> 
#>  expression methylation 
#>         242         242 

## The "primary" column should note which of the patient IDs from the
## MultiAssayExperiment's colData object each sample maps to.
sampleMap(exampleTENETMultiAssayExperiment)$primary[seq_len(6)]
#> [1] "TCGA-5L-AAT0" "TCGA-A1-A0SK" "TCGA-A2-A0CO" "TCGA-A2-A0CR" "TCGA-A2-A0SU"
#> [6] "TCGA-A2-A0SX"

table(
    sampleMap(exampleTENETMultiAssayExperiment)$primary %in%
        rownames(colData(exampleTENETMultiAssayExperiment))
)
#> 
#> TRUE 
#>  484 

## The "colname" column should include the sample names of each of the samples
## as they are listed in the colnames of either the "expression" or
## "methylation" SummarizedExperiments' assay objects
sampleMap(exampleTENETMultiAssayExperiment)$colname[seq_len(6)]
#> [1] "TCGA-5L-AAT0-01A-12R-A41B-07" "TCGA-A1-A0SK-01A-12R-A084-07"
#> [3] "TCGA-A2-A0C0-01A-13R-A22K-07" "TCGA-A2-A0CR-01A-11R-A22K-07"
#> [5] "TCGA-A2-A0SU-01A-11R-A084-07" "TCGA-A2-A0SX-01A-12R-A084-07"

table(
    sampleMap(exampleTENETMultiAssayExperiment)$colname %in% c(
        colnames(
            assays(
                experiments(
                    exampleTENETMultiAssayExperiment
                )[["expression"]]
            )[[1]]
        ),
        colnames(
            assays(
                experiments(
                    exampleTENETMultiAssayExperiment
                )[["methylation"]]
            )[[1]]
        )
    )
)
#> 
#> TRUE 
#>  484 

## Finally, the "sampleType" column should list whether each sample in the
## "expression" or "methylation" SummarizedExperiment objects is a "Case" or
## "Control" sample for the purposes of TENET analyses.
sampleMap(exampleTENETMultiAssayExperiment)$sampleType[seq_len(6)]
#> [1] "Case" "Case" "Case" "Case" "Case" "Case"

table(sampleMap(exampleTENETMultiAssayExperiment)$sampleType)
#> 
#>    Case Control 
#>     400      84
```

Overview of main TENET functions

- `step1MakeExternalDatasets`: Create a GRanges object representing putative regulatory element regions, based on the data sources selected for inclusion, to be used in later TENET steps

- step2GetDifferentiallyMethylatedSites: Identify differentially methylated RE DNA methylation sites
- step3GetAnalysisZScores: Calculate Z-scores comparing the mean expression of each gene in the case samples that are hyper- or hypomethylated for each RE DNA methylation site chosen in step 2
- step4SelectMostSignificantLinksPerDNAMethylationSite: Select the most significant RE DNA methylation site-gene links to each RE DNA methylation site
- step5OptimizeLinks: Find final RE DNA methylation site-gene links using various optimization metrics
- step6DNAMethylationSitesPerGeneTabulation: Tabulate the total number of RE DNA methylation sites linked to each of the genes
- TCGADownloader: Download TCGA gene expression, DNA methylation, and clinical datasets and compile them into a MultiAssayExperiment object
- TENETCacheAllData: Cache all online datasets required by TENET examples and optional features

step1MakeExternalDatasets: Create a GRanges object representing putative regulatory element regions, based on the data sources selected for inclusion, to be used in later TENET steps

This function creates a GRanges object containing regions representing putative regulatory elements, either enhancers or promoters, of interest to the user based on the presence of specific histone marks and open chromatin/nucleosome-depleted regions. This function can take input from user-specified bed-like files (see https://genome.ucsc.edu/FAQ/FAQformat.html#format1) containing regions with histone modification (via the extHM argument) and/or open chromatin/nucleosome-depleted regions (via the extNDR argument), as well as preprocessed enhancer, promoter, and open chromatin datasets from many cell/tissue types included in the TENET.ExperimentHub package. The resulting GRanges object will be returned. GRanges objects created by this function can be used by the step2GetDifferentiallyMethylatedSites function or other downstream functions.

Regulatory element regions of interest identified by this function represent those with overlapping histone mark peaks as well as open chromatin regions, combined with any regions identified in the selected ENCODE SCREEN datasets (as these regions already represent the overlap of regions with relevant histone marks as well as with open chromatin).

```
## Create an example GRanges object representing putative enhancer regions for
## BRCA using all available enhancer-relevant BRCA datasets present in the
## TENET.ExperimentHub package. These datasets include consensus enhancer
## histone mark and open chromatin datasets from a wide variety of tissue and
## cell types, enhancer histone mark and open chromatin datasets from a
## variety of BRCA-relevant samples from the Cistrome database and TCGA, as well
## as identified distal enhancer regions from the ENCODE SCREEN project.
step1Output <- step1MakeExternalDatasets(
    consensusEnhancer = TRUE,
    consensusNDR = TRUE,
    publicEnhancer = TRUE,
    publicNDR = TRUE,
    cancerType = "BRCA",
    ENCODEdELS = TRUE
)
#> Loading Consensus enhancer regions (AH116724) from AnnotationHub
#> Loading Consensus open chromatin regions (AH116725) from AnnotationHub
#> Loading Public enhancer regions (AH116721) from AnnotationHub
#> Loading Public open chromatin regions (AH116722) from AnnotationHub
#> Loading ENCODE dELS regions (AH116727) from AnnotationHub

## View the first regions in the created GRanges object
```

```
head(step1Output)
#> GRanges object with 6 ranges and 0 metadata columns:
#>       seqnames        ranges strand
#>          <Rle>     <IRanges>  <Rle>
#>   [1]     chr1   10121-10270      *
#>   [2]     chr1   10389-10400      *
#>   [3]     chr1   16141-16290      *
#>   [4]     chr1   20061-20210      *
#>   [5]     chr1 135126-135275      *
#>   [6]     chr1 136281-136430      *
#>   -------
#>   seqinfo: 25 sequences from an unspecified genome; no seqlengths
```

`step2GetDifferentiallyMethylatedSites`: Identify differentially methylated RE DNA methylation sites

This function identifies DNA methylation sites that mark putative regulatory elements (REs), including enhancer and promoter regions. These are sites that lie within regions with specific histone modifications and open chromatin regions, from a user-supplied GRanges object, such as one created by the `step1MakeExternalDatasets` function, and which are located at a user-specified distance relative to the transcription start sites (TSS) listed in either the rowRanges of the elementMetadata of the "expression" SummarizedExperiment in the TENETMultiAssayExperiment object, or the selected `geneAnnotationDataset` (which will be filtered to only genes and transcripts). After identifying DNA methylation sites representing the specified REs, the function classifies the RE DNA methylation sites as methylated, unmethylated, hypermethylated, or hypomethylated based on their differential methylation between the control and case samples supplied by the user, defined by cutoff values which are either automatically based on the mean methylation densities of the identified RE DNA methylation sites, or manually set by the user. **Note:** Using the algorithm to set cutoffs is recommended for use with DNA methylation array data, and may not work for whole-genome DNA methylation data.

To run step 2, the user will need to provide a MultiAssayExperiment object constructed in the manner described previously, as well as a GRanges object, such as one created by the `step1MakeExternalDatasets` function. Additionally, the minimum number of case samples that must exhibit a difference in methylation for a given RE DNA methylation site to be considered hyper- or hypomethylated will need to be set by the user.

The output of the `step2GetDifferentiallyMethylatedSites` function, as well as later TENET functions, is saved in the metadata of the returned MultiAssayExperiment object.

```
## Identify differentially methylated RE DNA methylation sites using the
## step2GetDifferentiallyMethylatedSites function, using the
## exampleTENETMultiAssayExperiment object loaded previously from the
## TENET.ExperimentHub package as a reference, and the GRanges object that was
## just created using the step1MakeExternalDatasets function. At least 5 case
## samples in the dataset are required to show methylation levels above/below
## the hyper/hypomethylation cutoff for a given RE DNA methylation site to be
## potentially considered differentially methylated.
## All transcription start sites (TSS) included in the rowRanges of the
## elementMetadata of the TENETMultiAssayExperiment object will be considered
## when selecting enhancer DNA methylation sites (which must be at least 1500
## bp from these TSS).
exampleObject <- step2GetDifferentiallyMethylatedSites(
    TENETMultiAssayExperiment = exampleTENETMultiAssayExperiment,
    regulatoryElementGRanges = step1Output,
    minCaseCount = 5
)

## See the data that were saved by the step 2 function
names(metadata(exampleObject)$step2GetDifferentiallyMethylatedSites)
#>  [1] "unmethCutoff"                "methCutoff"
#>  [3] "hypermethCutoff"             "hypomethCutoff"
#>  [5] "minCaseCount"                "counts"
#>  [7] "siteIdentitiesList"          "regulatoryElementGRanges"
#>  [9] "methylationDistributionPlot" "methylationCutoffsPlot"

## Since cutoffs were set automatically by the step 2 function in this case,
## we can see what they are set to, using the hypomethylation cutoff as an
## example.
metadata(
    exampleObject
)$step2GetDifferentiallyMethylatedSites$hypomethCutoff
#> [1] 0.627

## The identities of all identified RE DNA methylation sites, as well as the
## methylated, unmethylated, and most importantly, hyper- and hypomethylated
## RE DNA methylation sites are also recorded in the siteIdentitiesList. To
## demonstrate this, view the first hypomethylated RE DNA methylation sites
## that were identified.
head(
    metadata(
        exampleObject
    )$step2GetDifferentiallyMethylatedSites$siteIdentitiesList$
        hypomethylatedSites
)
#> [1] "cg00002190" "cg00002809" "cg00018850" "cg00047815" "cg00051307"
#> [6] "cg00054971"
```

`step3GetAnalysisZScores`: Calculate Z-scores comparing the mean expression of each gene in the case samples that are hyper- or hypomethylated for each RE DNA methylation site chosen in step 2

This function calculates Z-scores comparing the mean expression of each gene in the case samples that are hyper- or hypomethylated for each RE DNA methylation site chosen in step 2 to the mean expression of the remaining non hyper- or hypomethylated case samples. By identifying significant Z-scores, initial RE DNA methylation site-gene links are identified, as case samples with hyper- or hypomethylation of a particular RE DNA methylation site also display particularly high or low expression of specific genes.

TENET supports the use of two different formulas for calculating Z-scores in this step. By setting the zScoreCalculation argument to “oneSample”, a one-sample Z-score calculation will be used (similar to previous versions of the TENET package), while a two-sample Z-score calculation will be used if the zScoreCalculation argument is set to “twoSample”.

Also, the sparseResults argument has been included in order to reduce the size of the MultiAssayExperiment object with TENET results. By setting this to TRUE, only links with significant Z-scores (as determined by the value of the pValue argument) are saved in the MultiAssayExperiment object returned by this function. However, setting this to TRUE affects the function of the subsequent `step4SelectMostSignificantLinksPerDNAMethylationSite` function if the user wishes to perform multiple testing correction to select the most significant links per RE DNA methylation site. Therefore, if you want to use multiple testing correction instead of just selecting the top n most significant links per RE DNA methylation site in the `step4SelectMostSignificantLinksPerDNAMethylationSite`, the sparseResults argument should be set to FALSE so the multiple testing correction is properly applied for all results, not just the significant ones.

Note: This function takes the longest of all TENET functions to run. It is highly recommended to use multiple cores if possible, especially when using large datasets.

```r
## Identify significant Z-scores and initial RE DNA methylation site-gene links
## using the exampleTENETMultiAssayExperiment with results from the
## step2GetDifferentiallyMethylatedSites function. For this analysis, we will
## use the one-sample Z-score function, consider only TFs, rather than all
## genes, and save only significant Z-scores, to cut down on computational time
## and reduce the size of the returned MultiAssayExperiment object.
exampleObject <- step3GetAnalysisZScores(
    TENETMultiAssayExperiment = exampleObject,
    pValue = 0.05,
    TFOnly = TRUE,
    zScoreCalculation = "oneSample",
    hypermethAnalysis = TRUE,
    hypomethAnalysis = TRUE,
    includeControl = FALSE,
    sparseResults = TRUE
)

## See the data that were saved by the step 3 function. They are subdivided into
## hypermeth and/or hypometh results based on function options.
names(
    metadata(
        exampleObject
    )$step3GetAnalysisZScores
)

## Since the sparseResults argument was set to TRUE, only
## significant Z-scores are saved, and since the TFOnly argument was also set
## to TRUE, only TF genes were analyzed.
## View the significant Z scores for the first TF genes with links to
## hypomethylated RE DNA methylation sites.
head(
    metadata(
        exampleObject
    )$step3GetAnalysisZScores$hypomethResults
)
```

`step4SelectMostSignificantLinksPerDNAMethylationSite`: Select the most significant RE DNA methylation site-gene links to each RE DNA methylation site

This function takes the calculated Z-scores for the hyper- or hypomethylated G+ RE DNA methylation site-gene links and selects the most significant links to each RE DNA methylation site, either up to a number specified by the user, or based on a significant p-value level set by the user after multiple testing correction is performed on the Z-scores output by the `step3GetAnalysisZScores` function per RE DNA methylation site in the RE DNA methylation site-gene pairs. This helps prioritize individual RE DNA methylation site-gene links where there are many genes linked to a single RE DNA methylation site.

As described previously, if you wish to use multiple testing correction, the sparseResults argument in the previous `step3GetAnalysisZScores` function should have been set to FALSE, otherwise it will affect the generated results (TENET will display a message warning about this if this is the case) as with sparseResults, only significant results are saved from step 3 and then used in the multiple testing which affects the values and number of tests accounted for.

This warning will also occur if multiple testing is done using the example MultiAssayExperiment object, as the results from step 3 in the object were created with sparseResults set to TRUE. This is just a warning however, and results will still be generated by the function and can be used in downstream functions.

```
## Get the 25 (if present) most significant links per RE DNA methylation site
## identified by the step3GetAnalysisZScores function
exampleObject <- step4SelectMostSignificantLinksPerDNAMethylationSite(
    TENETMultiAssayExperiment = exampleObject,
    hypermethGplusAnalysis = TRUE,
    hypomethGplusAnalysis = TRUE,
    linksPerREDNAMethylationSiteMaximum = 25
)

## See the data that were saved by the step 4 function. They are subdivided into
## hypermeth and/or hypometh results based on function options.
names(
    metadata(exampleObject)$step4SelectMostSignificantLinksPerDNAMethylationSite
)
#> [1] "hypermethGplusResults" "hypomethGplusResults"

## View the results for the most significant links to the hypomethylated RE
## DNA methylation sites
head(
    metadata(
        exampleObject
    )$step4SelectMostSignificantLinksPerDNAMethylationSite$hypomethGplusResults
)
#> $cg00002190
#> ENSG00000091831 ENSG00000129514 ENSG00000276644 ENSG00000118513 ENSG00000171604
#>         -3.1382         -2.7997         -2.7592         -2.3383         -1.9945
#> ENSG00000267179 ENSG00000107485 ENSG00000124664
#>         -1.8924         -1.8760         -1.7922
#>
#> $cg00002809
#> ENSG00000091831 ENSG00000129514 ENSG00000166949 ENSG00000177853 ENSG00000197343
#>         -4.8726         -3.3984         -3.1009         -3.0432         -2.6059
#> ENSG00000140987 ENSG00000171817 ENSG00000118513 ENSG00000124664 ENSG00000064489
#>         -2.1645         -2.0573         -2.0227         -2.0125         -1.9648
#> ENSG00000130182 ENSG00000106571 ENSG00000196652 ENSG00000197050 ENSG00000115966
#>         -1.8973         -1.8388         -1.8120         -1.7712         -1.6805
#>
#> $cg00047815
#> ENSG00000118513 ENSG00000129514 ENSG00000064489 ENSG00000124664 ENSG00000130751
#>         -3.1776         -2.9648         -2.8116         -2.3235         -2.3207
#> ENSG00000107485 ENSG00000144485 ENSG00000249961 ENSG00000106571 ENSG00000169951
#>         -2.1665         -2.1457         -2.0882         -1.9356         -1.9301
#> ENSG00000165643 ENSG00000127989 ENSG00000091831 ENSG00000198807 ENSG00000126733
#>         -1.9146         -1.8036         -1.7000         -1.6718         -1.6556
#>
```

```
#> $cg00051307
#> ENSG00000124664 ENSG00000129514
#>         -1.7865         -1.7031
#>
#> $cg00054971
#> ENSG00000064489 ENSG00000124232 ENSG00000107859 ENSG00000165643 ENSG00000215612
#>         -2.0858         -2.0174         -1.9708         -1.9612         -1.9100
#> ENSG00000198911
#>         -1.8578
#>
#> $cg00069003
#> ENSG00000129514 ENSG00000124664 ENSG00000169083 ENSG00000197343 ENSG00000100219
#>         -2.2471         -1.9416         -1.8633         -1.7237         -1.7146
```

step5OptimizeLinks: Find final RE DNA methylation site-gene links using various optimization metrics

This function takes the most significant hyper- or hypomethylated G+ RE DNA methylation site-gene links selected in step 4, and selects optimized links based on the relative expression of the given gene in hyper- or hypomethylated case samples compared to control samples, using an unpaired two-sided Wilcoxon rank-sum test to check that the hyper- or hypomethylated samples for that given RE DNA methylation site-gene link also show appropriately higher/lower expression of the linked gene in a number of case samples greater than or equal to the `minCaseCount` number specified in the `step2GetDifferentiallyMethylatedSites` function that have maximum/minimum methylation above/below the `hyperStringency`/`hypoStringency` cutoff values selected.

This identifies the final RE DNA methylation site-gene links by prioritizing those that meet the above criteria. The output of this function is used in many of the downstream TENET functions, and helps users examine the individual RE DNA methylation sites linked to each gene.

```
## Identify final optimized RE DNA methylation site-gene links
exampleObject <- step5OptimizeLinks(
    TENETMultiAssayExperiment = exampleObject,
    hypermethGplusAnalysis = TRUE,
    hypomethGplusAnalysis = TRUE,
    expressionPvalCutoff = 0.05
)

## See the data that were saved by the step 5 function. They are again
## subdivided into hypermeth and/or hypometh results based on function options.
names(metadata(exampleObject)$step5OptimizeLinks)
#> [1] "hypermethGplusResults" "hypomethGplusResults"

## Check the results, which include various metrics used to priortize the
## optimized final RE DNA methylation site-gene links.
## This is a subsection of the data frame detailing all the hypomethylated RE
## DNA methylation site-gene links as an example.
head(
    metadata(
        exampleObject
    )$step5OptimizeLinks$hypomethGplusResults
)
#>               geneID DNAMethylationSiteID  zScore       pValue
#> 1605 ENSG00000001167           cg06051912 -3.3244 4.430450e-04
#> 1103 ENSG00000006194           cg04134755 -1.9817 2.375641e-02
#> 1147 ENSG00000006704           cg04301738 -2.6380 4.169829e-03
#> 1300 ENSG00000006704           cg04824378 -3.7728 8.071284e-05
#> 3705 ENSG00000006704           cg14986222 -1.9452 2.587546e-02
#> 780  ENSG00000007372           cg03025986 -3.0505 1.142303e-03
#>      meanExpressionControl meanExpressionCase
#> 1605             17.543262          18.048079
#> 1103             16.779795          17.082728
#> 1147             15.760415          16.832159
#> 1300             15.760415          16.832159
#> 3705             15.760415          16.832159
#> 780               4.531898           7.039058
#>      wilcoxonPValExpressionControlVsCase hypomethCaseLength meanHypomethCase
#> 1605                        4.053345e-04                  5         18.63578
#> 1103                        4.616363e-04                  5         17.68783
#> 1147                        3.114415e-04                  5         17.36836
#> 1300                        3.114415e-04                  5         17.25570
#> 3705                        9.140798e-05                  6         17.23622
#> 780                         1.199587e-04                  6         13.12104
#>      hypomethHigherCaseLength minMethCaseValue
#> 1605                        5       0.51477122
#> 1103                        5       0.48421779
#> 1147                        5       0.53490426
#> 1300                        5       0.33877410
#> 3705                        6       0.45494150
#> 780                         6       0.04142473
#>      meanExpressionControlLowExpressionCase
#> 1605                                   TRUE
#> 1103                                   TRUE
#> 1147                                   TRUE
#> 1300                                   TRUE
#> 3705                                   TRUE
#> 780                                    TRUE
#>      wilcoxonPValExpressionControlVsCaseAdjusted
#> 1605                                0.0008872838
#> 1103                                0.0009882211
#> 1147                                0.0007395597
#> 1300                                0.0007395597
#> 3705                                0.0002790778
#> 780                                 0.0003453130
```

step6DNAMethylationSitesPerGeneTabulation: Tabulate the total number of RE DNA methylation sites linked to each of the genes

This function takes the final optimized RE DNA methylation site-gene links identified in step 5 and tabulates the number of these links per gene. This tabulation is done separately for both of the hyper- or hypomethylated G+ analysis quadrants, as selected by the user.

This aids in prioritizing the top genes for downstream analyses, as the genes with the most linked RE DNA methylation sites are the most likely to represent those with widespread genomic impact.

```
## Prioritize the top genes by adding up the number of RE DNA methylation sites
## linked to each of the genes
exampleObject <- step6DNAMethylationSitesPerGeneTabulation(
    TENETMultiAssayExperiment = exampleObject
)

## See the data that were saved by the step 6 function. They are again
## subdivided into hypermeth and/or hypometh results based on function options.
names(
    metadata(
        exampleObject
    )$step6DNAMethylationSitesPerGeneTabulation
)
#> [1] "hypermethGplusResults" "hypomethGplusResults"

## Check the results, which include a count of the RE DNA methylation sites per
## gene, and is organized by decreasing RE DNA methylation site count.
## This is a subsection of the data frame detailing the number of hypomethylated
## RE DNA methylation site links to the top TFs.
head(
    metadata(
        exampleObject
    )$step6DNAMethylationSitesPerGeneTabulation$hypomethGplusResults
)
#>                          geneID count geneName
#> ENSG00000129514 ENSG00000129514   331    FOXA1
#> ENSG00000124664 ENSG00000124664   243    SPDEF
#> ENSG00000107485 ENSG00000107485   170    GATA3
#> ENSG00000091831 ENSG00000091831   165     ESR1
#> ENSG00000118513 ENSG00000118513   139      MYB
#> ENSG00000100219 ENSG00000100219    73     XBP1
```

`TCGADownloader`: Download TCGA gene expression, DNA methylation, and clinical datasets and compile them into a MultiAssayExperiment object

This function downloads and compiles TCGA gene expression and DNA methylation datasets, as well as clinical data primarily intended for use with the TENET package. This simplifies the TCGAbiolinks download functions, identifies samples with matching gene expression and DNA methylation data, and can also remove duplicate tumor samples taken from the same patient donor. Data are compiled into a MultiAssayExperiment object, which is returned and optionally saved in an .rda file at the path specified by the `outputFile` argument.

```
## Download a TCGA BRCA dataset with log2-normalized
## FPKM-UQ expression values from tumor and adjacent normal tissue samples
## with matching expression and methylation data and keeping only one tumor
## sample from each patient. Additionally, survival data will be combined
## from three clinical datasets downloaded by TCGAbiolinks. Raw data files
## will be saved to the working directory, and the processed dataset will
## be returned as a variable.
TCGADataset <- TCGADownloader(
    rawDataDownloadDirectory = ".",
    TCGAStudyAbbreviation = "BRCA",
    RNASeqWorkflow = "STAR - FPKM-UQ",
    RNASeqLog2Normalization = TRUE,
    matchingExpAndMetSamples = TRUE,
    clinicalSurvivalData = "combined",
```

```r
    outputFile = NA
)
```

`TENETCacheAllData`: Cache all online datasets required by TENET examples and optional features

This function locally caches all online TENET and SeSAMe datasets required by TENET examples and optional features (TENET.ExperimentHub objects used in examples, TENET.AnnotationHub datasets used in step 1, and SeSAMe datasets loaded via the `DNAMethylationArray` argument). The main purpose of this function is to enable the use of TENET in an HPC cluster environment where compute nodes do not have internet access. In this case, you must run `TENETCacheAllData()` once while connected to the internet before using TENET examples or these optional features.

```r
## Cache all online datasets required by optional TENET features.
## This function takes no arguments and returns NULL.
TENETCacheAllData()
```

Overview of downstream step 7 functions

The step 7 functions aim to perform downstream analyses based on the identified RE DNA methylation site-gene links, integrating multi-omic datasets such as Hi-C, copy number variation, somatic mutation, and patient survival information.

- `step7ExpressionVsDNAMethylationScatterplots`: Create scatterplots displaying the expression of the top genes and the methylation levels of each of their linked RE DNA methylation sites, along with copy number variation, somatic mutation, and purity data, if provided by the user
- `step7LinkedDNAMethylationSiteCountHistograms`: Create histograms displaying the number of genes or transcription factors linked to a given number of RE DNA methylation sites
- `step7LinkedDNAMethylationSitesMotifSearching`: Perform motif searching for transcription factor motifs in the vicinity of RE DNA methylation sites linked to the specified transcription factors
- `step7SelectedDNAMethylationSitesCaseVsControlBoxplots`: Generate boxplots comparing the methylation level of the specified RE DNA methylation sites in case and control samples
- `step7StatesForLinks`: Identify which of the case samples harbor each of the identified regulatory element DNA methylation site-gene links
- `step7TopGenesCaseVsControlExpressionBoxplots`: Create boxplots comparing the expression level of the top genes/transcription factors in case and control samples
- `step7TopGenesCircosPlots`: Generate Circos plots displaying the links between top identified genes and each of the RE DNA methylation sites linked to them
- `step7TopGenesDNAMethylationHeatmaps`: Generate heatmaps displaying the methylation level of all RE DNA methylation sites linked to the top genes/transcription factors, along with the expression of those genes in the column headers, in the case samples within the supplied MultiAssayExperiment object
- `step7TopGenesExpressionCorrelationHeatmaps`: Generate mirrored heatmaps displaying the correlation of the expression values of the top genes/TFs
- `step7TopGenesOverlappingLinkedDNAMethylationSitesHeatmaps`: Generate binary heatmaps displaying which of the top genes/transcription factors share links with each of the unique regulatory element DNA methylation sites linked to at least one top gene/TF
- `step7TopGenesSurvival`: Perform Kaplan-Meier and Cox regression analyses to assess the association of top gene expression and linked RE DNA methylation site methylation with patient survival
- `step7TopGenesTADTables`: Create tables using user-supplied topologically associating domain (TAD) information which identify the topologically associating domains containing each RE DNA methylation site linked to the top genes/transcription factors, as well as other genes in the same topologically associating domain as potential downstream targets

- `step7TopGenesUCSCBedFiles`: Create bed-formatted interact files which can be loaded on the UCSC Genome Browser to display links between top genes and transcription factors and their linked RE DNA methylation sites
- `step7TopGenesUserPeakOverlap`: Identify if RE DNA methylation sites linked to top genes/transcription factors are located within a specific distance of specified genomic regions

Here we will demonstrate the usage of some step 7 functions.

`step7ExpressionVsDNAMethylationScatterplots`: Create scatterplots displaying the expression of the top genes and the methylation levels of each of their linked RE DNA methylation sites, along with copy number variation, somatic mutation, and purity data, if provided by the user

These scatterplots show the relationship between genes and RE DNA methylation sites, displaying the expression of the genes in the X-axis and the methylation of the sites in the Y-axis. The sample type (case or control) is also displayed in these plots. The shape and size of the points on the scatterplots represent copy number variation (CNV), somatic mutation (SM) status, and purity for the samples in the scatterplots.

First, we load the example CNV, SM, and purity data from the `exampleTENETClinicalDataFrame` object.

```r
## Load the exampleTENETClinicalDataFrame object from the TENET.ExperimentHub
## package. It contains copy number variation (CNV), somatic mutation (SM),
## and purity data for the top 10 TFs by linked hypomethylated RE
## DNA methylation sites in the exampleTENETMultiAssayExperiment object.
exampleTENETClinicalDataFrame <-
    TENET.ExperimentHub::exampleTENETClinicalDataFrame()
#> see ?TENET.ExperimentHub and browseVignettes('TENET.ExperimentHub') for documentation
#> loading from cache
CNVData <- subset(exampleTENETClinicalDataFrame,
    select = grepl("_CNV$", colnames(exampleTENETClinicalDataFrame))
)
SMData <- subset(exampleTENETClinicalDataFrame,
    select = grepl("_SM$", colnames(exampleTENETClinicalDataFrame))
)
purityData <- subset(exampleTENETClinicalDataFrame, select = "purity")
```

The CNV dataset is a numeric data frame with rownames representing sample names and colnames representing gene IDs folllowed by "_CNV", with -2 representing a loss of both copies, -1 a single copy loss, 0 no copy number change, and positive integer values representing a gain of that many copies (though changes of +2 or greater are grouped together in the scatterplots).

```r
## Show the CNV data for the first 4 TFs
head(CNVData[, 1:4])
#>              ENSG00000165821_CNV ENSG00000169989_CNV ENSG00000197343_CNV
#> TCGA-5L-AAT0                  -1                  -1                  -2
#> TCGA-A1-A0SK                  -2                   0                   2
#> TCGA-A2-A0C0                   2                  -1                   1
#> TCGA-A2-A0CR                   0                  -1                  -2
#> TCGA-A2-A0SU                  -2                   2                   0
#> TCGA-A2-A0SX                  -2                   2                   0
#>              ENSG00000169083_CNV
#> TCGA-5L-AAT0                  -1
#> TCGA-A1-A0SK                   1
#> TCGA-A2-A0C0                   1
#> TCGA-A2-A0CR                  -2
#> TCGA-A2-A0SU                   1
#> TCGA-A2-A0SX                   2
```

The SM dataset is a numeric data frame with rownames representing sample names and colnames representing gene IDs folllowed by "_SM", with 0 representing no mutation and 1 representing mutation.

```
## Show the SM data for the first 4 TFs
head(SMData[, 1:4])
#>              ENSG00000165821_SM ENSG00000169989_SM ENSG00000197343_SM
#> TCGA-5L-AAT0                  0                  1                  1
#> TCGA-A1-A0SK                  0                  0                  1
#> TCGA-A2-A0CO                  1                  1                  0
#> TCGA-A2-A0CR                  0                  0                  1
#> TCGA-A2-A0SU                  0                  1                  0
#> TCGA-A2-A0SX                  0                  1                  1
#>              ENSG00000169083_SM
#> TCGA-5L-AAT0                  0
#> TCGA-A1-A0SK                  1
#> TCGA-A2-A0CO                  0
#> TCGA-A2-A0CR                  0
#> TCGA-A2-A0SU                  1
#> TCGA-A2-A0SX                  1
```

The purity dataset in this example is a numeric data frame with rownames representing sample names and the first column representing the purity, with the values ranging from 0 (low purity) to 1 (high purity). It can also be a numeric vector with names representing the sample names.

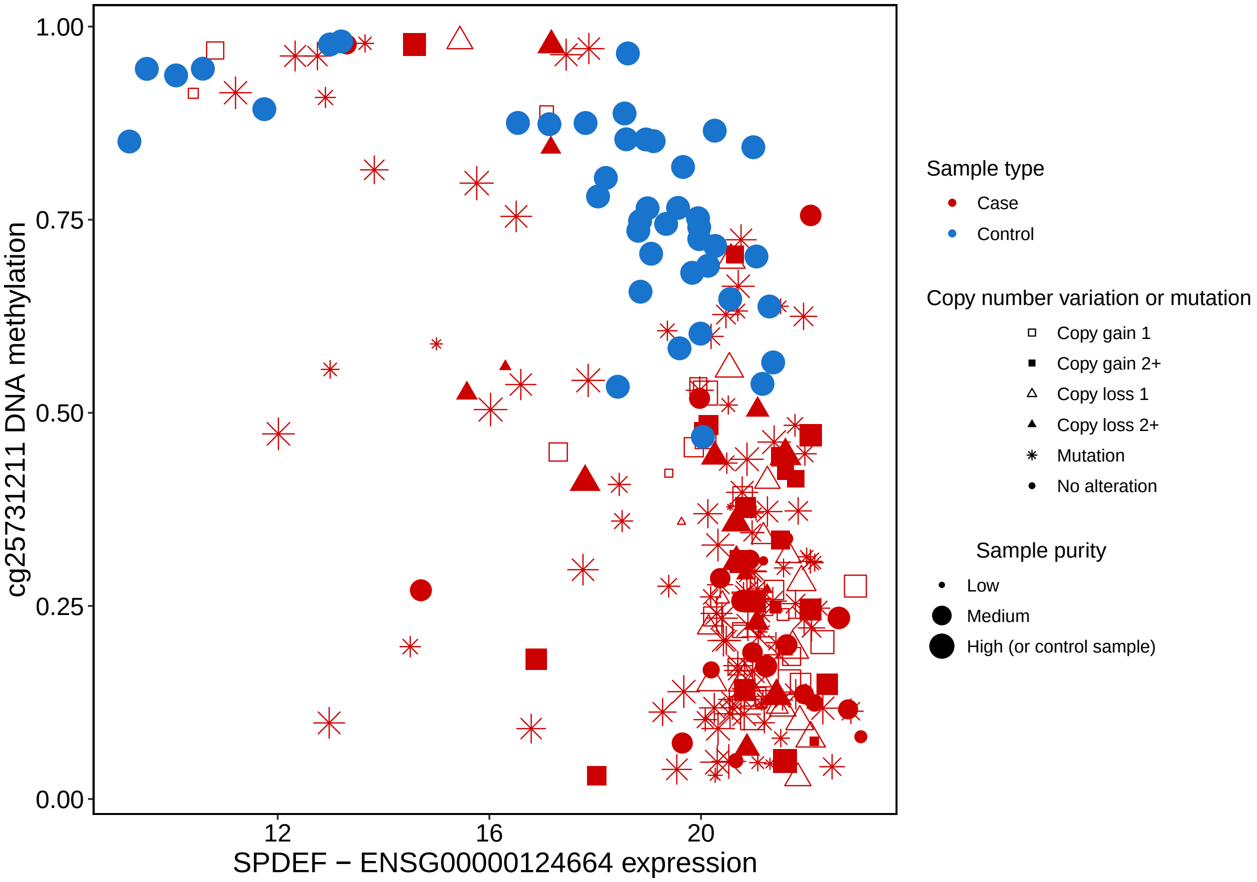
```
## Show the first few rows of the purity data
head(purityData)
#>                 purity
#> TCGA-5L-AAT0 0.7750025
#> TCGA-A1-A0SK 0.5658861
#> TCGA-A2-A0CO 0.2231461
#> TCGA-A2-A0CR 0.7627627
#> TCGA-A2-A0SU 0.8245055
#> TCGA-A2-A0SX 0.3525651
```

The options `CNVData`, `SMData`, and `purityData` are not required. If they are supplied and `simpleOrComplex` is set to "complex", complex scatterplots will be created displaying this information. Otherwise, simple scatterplots will be created. At this time, either all or none of `CNVData`, `SMData`, and `purityData` must be specified.

```
## Create complex scatterplots using the previously acquired data.
## Since we performed analyses only using TFs in the step 3 function, the
## top genes are all TFs, so a message that separate output for TFs will be
## skipped is displayed.
exampleObject <- step7ExpressionVsDNAMethylationScatterplots(
    TENETMultiAssayExperiment = exampleObject,
    hypermethGplusAnalysis = FALSE,
    hypomethGplusAnalysis = TRUE,
    purityData = purityData,
    SMData = SMData,
    CNVData = CNVData,
    simpleOrComplex = "complex",
    topGeneNumber = 10
)
#> All genes with hypomethylated G+ links are TFs, so the separate output for TFs will be
↪  skipped.

## Results are included in the metadata of the returned MultiAssayExperiment
## object under the step7ExpressionVsDNAMethylationScatterplots list.
## For each analysis type, results are included in sub-lists, each of which
## contains results for topGenes and topTFs, unless the top genes are
## all TFs, in which case the separate top TFs output is skipped.
names(
    metadata(
        exampleObject
    )$step7ExpressionVsDNAMethylationScatterplots$hypomethGplusResults$topGenes
)
#>  [1] "ENSG00000129514" "ENSG00000124664" "ENSG00000107485" "ENSG00000091831"
#>  [5] "ENSG00000118513" "ENSG00000100219" "ENSG00000152192" "ENSG00000105261"
#>  [9] "ENSG00000178935" "ENSG00000115163"

## For each gene, scatterplots are generated showing the expression of that
## gene and the methylation of each RE DNA methylation site linked to it for
## the given analysis type.
head(
    names(
        metadata(
            exampleObject
        )$step7ExpressionVsDNAMethylationScatterplots$hypomethGplusResults$
            topGenes$ENSG00000124664
    )
)
#> [1] "cg00002190" "cg00002809" "cg00047815" "cg00051307" "cg00069003"
#> [6] "cg00085256"
```

As an example, we examine the scatterplot with the expression of the TF SPDEF (ENSG00000124664) and the methylation of its linked hypomethylated RE DNA methylation site with the ID cg25731211. Gene expression is given in the X-axis and methylation is given in the Y-axis. Samples are colored based on whether they are are cases (red) or controls (blue). The shape and size of the points are determined by each sample's CNV/SM status and purity, respectively, since complex scatterplots were selected.

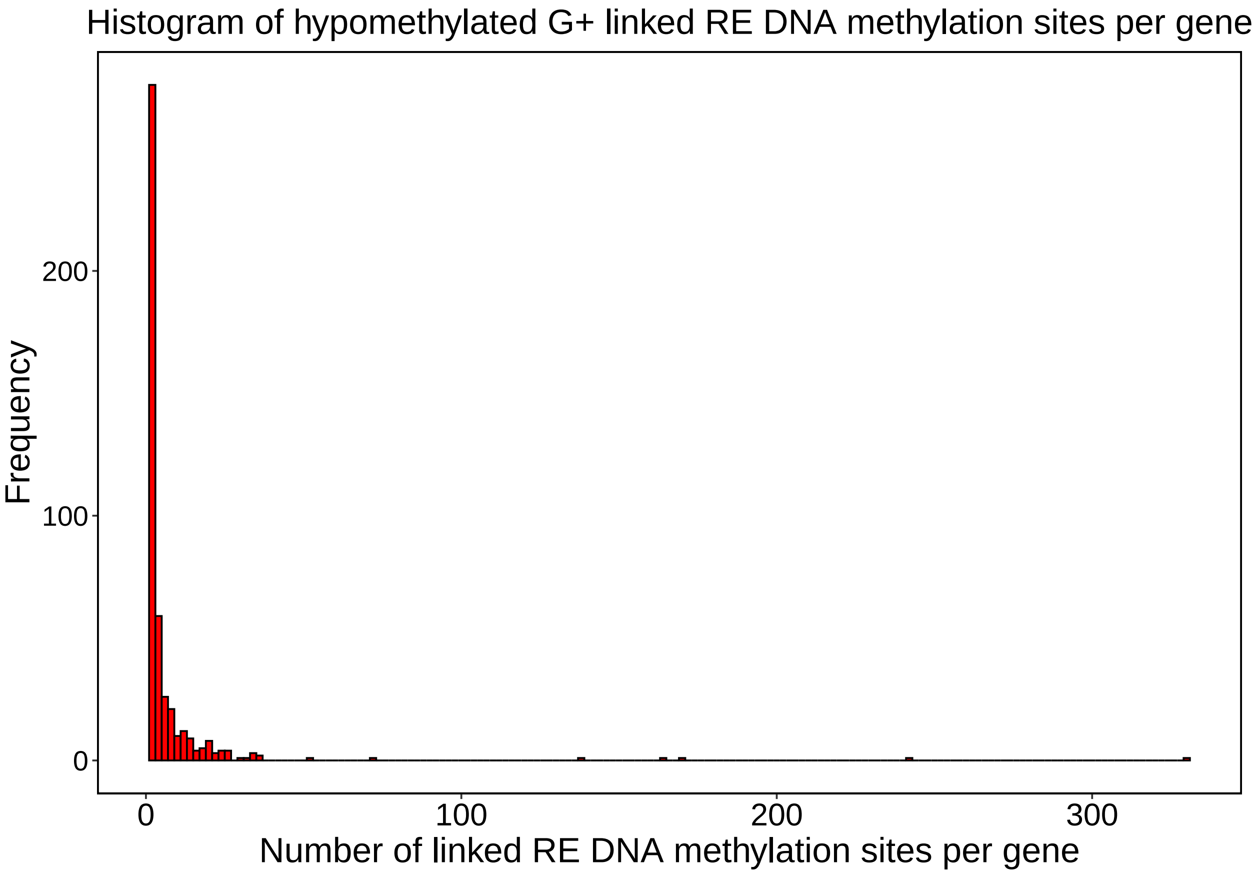
```
metadata(
    exampleObject
)$step7ExpressionVsDNAMethylationScatterplots$hypomethGplusResults$
    topGenes$ENSG00000124664$cg25731211
```



`step7LinkedDNAMethylationSiteCountHistograms`: Create histograms displaying the number of genes or transcription factors linked to a given number of RE DNA methylation sites

This function generates histograms displaying the frequency of genes by the number of RE DNA methylation sites linked to them. These are designed to highlight the top genes/TFs, which likely have a disproportionately large number of linked RE DNA methylation sites compared to most genes/TFs.

```
## Run the step7LinkedDNAMethylationSiteCountHistograms function.
## Since we performed analyses only using TFs in the step 3 function, the
## top genes are all TFs, so a message that separate output for
## TFs will be skipped is displayed.
exampleObject <- step7LinkedDNAMethylationSiteCountHistograms(
    TENETMultiAssayExperiment = exampleObject,
    hypomethGplusAnalysis = TRUE,
    hypermethGplusAnalysis = FALSE
)
#> All genes with hypomethylated G+ links are TFs, so the separate output for TFs will be
↪  skipped.

## Results are included in the metadata of the returned MultiAssayExperiment
## object under the step7LinkedDNAMethylationSiteCountHistograms list.
## For each analysis type, histograms are included in sub-lists, each of which
## contains results for topGenes and topTFs, unless the top genes are all TFs,
## in which case the separate top TFs output is skipped.
## Display the histogram. Note the relatively small number of top TF genes with
## larger numbers of linked RE DNA methylation sites.
metadata(
    exampleObject
)$step7LinkedDNAMethylationSiteCountHistograms$hypomethGplusResults$topGenes
```



`step7LinkedDNAMethylationSitesMotifSearching`: Perform motif searching for transcription factor motifs in the vicinity of RE DNA methylation sites linked to the specified transcription factors

To run the motif searching function, it is necessary to provide position weight matrices (PWMs), which represent DNA binding motifs for the TFs of interest in a named list, with each of the PWMs named after the TFs they represent in the TENET dataset.

The easiest way to get PWMs is to use the MotifDb package to search for available PWMs for a given TF. In this example, we search for PWMs which are available for the FOXA1 and ESR1 TFs.

```
## View the first few available motifs for the FOXM1 TF
head(names(MotifDb::query(MotifDb::MotifDb, "FOXA1")))
#> See system.file("LICENSE", package="MotifDb") for use restrictions.
#> [1] "Hsapiens-SwissRegulon-FOXA1.SwissRegulon"
#> [2] "Hsapiens-HOCOMOCOv10-FOXA1_HUMAN.H10MO.A"
#> [3] "Mmusculus-HOCOMOCOv10-FOXA1_MOUSE.H10MO.B"
#> [4] "Hsapiens-HOCOMOCOv11-core-A-FOXA1_HUMAN.H11MO.0.A"
#> [5] "NA-HOMER-FOXA1:AR(Forkhead,NR)/LNCAP-AR-ChIP-Seq(GSE27824)/Homer"
#> [6] "NA-HOMER-FOXA1(Forkhead)/LNCAP-FOXA1-ChIP-Seq(GSE27824)/Homer"

## View the first few available motifs for the ESR1 TF
head(names(MotifDb::query(MotifDb::MotifDb, "ESR1")))
#> [1] "Hsapiens-SwissRegulon-ESR1.SwissRegulon"
#> [2] "Hsapiens-HOCOMOCOv10-ESR1_HUMAN.H10MO.A"
#> [3] "Hsapiens-HOCOMOCOv10-ESR1_HUMAN.H10MO.S"
#> [4] "Mmusculus-HOCOMOCOv10-ESR1_MOUSE.H10MO.B"
#> [5] "Hsapiens-HOCOMOCOv11-core-A-ESR1_HUMAN.H11MO.0.A"
#> [6] "Hsapiens-HOCOMOCOv11-secondary-A-ESR1_HUMAN.H11MO.1.A"
```

Next, we create a named list using the human HOCOMOCO v11 core motif PWMs available for both TFs, which will be used when running the `step7LinkedDNAMethylationSitesMotifSearching` function.

```
## The human HOCOMOCO v11 core motif is the 3rd listed for FOXA1, and 4th for
## ESR1
TFMotifList <- list(
    "FOXA1" = MotifDb::query(MotifDb::MotifDb, "FOXA1")[[3]],
    "ESR1" = MotifDb::query(MotifDb::MotifDb, "ESR1")[[4]]
)

TFMotifList
#> $FOXA1
#>             1           2           3           4           5          6
#> A 0.020202020 0.208754209 0.001122334 0.001122334 0.005611672 0.48035915
#> C 0.002244669 0.005611672 0.011223345 0.005611672 0.001122334 0.01010101
#> G 0.003367003 0.780022447 0.003367003 0.020202020 0.078563412 0.47586981
#> T 0.974186308 0.005611672 0.984287318 0.973063973 0.914702581 0.03367003
#>             7           8          9         10        11
#> A 0.012345679 0.283950617 0.06509540 0.40291807 0.1257015
#> C 0.821548822 0.131313131 0.36251403 0.01122334 0.1481481
#> G 0.003367003 0.003367003 0.06285073 0.04040404 0.4736251
#> T 0.162738496 0.581369248 0.50953984 0.54545455 0.2525253
#>
#> $ESR1
#>            1          2          3          4           5          6          7
#> A 0.52571429 0.13142857 0.01714286 0.01523810 0.003809524 0.90476190 0.07238095
#> C 0.10095238 0.02857143 0.01333333 0.03809524 0.914285714 0.02666667 0.40190476
#> G 0.32190476 0.72571429 0.95809524 0.12190476 0.055238095 0.01714286 0.39428571
#> T 0.05142857 0.11428571 0.01142857 0.82476190 0.026666667 0.05142857 0.13142857
#>           8         9         10         11        12         13         14
#> A 0.2000000 0.1447619 0.08190476 0.07809524 0.5219048 0.03809524 0.03428571
#> C 0.2609524 0.2533333 0.12952381 0.11619048 0.2457143 0.81142857 0.86095238
#> G 0.3980952 0.4247619 0.03238095 0.78476190 0.1295238 0.06476190 0.03428571
#> T 0.1409524 0.1771429 0.75619048 0.02095238 0.1028571 0.08571429 0.07047619
#>          15        16        17        18
#> A 0.1257143 0.1104762 0.2895238 0.1733333
#> C 0.2476190 0.2171429 0.1657143 0.3257143
#> G 0.0247619 0.4533333 0.3961905 0.3638095
#> T 0.6019048 0.2190476 0.1485714 0.1371429
```

Finally, we run the `step7LinkedDNAMethylationSitesMotifSearching` function to search for the selected FOXA1 and ESR1 motifs in the vicinity of identified hypomethylated RE DNA methylation sites linked to the RE DNA methylation sites found to be linked to those TFs in the TENET analyses that were performed earlier. A threshold of 80% is chosen to assess motif accuracy to surrounding DNA sequences within 100 base pairs of RE DNA methylation sites. Increasing the threshold value makes the search more strict, and reduces the number of motifs found, while decreasing this value does the opposite. Also, longer motifs will tend to have fewer matches than shorter motifs.

Note: Motif searching can take some time. It is highly recommended to run this function with multiple cores if possible.

```
exampleObject <- step7LinkedDNAMethylationSitesMotifSearching(
    TENETMultiAssayExperiment = exampleObject,
    TFMotifList = TFMotifList,
    matchPWMMinScore = "80%"
)

## For each analysis type and TF, a seqLogo diagram of the chosen PWM and two
## data frames with information on the found motifs in the vicinity of RE
## DNA methylation sites, and total motifs found per RE DNA methylation site
## linked to those TFs, respectively, are saved in the metadata of the returned
## MultiAssayExperiment object under the
## step7LinkedDNAMethylationSitesMotifSearching list
names(
    metadata(
        exampleObject
    )$step7LinkedDNAMethylationSitesMotifSearching$hypomethGplusResults$FOXA1
)

## View the motif occurrences near hypomethylated RE DNA methylation sites
## linked to the FOXA1 TF
head(
    metadata(
        exampleObject
    )$step7LinkedDNAMethylationSitesMotifSearching$hypomethGplusResults$
        FOXA1$DNAMethylationSiteMotifOccurrences
)

## Also see the total number of motifs found in the vicinity of each RE DNA
## methylation site
head(
    metadata(
        exampleObject
    )$step7LinkedDNAMethylationSitesMotifSearching$hypomethGplusResults$
        FOXA1$totalMotifOccurrencesPerREDNAMethylationSite
)
```

`step7SelectedDNAMethylationSitesCaseVsControlBoxplots`: Generate boxplots comparing the methylation level of the specified RE DNA methylation sites in case and control samples

We begin this example by identifying some RE DNA methylation sites for which to generate methylation boxplots.

First, we look at the top genes by number of linked hypomethylated RE DNA methylation sites.

```
head(
    metadata(
        exampleObject
    )$step6DNAMethylationSitesPerGeneTabulation$hypomethGplusResults
)
#>                          geneID count geneName
#> ENSG00000129514 ENSG00000129514   331    FOXA1
#> ENSG00000124664 ENSG00000124664   243    SPDEF
#> ENSG00000107485 ENSG00000107485   170    GATA3
#> ENSG00000091831 ENSG00000091831   165     ESR1
#> ENSG00000118513 ENSG00000118513   139      MYB
#> ENSG00000100219 ENSG00000100219    73     XBP1
```

Next, we retrieve hypomethylated RE DNA methylation sites linked to the FOXA1 (ENSG00000129514) TF. They can be acquired from the output of the step5OptimizeLinks function.

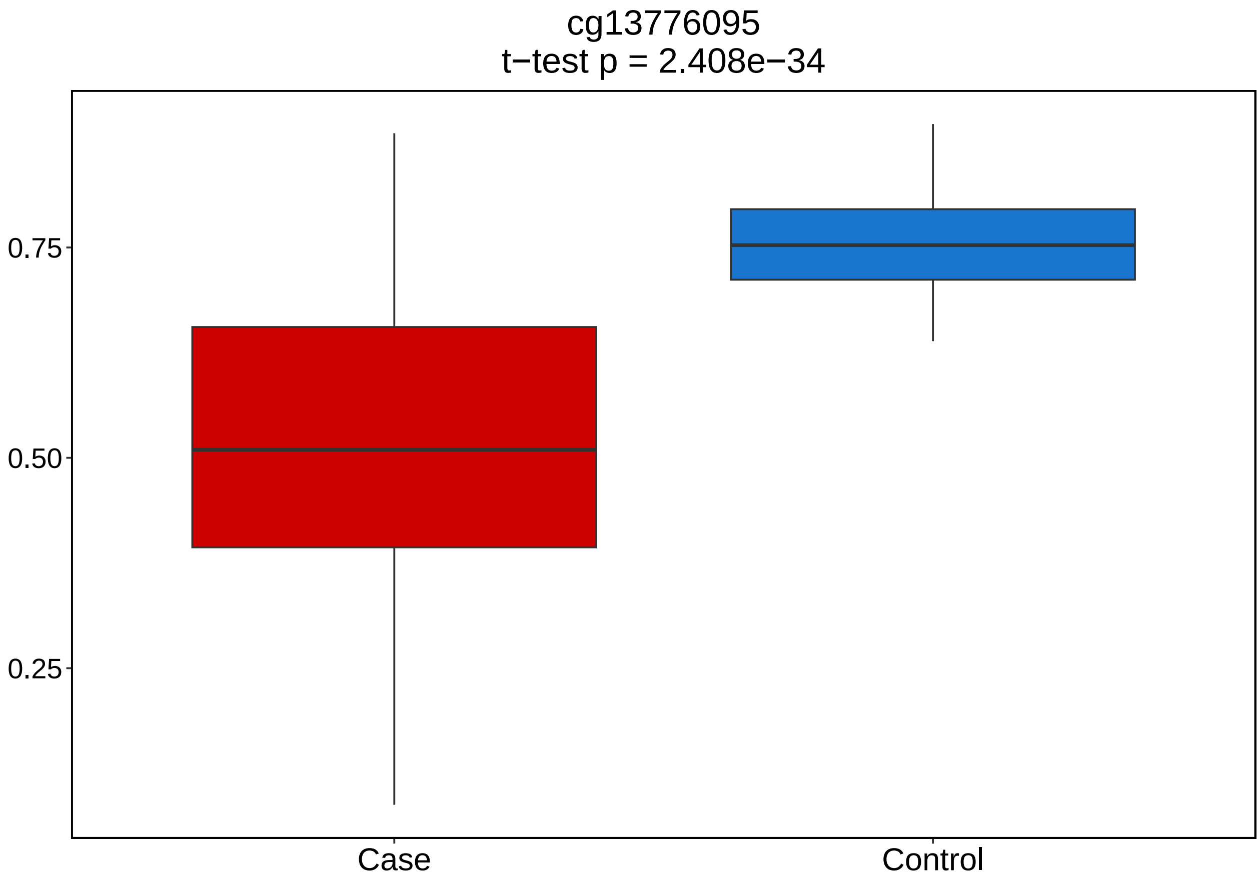
```
DNAMethylationSites <- subset(
    metadata(
        exampleObject
    )$step5OptimizeLinks$hypomethGplusResults,
    geneID == "ENSG00000129514"
)$DNAMethylationSiteID
head(DNAMethylationSites)
#> [1] "cg00002190" "cg00002809" "cg00047815" "cg00051307" "cg00069003"
#> [6] "cg00085256"
```

Finally, we generate boxplots for the selected RE DNA methylation sites.

```
exampleObject <- step7SelectedDNAMethylationSitesCaseVsControlBoxplots(
    TENETMultiAssayExperiment = exampleObject,
    DNAMethylationSites = DNAMethylationSites
)

## Each plot is saved under the ID of the RE DNA methylation site and included
## in the metadata of the returned MultiAssayExperiment object under the
## step7SelectedDNAMethylationSitesCaseVsControlBoxplots list
head(names(
    metadata(
        exampleObject
    )$step7SelectedDNAMethylationSitesCaseVsControlBoxplots
))
#> [1] "cg00002190" "cg00002809" "cg00047815" "cg00051307" "cg00069003"
#> [6] "cg00085256"
```

As an example, we examine the boxplot for the RE DNA methylation site with the ID cg13776095.

```
## Note: There may be a warning for "rows containing non-finite values" if there
## are any samples lacking methylation data for the RE DNA methylation site.
metadata(
    exampleObject
)$step7SelectedDNAMethylationSitesCaseVsControlBoxplots$cg13776095
```



`step7StatesForLinks`: Identify which of the case samples harbor each of the identified regulatory element DNA methylation site-gene links

This function identifies which of the samples provided in the dataset likely harbor each of the RE DNA methylation site-gene links. This is accomplished by examining if the RE DNA methylation site in a given link is hyper- or hypomethylated in a given case sample, and if expression of the gene in the link for that case sample is significantly less or greater than, respectively, mean expression of the gene in the control samples. This is potentially helpful by identifying case samples for further analyses.

```
## Calculate potential link status for case samples for hypomethylated RE DNA
## methylation site-gene links
exampleObject <- step7StatesForLinks(
    TENETMultiAssayExperiment = exampleObject,
    hypomethGplusAnalysis = TRUE,
    hypermethGplusAnalysis = FALSE
)

## Results are included in the metadata of the returned MultiAssayExperiment
## object under the step7StatesForLinks list.
## A single data frame is returned for each analysis type, with the combined RE
## DNA methylation site ID and gene ID from each link in the rows, and each case
## sample in the columns.
dim(
    metadata(
        exampleObject
    )$step7StatesForLinks$hypomethGplusResults
)
#> [1] 3479  200

## Show the results for the first 6 case samples. 1 indicates that a given case
## sample might harbor a given RE DNA methylation site-gene link, and 0
## indicates that it does not. NA values are shown for samples that lack
## methylation data for the site or expression data for the gene.
head(
    metadata(
        exampleObject
    )$step7StatesForLinks$hypomethGplusResults[
        , seq_len(6)
    ]
)
#>                              TCGA-5L-AAT0-01A-12R-A41B-07
#> cg06051912_ENSG00000001167                             0
#> cg04134755_ENSG00000006194                             0
#> cg04301738_ENSG00000006704                             0
#> cg04824378_ENSG00000006704                            NA
#> cg14986222_ENSG00000006704                             0
#> cg03025986_ENSG00000007372                             0
#>                              TCGA-A1-A0SK-01A-12R-A084-07
#> cg06051912_ENSG00000001167                             0
#> cg04134755_ENSG00000006194                             0
#> cg04301738_ENSG00000006704                             0
#> cg04824378_ENSG00000006704                            NA
#> cg14986222_ENSG00000006704                             0
#> cg03025986_ENSG00000007372                             1
#>                              TCGA-A2-A0CO-01A-13R-A22K-07
#> cg06051912_ENSG00000001167                             0
#> cg04134755_ENSG00000006194                             0
#> cg04301738_ENSG00000006704                             0
#> cg04824378_ENSG00000006704                             0
#> cg14986222_ENSG00000006704                             0
#> cg03025986_ENSG00000007372                             0
#>                              TCGA-A2-A0CR-01A-11R-A22K-07
#> cg06051912_ENSG00000001167                             0
#> cg04134755_ENSG00000006194                             0
#> cg04301738_ENSG00000006704                             0
#> cg04824378_ENSG00000006704                             0
#> cg14986222_ENSG00000006704                             0
#> cg03025986_ENSG00000007372                             0
#>                              TCGA-A2-A0SU-01A-11R-A084-07
#> cg06051912_ENSG00000001167                             0
#> cg04134755_ENSG00000006194                             0
#> cg04301738_ENSG00000006704                             0
#> cg04824378_ENSG00000006704                             0
#> cg14986222_ENSG00000006704                             0
#> cg03025986_ENSG00000007372                             0
#>                              TCGA-A2-A0SX-01A-12R-A084-07
#> cg06051912_ENSG00000001167                             0
#> cg04134755_ENSG00000006194                             0
#> cg04301738_ENSG00000006704                             0
#> cg04824378_ENSG00000006704                            0
#> cg14986222_ENSG00000006704                            0
#> cg03025986_ENSG00000007372                            0
```

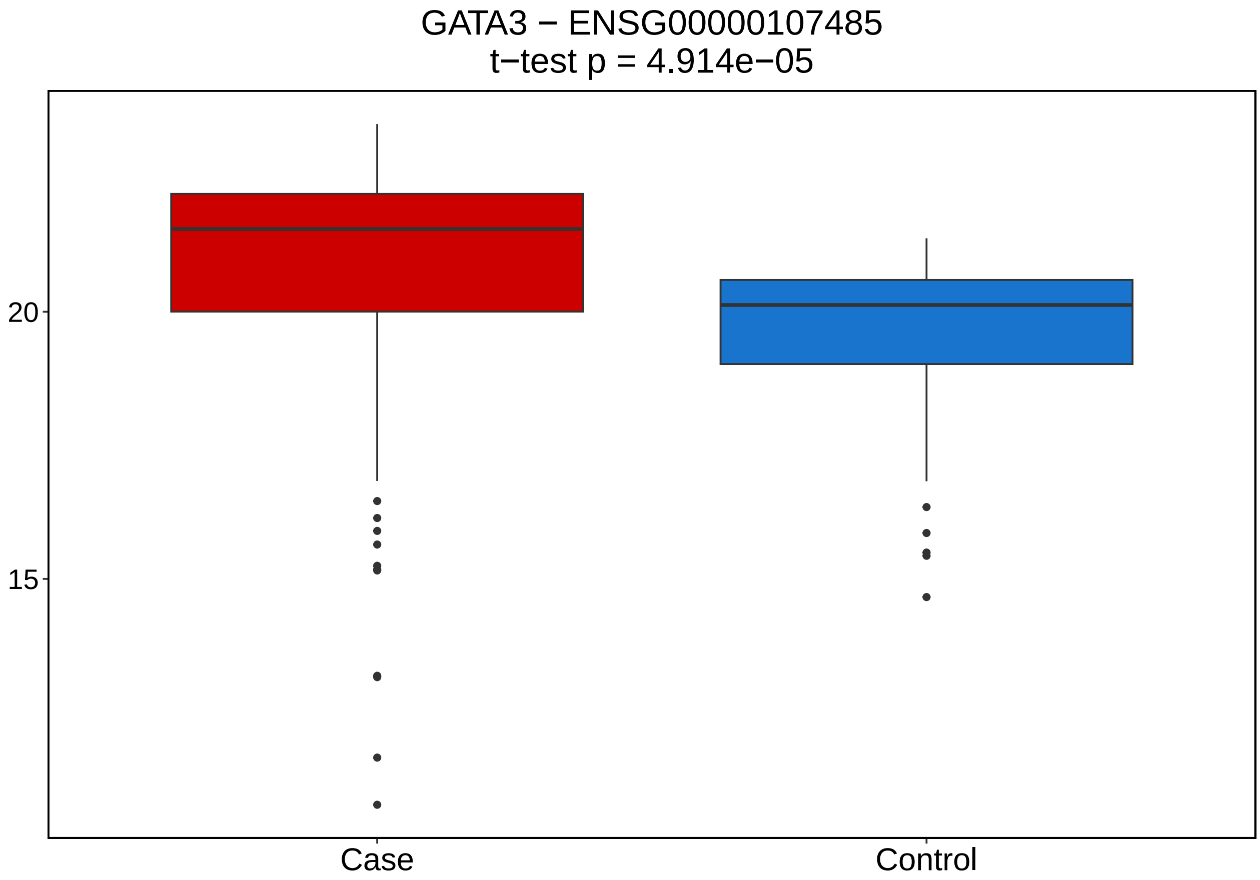
step7TopGenesCaseVsControlExpressionBoxplots: Create boxplots comparing the expression level of the top genes/transcription factors in case and control samples

This function generates boxplots comparing the expression of the top genes/TFs by number of linked RE DNA methylation sites in the case versus control samples.

```
## Run the step7TopGenesCaseVsControlExpressionBoxplots function.
## Since we performed analyses only using TFs in the step 3 function, the
## top genes are all TFs, so a message that separate output for
## TFs will be skipped is displayed.
exampleObject <- step7TopGenesCaseVsControlExpressionBoxplots(
    TENETMultiAssayExperiment = exampleObject,
    hypomethGplusAnalysis = TRUE,
    hypermethGplusAnalysis = FALSE,
    topGeneNumber = 10
)
#> All genes with hypomethylated G+ links are TFs, so the separate output for TFs will be
↪  skipped.

## Results are included in the metadata of the returned MultiAssayExperiment
## object under the step7TopGenesCaseVsControlExpressionBoxplots list.
## For each analysis type, results are included in sub-lists, each of which
## contains lists with results for topGenes and topTFs, unless the
## top genes are all TFs, in which case the separate top TFs output is skipped.
## Each boxplot is saved under its gene ID.
names(
    metadata(
        exampleObject
    )$step7TopGenesCaseVsControlExpressionBoxplots$
        hypomethGplusResults$topGenes
)
#>  [1] "ENSG00000129514" "ENSG00000124664" "ENSG00000107485" "ENSG00000091831"
#>  [5] "ENSG00000118513" "ENSG00000100219" "ENSG00000152192" "ENSG00000105261"
#>  [9] "ENSG00000178935" "ENSG00000115163"
```

As an example, we examine the boxplot for gene ENSG00000107485 (GATA3).

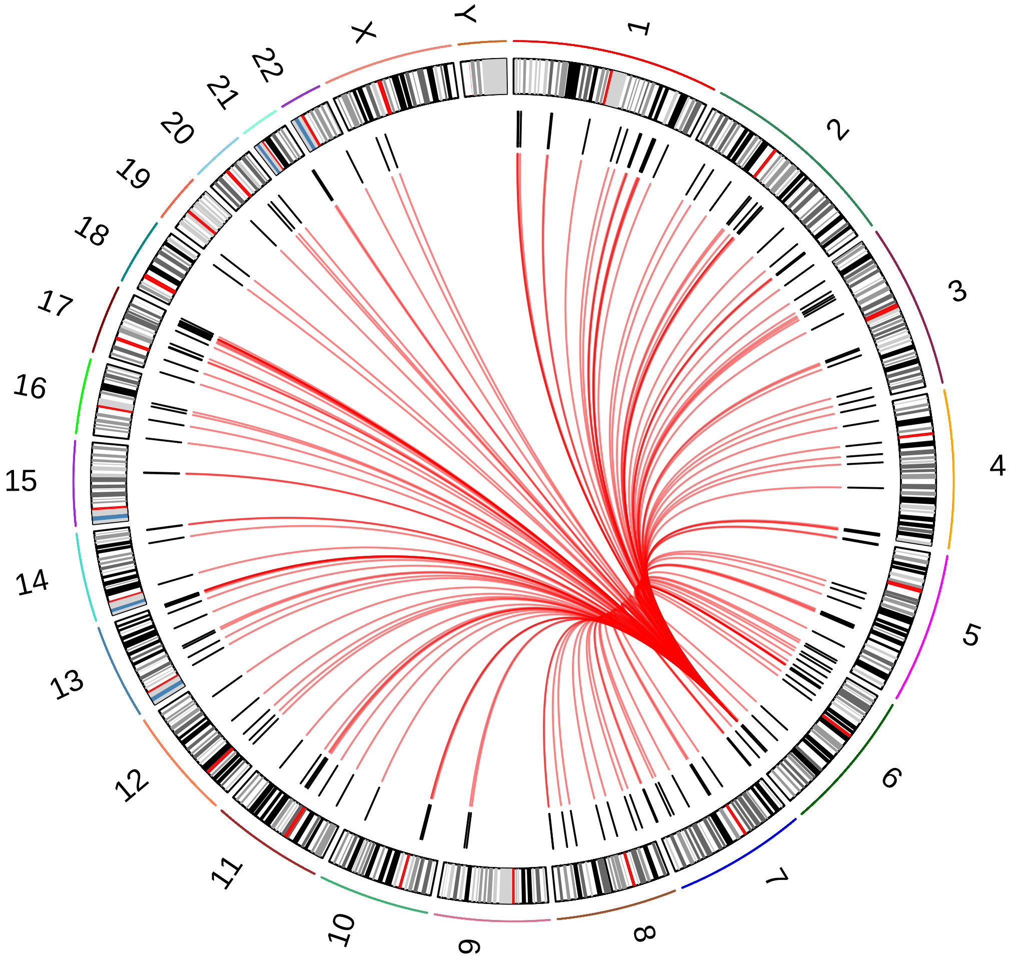
```
## Note: There may be a warning for "rows containing non-finite values" if there
## are any samples lacking expression data for the given gene.
metadata(
    exampleObject
)$step7TopGenesCaseVsControlExpressionBoxplots$hypomethGplusResults$
    topGenes$ENSG00000107485
```



`step7TopGenesCircosPlots`: Generate Circos plots displaying the links between top identified genes and each of the RE DNA methylation sites linked to them

This function generates Circos plots for each of the top genes by number of linked RE DNA methylation sites showing the links between the gene and each of its RE DNA methylation sites.

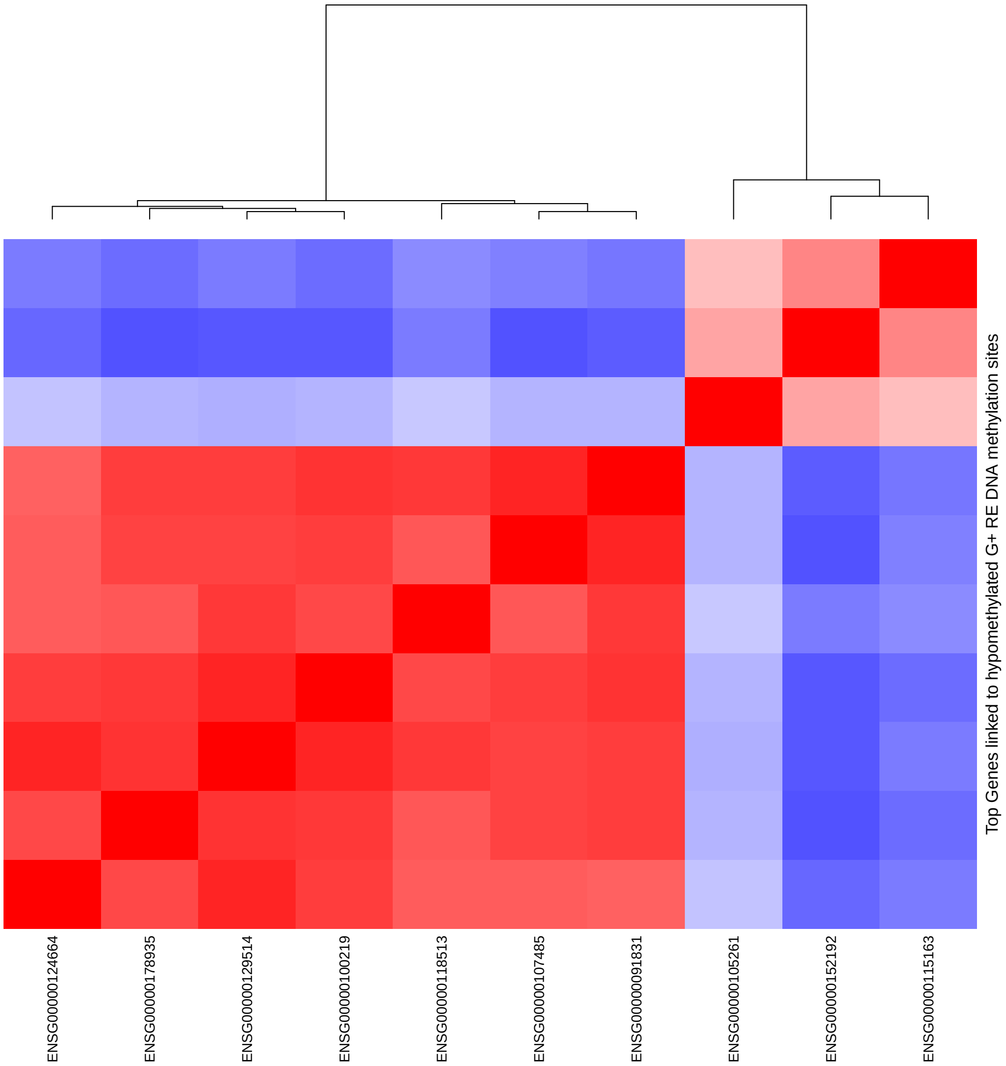
```
## Run the step7TopGenesCircosPlots function
exampleObject <- step7TopGenesCircosPlots(
    TENETMultiAssayExperiment = exampleObject,
    hypermethGplusAnalysis = FALSE,
    hypomethGplusAnalysis = TRUE,
    topGeneNumber = 10
)
#>
#> RCircos.Core.Components initialized.
#> Type ?RCircos.Reset.Plot.Parameters to see how to modify the core components.
#>
#> Note: chrom.padding 300  was reset to 1995
#> All genes with hypomethylated G+ links are TFs, so the separate output for TFs will be
↪  skipped.

## Results are included in the metadata of the returned MultiAssayExperiment
## object under the step7TopGenesCircosPlots list.
## For each analysis type, results are included in sub-lists, each
## of which contains lists with results for topGenes and topTFs, unless the
## top genes are all TFs, in which case the separate top TFs output is skipped.
## Each Circos plot is saved under its gene ID.

## Note: Since we performed analyses only using TFs in the step 3 function,
## the top genes are all TFs, so topTFs will be NA here.
names(
    metadata(
        exampleObject
    )$step7TopGenesCircosPlots$hypomethGplusResults$topGenes
)
#>  [1] "ENSG00000129514" "ENSG00000124664" "ENSG00000107485" "ENSG00000091831"
#>  [5] "ENSG00000118513" "ENSG00000100219" "ENSG00000152192" "ENSG00000105261"
#>  [9] "ENSG00000178935" "ENSG00000115163"

## Display an example Circos plot for ENSG00000091831 (ESR1).
## Note: Plots may take some time to display.
metadata(
    exampleObject
)$step7TopGenesCircosPlots$hypomethGplusResults$topGenes$ENSG00000091831
```



`step7TopGenesExpressionCorrelationHeatmaps`: Generate mirrored heatmaps displaying the correlation of the expression values of the top genes/TFs

This function generates heatmaps displaying the correlation of the expression of each of the top genes in the case samples. Each of the top genes is displayed in both the rows and columns, so the heatmaps are mirrored, with correlation values of each gene to itself displayed in a diagonal line up the center of the heatmaps. Red values represent positive correlation and blue values represent negative correlation, with darker colors representing a stronger correlation. Dendrograms are included to identify genes which are closely related in expression correlation.

```
## Run the step7TopGenesExpressionCorrelationHeatmaps function
exampleObject <- step7TopGenesExpressionCorrelationHeatmaps(
    TENETMultiAssayExperiment = exampleObject,
    hypermethGplusAnalysis = FALSE,
    hypomethGplusAnalysis = TRUE,
    topGeneNumber = 10
)
#> All genes with hypomethylated G+ links are TFs, so the separate output for TFs will be
↪  skipped.

## Results are included in the metadata of the returned MultiAssayExperiment
## object under the step7TopGenesExpressionCorrelationHeatmaps list.
## For each analysis type, results are included in sub-lists, each
## of which contains lists with results for the topGenes and topTFs, unless
## the top genes are all TFs, in which case the separate top TFs output is
## skipped. For each of these, the heatmap is generated along with a data frame
## with the correlation values displayed in the heatmap.

## Display the mirrored heatmap.
## Note: Since we performed analyses only using TFs in the step 3 function,
## the top genes are all TFs, so topTFs will be NA here.
metadata(
    exampleObject
)$step7TopGenesExpressionCorrelationHeatmaps$hypomethGplusResults$
    topGenes$heatmap
```



```
## Display the data frame with correlation values
head(
    metadata(
        exampleObject
    )$step7TopGenesExpressionCorrelationHeatmaps$hypomethGplusResults$
        topGenes$correlationMatrix
)
#>                 geneNames ENSG00000124664 ENSG00000178935 ENSG00000129514
#> ENSG00000115163     CENPA      -0.5059431      -0.5772885      -0.5148119
#> ENSG00000152192    POU4F1      -0.5836921      -0.6689745      -0.6599532
#> ENSG00000105261     OVOL3      -0.2273260      -0.2926892      -0.3050159
#> ENSG00000091831      ESR1       0.6108950       0.7584950       0.7513023
#> ENSG00000107485     GATA3       0.6261998       0.7228711       0.7305243
#> ENSG00000118513       MYB       0.6309581       0.6562566       0.7765156
#>                 ENSG00000100219 ENSG00000118513 ENSG00000107485 ENSG00000091831
#> ENSG00000115163      -0.5776517      -0.4500729      -0.4954987      -0.5236431
#> ENSG00000152192      -0.6560849      -0.5001467      -0.6705779      -0.6320355
#> ENSG00000105261      -0.2921732      -0.2061876      -0.2989210      -0.2969264
#> ENSG00000091831       0.7941408       0.7727373       0.8413892       1.0000000
#> ENSG00000107485       0.7560965       0.6429943       1.0000000       0.8413892
#> ENSG00000118513       0.7006877       1.0000000       0.6429943       0.7727373
#>                 ENSG00000105261 ENSG00000152192 ENSG00000115163
#> ENSG00000115163       0.2415614       0.4619340       1.0000000
#> ENSG00000152192       0.3517048       1.0000000       0.4619340
#> ENSG00000105261       1.0000000       0.3517048       0.2415614
#> ENSG00000091831      -0.2969264      -0.6320355      -0.5236431
#> ENSG00000107485      -0.2989210      -0.6705779      -0.4954987
#> ENSG00000118513      -0.2061876      -0.5001467      -0.4500729
```

`step7TopGenesDNAMethylationHeatmaps`: Generate heatmaps displaying the methylation level of all RE DNA methylation sites linked to the top genes/transcription factors, along with the expression of those genes in the column headers, in the case samples within the supplied MultiAssayExperiment object

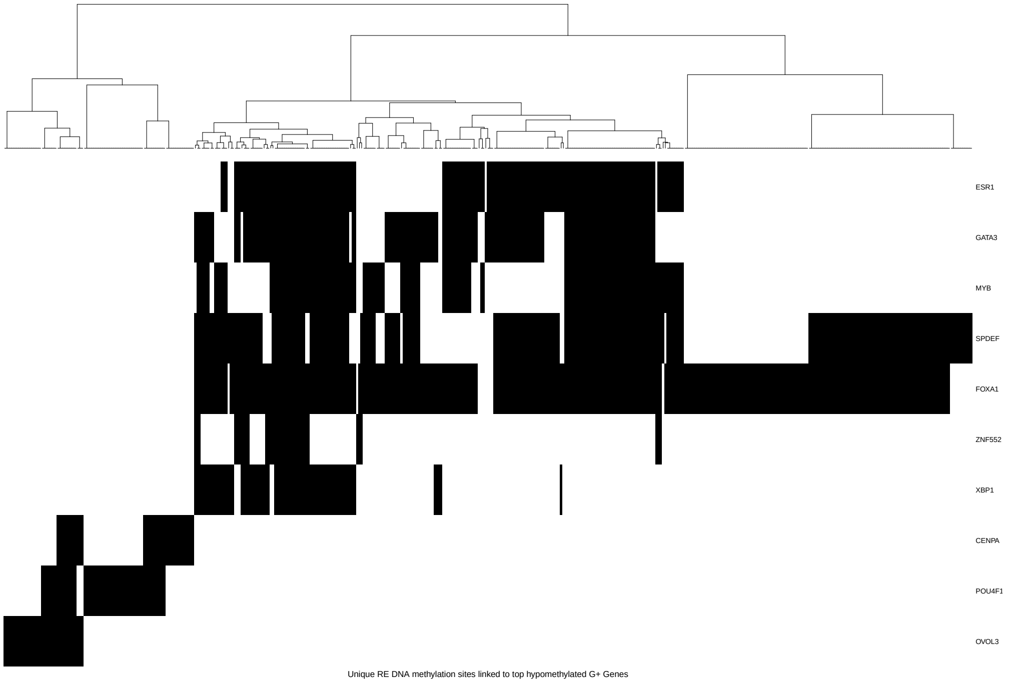
This function creates heatmaps displaying the methylation of unique RE DNA methylation sites linked to the top genes in the main body of the heatmaps, as well as a smaller heatmap showing expression of the top genes labeling the columns. Expression/methylation for each case sample is shown per column, while expression of each of the top genes or methylation of their linked RE DNA methylation sites is shown in the rows. Warm colors represent relatively higher expression/methylation levels, while cold colors represent relatively lower expression/methylation levels. These are determined per gene/RE DNA methylation site, and are not comparable between genes/RE DNA methylation sites, only between samples. Column dendrograms are included to identify subsets of the case samples which display particular expression or methylation patterns in the top genes and their linked RE DNA methylation sites.

```
## Run the step7TopGenesDNAMethylationHeatmaps function
exampleObject <- step7TopGenesDNAMethylationHeatmaps(
    TENETMultiAssayExperiment = exampleObject,
    hypermethGplusAnalysis = FALSE,
    hypomethGplusAnalysis = TRUE,
    topGeneNumber = 10
)
#> All genes with hypomethylated G+ links are TFs, so the separate output for TFs will be
↪  skipped.

## Results are included in the metadata of the returned MultiAssayExperiment
## object under the step7TopGenesDNAMethylationHeatmaps list.
## For each analysis type, results are included in sub-lists, each
## of which contains heatmaps for the topGenes and topTFs, unless the
## top genes are all TFs, in which case the separate top TFs output is skipped.

## Note: Since we performed analyses only using TFs in the step 3 function,
## the top genes are all TFs, so topTFs will be NA here.
names(metadata(
    exampleObject
)$step7TopGenesDNAMethylationHeatmaps$hypomethGplusResults)
#> [1] "topGenes" "topTFs"
```

step7TopGenesOverlappingLinkedDNAMethylationSitesHeatmaps: Generate binary heatmaps displaying which of the top genes/transcription factors share links with each of the unique regulatory element DNA methylation sites linked to at least one top gene/TF

These binary heatmaps provide a visual representation of which of the top genes each RE DNA methylation site is linked to, as RE DNA methylation sites can be linked to multiple genes. RE DNA methylation sites are displayed in the columns, while the top genes are displayed in the rows. Black indicates that a given RE DNA methylation site is linked to that gene, and white indicates it is not. Dendrograms are included to identify blocks of RE DNA methylation sites that are linked to similar genes.

```
## Run the step7TopGenesOverlappingLinkedDNAMethylationSitesHeatmaps function
exampleObject <- step7TopGenesOverlappingLinkedDNAMethylationSitesHeatmaps(
    TENETMultiAssayExperiment = exampleObject,
    hypermethGplusAnalysis = FALSE,
    hypomethGplusAnalysis = TRUE,
    topGeneNumber = 10
)
#> All genes with hypomethylated G+ links are TFs, so the separate output for TFs will be
↪  skipped.

## Results are included in the metadata of the returned MultiAssayExperiment
## object under the step7TopGenesOverlappingLinkedDNAMethylationSitesHeatmaps
## list. For each analysis type, results are included in sub-lists, each of
## which contains lists with results for the topGenes and topTFs, unless
## the top genes are all top TFs,  in which case the separate top TFs output is
## skipped. For each of these, the heatmap is generated along with a data frame
## with the correlation values displayed in the heatmap.

## Display the binary heatmap.
## Note: Since we performed analyses only using TFs in the step 3 function,
## the top genes are all TFs, so topTFs will be NA here.
metadata(
    exampleObject
)$step7TopGenesOverlappingLinkedDNAMethylationSitesHeatmaps$
    hypomethGplusResults$topGenes$heatmap
```



```r
## Display a subset of the data frame noting the presence/absence of links.
## 1 indicates a link, while 0 indicates no link.
head(
    metadata(
        exampleObject
    )$step7TopGenesOverlappingLinkedDNAMethylationSitesHeatmaps$
        hypomethGplusResults$topGenes$linkTable[
        , seq_len(6)
    ]
)
#>        cg20627754 cg19367933 cg17837127 cg17418276 cg15473218 cg14136260
#> ESR1            0          0          0          0          0          0
#> GATA3           0          0          0          0          0          0
#> MYB             0          0          0          0          0          0
#> SPDEF           0          0          0          0          0          0
#> FOXA1           0          0          0          0          0          0
#> ZNF552          0          0          0          0          0          0
```

`step7TopGenesSurvival`: Perform Kaplan-Meier and Cox regression analyses to assess the association of top gene expression and linked RE DNA methylation site methylation with patient survival

This function uses the survival status and time for each of the samples in the dataset to perform survival analyses on the expression of the top genes and methylation of their linked RE DNA methylation sites. For each gene and RE DNA methylation site, a Kaplan-Meier survival plot can be generated, and statistics from both Kaplan-Meier and univariate Cox regression analyses are output.

First, we load the example survival status and survival time data from the `exampleTENETClinicalDataFrame` object.

```r
## Load the exampleTENETClinicalDataFrame object from the TENET.ExperimentHub
## package. It contains the vital_status (survival status) and time (survival
## time) data for each sample in the exampleTENETMultiAssayExperiment
exampleTENETClinicalDataFrame <-
    TENET.ExperimentHub::exampleTENETClinicalDataFrame()
#> see ?TENET.ExperimentHub and browseVignettes('TENET.ExperimentHub') for documentation
#> loading from cache
vitalStatusData <- subset(
    exampleTENETClinicalDataFrame,
    select = "vital_status"
)
survivalTimeData <- subset(exampleTENETClinicalDataFrame, select = "time")
```

The vital status dataset is a data frame with rownames representing sample names and the first column representing the vital status. Sample values are either “alive” or “dead” (case-insensitive) or 1 or 2, indicating that samples were collected from a patient who was alive/censored or dead/reached the outcome of interest, respectively.

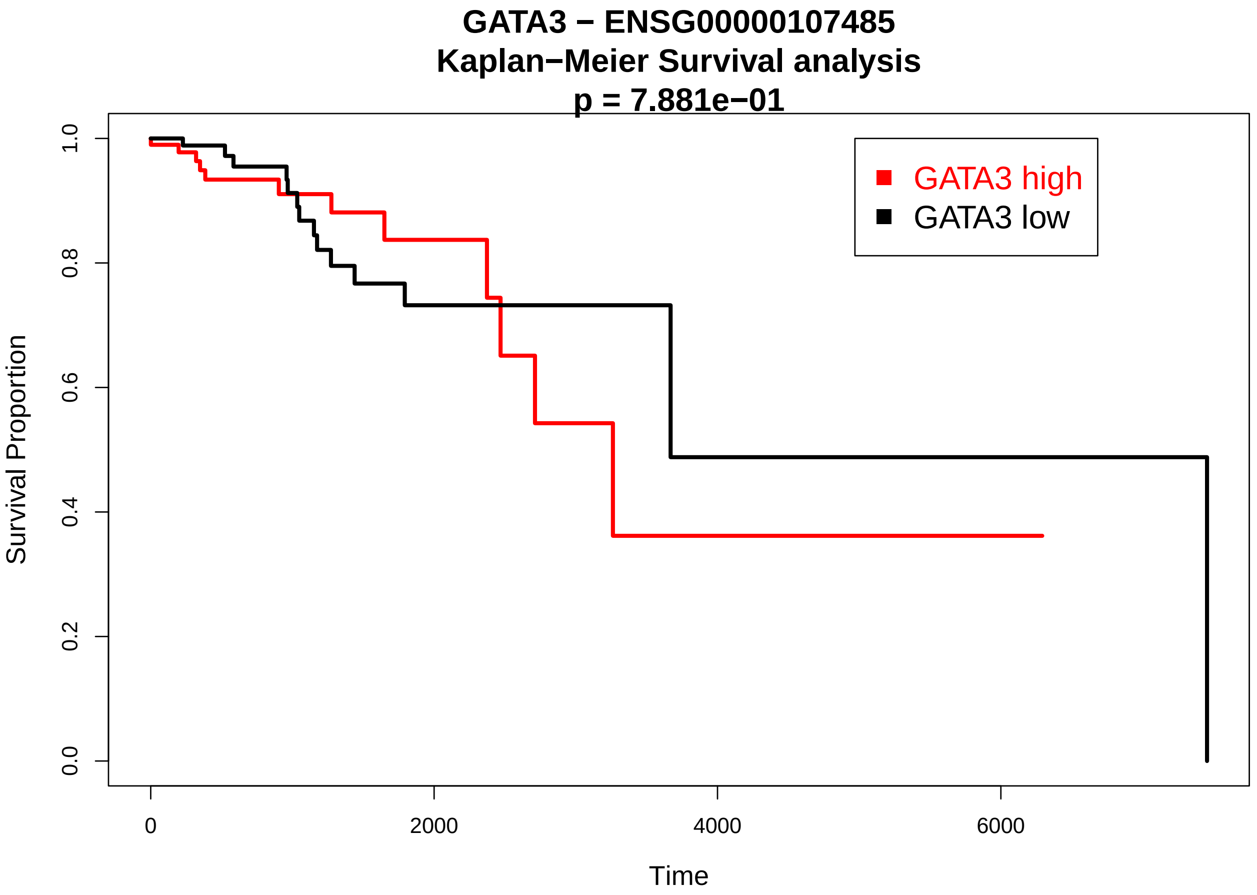
Similarly, the survival time dataset is a data frame with rownames representing sample names and the first column representing the survival time of the patient the sample was derived from.

```r
## Show the vital status data
head(vitalStatusData)
#>              vital_status
#> TCGA-5L-AAT0        Alive
#> TCGA-A1-A0SK         Dead
#> TCGA-A2-A0CO         <NA>
#> TCGA-A2-A0CR        Alive
#> TCGA-A2-A0SU        Alive
#> TCGA-A2-A0SX        Alive

## Show the survival time data
head(survivalTimeData)
#>              time
#> TCGA-5L-AAT0 1477
#> TCGA-A1-A0SK  967
#> TCGA-A2-A0CO   NA
#> TCGA-A2-A0CR 3283
#> TCGA-A2-A0SU 1352
#> TCGA-A2-A0SX 1288
```

Next, we perform the survival analysis using the vital status and survival time data.

```r
## Since we performed analyses only using TFs in the step 3 function, the
## top genes are all TFs, so a message that separate output for
## TFs will be skipped is displayed.
exampleObject <- step7TopGenesSurvival(
    TENETMultiAssayExperiment = exampleObject,
    hypermethGplusAnalysis = FALSE,
    hypomethGplusAnalysis = TRUE,
    vitalStatusData = vitalStatusData,
    survivalTimeData = survivalTimeData,
    topGeneNumber = 10,
    generatePlots = TRUE
)
```

```
#> All genes with hypomethylated G+ links are TFs, so the separate output for TFs will be
↪  skipped.
#> All genes with hypomethylated G+ links are TFs, so the separate output for TFs will be
↪  skipped.
#> All genes with hypomethylated G+ links are TFs, so the separate output for TFs will be
↪  skipped.
#> All genes with hypomethylated G+ links are TFs, so the separate output for TFs will be
↪  skipped.

## Results are included in the metadata of the returned MultiAssayExperiment
## object under the step7TopGenesSurvival list.
## For each analysis type, results are included in sub-lists, each
## of which contains lists with results for topGenes and topTFs, unless the
## top genes are all TFs, in which case the separate top TFs output is skipped.
## Each includes two data frames with the survival statistics for Kaplan-Meier
## and Cox regression survival analyses, and if the generatePlots
## argument is TRUE, topGenesSurvivalPlots and topMethylationSitesSurvivalPlots
## lists are included which contain the Kaplan-Meier survival plots for the top
## genes and each of their unique linked RE DNA methylation sites, respectively.
names(
    metadata(
        exampleObject
    )$step7TopGenesSurvival$hypomethGplusResults$topGenes
)
#> [1] "topGenesSurvivalStats"               "topGenesSurvivalPlots"
#> [3] "topDNAMethylationSitesSurvivalStats" "topDNAMethylationSitesSurvivalPlots"

## The topGenesSurvivalStats and topMethylationSitesSurvivalStats variables are
## data frames containing survival statistics.
## Note: A significant amount of data is output, so selected values are shown
## here.
head(
    metadata(
        exampleObject
    )$step7TopGenesSurvival$hypomethGplusResults$
        topGenes$topGenesSurvivalStats[
        , c(1:2, 15, 17, 22:24, 26)
    ]
)
#>                          geneID geneName caseMeanExpressionHighExpressionGroup
#> ENSG00000129514 ENSG00000129514    FOXA1                      20.9779047434146
#> ENSG00000124664 ENSG00000124664    SPDEF                      21.5321701392785
#> ENSG00000107485 ENSG00000107485    GATA3                       22.241449814161
#> ENSG00000091831 ENSG00000091831     ESR1                       20.107344539244
#> ENSG00000118513 ENSG00000118513      MYB                       18.869634836851
#> ENSG00000100219 ENSG00000100219     XBP1                      24.1923366370896
#>                 caseMeanExpressionLowExpressionGroup  KMSurvivalPValue
#> ENSG00000129514                     17.4049311604624 0.274250364140255
#> ENSG00000124664                     18.4482599333322 0.645194080721154
#> ENSG00000107485                     19.3431191880042 0.788128422897537
#> ENSG00000091831                     15.2375966010389 0.537171334328383
#> ENSG00000118513                     16.2760135424539 0.288356560948957
#> ENSG00000100219                     21.8681885026766 0.558622389141247
#>                  CoxRegressionCoefficient   CoxHazardRatio CoxSurvivalPValue
#> ENSG00000129514       0.0519495788237298 1.05332263140835 0.514830685267593
#> ENSG00000124664       0.0500143832133511 1.05128621714124 0.579259697615787
#> ENSG00000107485       0.0555342371138743 1.05710520868428 0.540119132486782
#> ENSG00000091831       0.0391278341592634 1.03990341033187 0.555889128738812
#> ENSG00000118513        0.018847076493218 1.01902580370213 0.868434703354218
#> ENSG00000100219       0.0307132824310266 1.03118980126401 0.821194100010203

head(
    metadata(
        exampleObject
    )$step7TopGenesSurvival$hypomethGplusResults$
        topGenes$topMethylationSitesSurvivalStats[
        , c(1, 24, 26, 31:33, 35)
    ]
)
#> NULL

## Show the names of the gene survival plots
names(
    metadata(
        exampleObject
    )$step7TopGenesSurvival$hypomethGplusResults$
        topGenes$topGenesSurvivalPlots
)
#>  [1] "ENSG00000129514" "ENSG00000124664" "ENSG00000107485" "ENSG00000091831"
#>  [5] "ENSG00000118513" "ENSG00000100219" "ENSG00000152192" "ENSG00000105261"
#>  [9] "ENSG00000178935" "ENSG00000115163"

## Plot the Kaplan-Meier survival plot for GATA3 (ENSG00000107485) as an example
metadata(
    exampleObject
)$step7TopGenesSurvival$hypomethGplusResults$
    topGenes$topGenesSurvivalPlots$ENSG00000107485
```



`step7TopGenesTADTables`: Create tables using user-supplied topologically associating domain (TAD) information which identify the topologically associating domains containing each RE DNA methylation site linked to the top genes/transcription factors, as well as other genes in the same topologically associating domain as potential downstream targets

This function requires the user to supply either a path to a directory containing bed-like files with TAD regions of interest, or a single TAD object given as a GRanges, data frame, or matrix object, as seen in the example below. To illustrate the use of this function, we will use an example TAD dataset from the TENET.ExperimentHub package.

```
## Load the example TAD dataset from the TENET.ExperimentHub package
exampleTADRegions <- TENET.ExperimentHub::exampleTENETTADRegions()
#> see ?TENET.ExperimentHub and browseVignettes('TENET.ExperimentHub') for documentation
#> loading from cache

## TAD files for this function must include the chromosome of each TAD region
## in the first column, and the start and end positions of each in the second
## and third columns respectively. Additional columns can be included but
## are not considered in this function.
class(exampleTADRegions)
#> [1] "GRanges"
#> attr(,"package")
#> [1] "GenomicRanges"
head(exampleTADRegions)
```

```
#> GRanges object with 6 ranges and 0 metadata columns:
#>     seqnames           ranges strand
#>        <Rle>        <IRanges>  <Rle>
#>   1     chr1   800001-3680000      *
#>   2     chr1  3800001-6000000      *
#>   3     chr1  6520001-7640000      *
#>   4     chr1  7960001-8920000      *
#>   5     chr1  9240001-9600000      *
#>   6     chr1 9760001-10360000      *
#>   -------
#>   seqinfo: 23 sequences from an unspecified genome; no seqlengths
```

The unique RE DNA methylation sites linked to the top genes, as selected by the user, will be overlapped with the TAD files, and genes within the same TAD of each RE DNA methylation site will be recorded (as possible downstream target genes for the regulatory elements represented by those RE DNA methylation sites, for further analysis purposes).

```
## Use the example TAD object to perform TAD overlapping.
## Since we performed analyses only using TFs in the step 3 function, the
## top genes are all TFs, so a message that separate output for
## TFs will be skipped is displayed.
exampleObject <- step7TopGenesTADTables(
    TENETMultiAssayExperiment = exampleObject,
    TADFiles = exampleTADRegions,
    hypomethGplusAnalysis = TRUE,
    hypermethGplusAnalysis = FALSE,
    topGeneNumber = 10
)
#> All genes with hypomethylated G+ links are TFs, so the separate output for TFs will be
↪  skipped.

## Results are included in the metadata of the returned MultiAssayExperiment
## object under the step7TopGenesTADTables list.
## For each analysis type, results are included in sub-lists, each
## of which contains results in the form of a data frame for topGenes and
## topTFs, unless the top genes are all TFs, in which case
## the separate top TFs output is skipped.
class(
    metadata(
        exampleObject
    )$step7TopGenesTADTables$hypomethGplusResults$topGenes
)
#> [1] "data.frame"

## Display results for selected hypomethylated RE DNA methylation sites. A
## variety of data are included for each RE DNA methylation site, including its
## location, the top genes it is linked to, and information on the count and
## identities of other genes found within the same TAD of the RE DNA methylation
## site. Note: A significant amount of data is output, so selected values are
## shown here.
head(
    metadata(
        exampleObject
    )$step7TopGenesTADTables$hypomethGplusResults$topGenes[
        c(1:6, 16:17)
    ]
)
#>   DNAMethylationSiteID chromosome     start       end
#> 1           cg00002190       chr8  19697522  19697523
#> 2           cg00002809      chr17  78486271  78486272
#> 3           cg00047815      chr22  44915621  44915622
#> 4           cg00051307       chr5  73228366  73228367
#> 5           cg00069003       chr5 140784521 140784522
#> 6           cg00085256       chr2 217386723 217386724
#>   FOXA1_ENSG00000129514_linked SPDEF_ENSG00000124664_linked
#> 1                         TRUE                         TRUE
#> 2                         TRUE                         TRUE
#> 3                         TRUE                         TRUE
#> 4                         TRUE                         TRUE
#> 5                         TRUE                         TRUE
#> 6                         TRUE                         TRUE
#>   TADFile_geneCountInTAD
#> 1                      1
#> 2                      1
#> 3                      0
#> 4                      3
#> 5                      2
#> 6                     20
#>
↪   TADFile_TADGeneIDs
#> 1
↪   ENSG00000242709
#> 2
↪   ENSG00000267770
#> 3
↪   No_TAD_identified
#> 4
↪   ENSG00000157111,ENSG00000251493,ENSG00000251543
#> 5
↪   ENSG00000204969,ENSG00000248106
#> 6
↪   ENSG00000115568,ENSG00000135912,ENSG00000135929,ENSG00000138375,ENSG00000144583,ENSG00000163464,ENS
```

`step7TopGenesUCSCBedFiles`: Create bed-formatted interact files which can be loaded on the UCSC Genome Browser to display links between top genes and transcription factors and their linked RE DNA methylation sites

This function will output the interact file in the specified folder. For the purposes of this example, we will save the output file in a temporary folder. This interact file can be uploaded to the UCSC Genome Browser to visualize the links between each of the top genes/TFs and their linked RE DNA methylation sites.

```
## Get the path to a temporary directory in which to save the output interact
## file
tempDirectory <- tempdir()

## Run the step7TopGenesUCSCBedFiles function.
## Since we performed analyses only using TFs in the step 3 function, the
## top genes are all TFs, so a message that separate output for
## TFs will be skipped is displayed.
filePaths <- step7TopGenesUCSCBedFiles(
    TENETMultiAssayExperiment = exampleObject,
    outputDirectory = tempDirectory,
    hypomethGplusAnalysis = TRUE,
    hypermethGplusAnalysis = FALSE,
    topGeneNumber = 10
)
#> All genes with hypomethylated G+ links are TFs, so the separate output for TFs will be
↪  skipped.

## Unlike other functions, this function does not return the
## given TENETMultiAssayExperiment with additional information generated by the
## function in its metadata, but rather returns an object with information on
## where to find the created interact file.

## Get the path of the output file for top genes with hypomethylated G+ links
bedPath <- filePaths$hypoGplus$topGenes

## Read the first few lines of the file.
## The file largely contains information about each RE DNA methylation site-gene
## link, with additional information in the first line which allows it to be
## loaded by the UCSC Genome Browser.
cat(head(readLines(bedPath)), sep = "\n")
#> track type=interact name="TENETHypoG+Interactions" description="TENET top gene to RE
↪  DNA methylation site links"
#> chr14 37596058 37596059 ENSG00000129514_cg00002190_link 0 0 . #FF0000 chr14 37596058
↪  37596059 FOXA1 . chr8 19697522 19697523 cg00002190 .
#> chr14 37596058 37596059 ENSG00000129514_cg00002809_link 0 0 . #FF0000 chr14 37596058
↪  37596059 FOXA1 . chr17 78486271 78486272 cg00002809 .
#> chr14 37596058 37596059 ENSG00000129514_cg00047815_link 0 0 . #FF0000 chr14 37596058
↪  37596059 FOXA1 . chr22 44915621 44915622 cg00047815 .
#> chr14 37596058 37596059 ENSG00000129514_cg00051307_link 0 0 . #FF0000 chr14 37596058
↪  37596059 FOXA1 . chr5 73228366 73228367 cg00051307 .
#> chr14 37596058 37596059 ENSG00000129514_cg00069003_link 0 0 . #FF0000 chr14 37596058
↪  37596059 FOXA1 . chr5 140784521 140784522 cg00069003 .

## Delete the output file, since this is just an example.
## Do not use this line if running on real data, as it will delete your created
## file.
invisible(file.remove(unlist(bedPath)))
```

step7TopGenesUserPeakOverlap: Identify if RE DNA methylation sites linked to top genes/transcription factors are located within a specific distance of specified genomic regions

The step7TopGenesUserPeakOverlap function requires the user to supply either a path to a directory containing bed-like files with peaks of interest, or a single peak object given as a GRanges, data frame, or matrix object, as seen in the example below. To illustrate the use of this function, we will use an example peak dataset from the TENET.ExperimentHub package.

```
## Load the example peak dataset from the TENET.ExperimentHub package
examplePeakFile <- TENET.ExperimentHub::exampleTENETPeakRegions()
#> see ?TENET.ExperimentHub and browseVignettes('TENET.ExperimentHub') for documentation
#> loading from cache

## Peak files for this function must include the chromosome of each peak region
## in the first column, and the start and end positions of each peak in the
## second and third columns respectively. Additional columns can be included,
## but are not considered in this function.
class(examplePeakFile)
#> [1] "GRanges"
#> attr(,"package")
#> [1] "GenomicRanges"
head(examplePeakFile)
#> GRanges object with 6 ranges and 0 metadata columns:
#>       seqnames              ranges strand
#>          <Rle>           <IRanges>  <Rle>
#>   [1]    chr20   41650340-41650989      *
#>   [2]     chr1 147612278-147612917      *
#>   [3]    chr20   48812030-48812609      *
#>   [4]    chr15   69594337-69595180      *
#>   [5]     chr8 101607580-101608404      *
#>   [6]     chr8 125429992-125430539      *
#>   -------
#>   seqinfo: 23 sequences from an unspecified genome; no seqlengths
```

The unique RE DNA methylation sites linked to the top genes will be overlapped with the peak files, with a specified buffer region added to the RE DNA methylation sites, so RE DNA methylation sites can be found in the vicinity of peaks, rather than directly inside of them.

```
## Run the step7TopGenesUserPeakOverlap function.
## Since we performed analyses only using TFs in the step 3 function, the
## top genes are all TFs, so a message that separate output for
## TFs will be skipped is displayed.
exampleObject <- step7TopGenesUserPeakOverlap(
    TENETMultiAssayExperiment = exampleObject,
    peakData = examplePeakFile,
    hypermethGplusAnalysis = FALSE,
    hypomethGplusAnalysis = TRUE,
    topGeneNumber = 10,
    distanceFromREDNAMethylationSites = 100
)
#> All genes with hypomethylated G+ links are TFs, so the separate output for TFs will be
↪  skipped.

## Results are included in the metadata of the returned MultiAssayExperiment
## object under the step7TopGenesSurvival list.
## For each analysis type, data frames with peak overlap information are
## included in sub-lists, with each data frame saved under the names
## topGenes and topTFs, unless the top genes are all TFs, in which case
## the separate top TFs output is skipped.

## Display the data frame of results for RE DNA methylation sites linked to the
## top TFs. A variety of data are included for each RE DNA methylation site,
## including its location, the coordinates of its search window, the top genes
## it is linked to, and whether it was found within the specified distance to
## any peak in each of the peak files.
head(
    metadata(
        exampleObject
    )$step7TopGenesUserPeakOverlap$hypomethGplusResults$topGenes
)
#>            DNAMethylationSiteID chromosome     start       end searchStart
#> cg00002190           cg00002190       chr8  19697522  19697523    19697422
#> cg00002809           cg00002809      chr17  78486271  78486272    78486171
#> cg00047815           cg00047815      chr22  44915621  44915622    44915521
#> cg00051307           cg00051307       chr5  73228366  73228367    73228266
#> cg00069003           cg00069003       chr5 140784521 140784522   140784421
#> cg00085256           cg00085256       chr2 217386723 217386724   217386623
#>            searchEnd FOXA1_ENSG00000129514_linked SPDEF_ENSG00000124664_linked
#> cg00002190  19697623                         TRUE                         TRUE
#> cg00002809  78486372                         TRUE                         TRUE
#> cg00047815  44915722                         TRUE                         TRUE
#> cg00051307  73228467                         TRUE                         TRUE
#> cg00069003 140784622                         TRUE                         TRUE
#> cg00085256 217386824                         TRUE                         TRUE
#>            GATA3_ENSG00000107485_linked ESR1_ENSG00000091831_linked
#> cg00002190                         TRUE                        TRUE
#> cg00002809                        FALSE                        TRUE
#> cg00047815                         TRUE                        TRUE
#> cg00051307                        FALSE                       FALSE
#> cg00069003                        FALSE                       FALSE
#> cg00085256                        FALSE                       FALSE
#>            MYB_ENSG00000118513_linked XBP1_ENSG00000100219_linked
#> cg00002190                       TRUE                       FALSE
#> cg00002809                       TRUE                       FALSE
#> cg00047815                       TRUE                       FALSE
#> cg00051307                      FALSE                       FALSE
#> cg00069003                      FALSE                        TRUE
#> cg00085256                      FALSE                       FALSE
#>            POU4F1_ENSG00000152192_linked OVOL3_ENSG00000105261_linked
#> cg00002190                         FALSE                        FALSE
#> cg00002809                         FALSE                        FALSE
#> cg00047815                         FALSE                        FALSE
#> cg00051307                         FALSE                        FALSE
#> cg00069003                         FALSE                        FALSE
#> cg00085256                         FALSE                        FALSE
#>            ZNF552_ENSG00000178935_linked CENPA_ENSG00000115163_linked peakFile
#> cg00002190                         FALSE                        FALSE    FALSE
#> cg00002809                         FALSE                        FALSE    FALSE
#> cg00047815                         FALSE                        FALSE    FALSE
#> cg00051307                         FALSE                        FALSE    FALSE
#> cg00069003                         FALSE                        FALSE    FALSE
#> cg00085256                         FALSE                        FALSE    FALSE
```

Datasets included in the TENET package

The following objects are contained in the TENET package. Since LazyData is not enabled, objects will need to be accessed using the `data()` function, as demonstrated below.

`humanTranscriptionFactorList`: Human transcription factor list

A character vector of gene Ensembl IDs which were identified as human TFs by Lambert SA et al (PMID: 29425488). Candidate proteins were manually examined by a panel of experts based on available data. Proteins with experimentally demonstrated DNA binding specificity were considered TFs. Other proteins, such as co-factors and RNA binding proteins, were classified as non-TFs. **Citation:** Lambert SA, Jolma A, Campitelli LF, et al. The Human Transcription Factors. Cell. 2018 Feb 8;172(4):650-665. doi: 10.1016/j.cell.2018.01.029. Erratum in: Cell. 2018 Oct 4;175(2):598-599. PMID: 29425488.

```r
## Load the humanTranscriptionFactorList dataset
data("humanTranscriptionFactorList", package = "TENET")
## Display the names of the first few TFs on the list
head(humanTranscriptionFactorList)
#> [1] "DUX1_HUMAN"      "DUX3_HUMAN"      "ENSG00000001167" "ENSG00000004848"
#> [5] "ENSG00000005073" "ENSG00000005102"
```

Session info

```r
sessionInfo()
#> R Under development (unstable) (2024-10-21 r87258)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.1 LTS
#> 
#> Matrix products: default
#> BLAS:   /home/biocbuild/bbs-3.21-bioc/R/lib/libRblas.so 
#> LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.12.0
#> 
#> locale:
#>  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C              
#>  [3] LC_TIME=en_GB              LC_COLLATE=C              
#>  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8   
#>  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C                 
#>  [9] LC_ADDRESS=C               LC_TELEPHONE=C            
#> [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C       
#> 
#> time zone: America/New_York
#> tzcode source: system (glibc)
#> 
#> attached base packages:
#> [1] stats4    stats     graphics  grDevices utils     datasets  methods  
#> [8] base     
#> 
#> other attached packages:
#>  [1] MultiAssayExperiment_1.33.1 SummarizedExperiment_1.37.0
#>  [3] Biobase_2.67.0              GenomicRanges_1.59.1       
#>  [5] GenomeInfoDb_1.43.1         IRanges_2.41.1             
#>  [7] S4Vectors_0.45.2            BiocGenerics_0.53.3        
#>  [9] generics_0.1.3              MatrixGenerics_1.19.0      
#> [11] matrixStats_1.4.1           TENET.AnnotationHub_0.99.4
#> [13] TENET.ExperimentHub_0.99.0  TENET_0.99.0
#>
#> loaded via a namespace (and not attached):
#>   [1] bitops_1.0-9             DBI_1.2.3                pastecs_1.4.2
#>   [4] rlang_1.1.4              magrittr_2.0.3           MotifDb_1.49.0
#>   [7] compiler_4.5.0           RSQLite_2.3.8            png_0.1-8
#>  [10] vctrs_0.6.5              reshape2_1.4.4           stringr_1.5.1
#>  [13] pkgconfig_2.0.3          crayon_1.5.3             fastmap_1.2.0
#>  [16] dbplyr_2.5.0             XVector_0.47.0           labeling_0.4.3
#>  [19] splitstackshape_1.4.8    utf8_1.2.4               Rsamtools_2.23.0
#>  [22] rmarkdown_2.29           tzdb_0.4.0               UCSC.utils_1.3.0
#>  [25] preprocessCore_1.69.0    tinytex_0.54             purrr_1.0.2
#>  [28] bit_4.5.0                xfun_0.49                zlibbioc_1.53.0
#>  [31] cachem_1.1.0             jsonlite_1.8.9           blob_1.2.4
#>  [34] DelayedArray_0.33.2      BiocParallel_1.41.0      parallel_4.5.0
#>  [37] R6_2.5.1                 stringi_1.8.4            RColorBrewer_1.1-3
#>  [40] sesame_1.25.0            rtracklayer_1.67.0       boot_1.3-31
#>  [43] Rcpp_1.0.13-1            knitr_1.49               wheatmap_0.2.0
#>  [46] R.utils_2.12.3           matlab_1.0.4.1           readr_2.1.5
#>  [49] BiocBaseUtils_1.9.0      splines_4.5.0            Matrix_1.7-1
#>  [52] tidyselect_1.2.1         abind_1.4-8              yaml_2.3.10
#>  [55] codetools_0.2-20         curl_6.0.1               lattice_0.22-6
#>  [58] tibble_3.2.1             plyr_1.8.9               withr_3.0.2
#>  [61] KEGGREST_1.47.0          evaluate_1.0.1           survival_3.7-0
#>  [64] BiocFileCache_2.15.0     ExperimentHub_2.15.0     Biostrings_2.75.1
#>  [67] pillar_1.9.0             BiocManager_1.30.25      filelock_1.0.3
#>  [70] RCircos_1.2.2            RCurl_1.98-1.16          BiocVersion_3.21.1
#>  [73] hms_1.1.3                ggplot2_3.5.1            munsell_0.5.1
#>  [76] scales_1.3.0             glue_1.8.0               tools_4.5.0
#>  [79] BiocIO_1.17.1            AnnotationHub_3.15.0     data.table_1.16.2
#>  [82] GenomicAlignments_1.43.0 XML_3.99-0.17            grid_4.5.0
#>  [85] sesameData_1.25.0        AnnotationDbi_1.69.0     colorspace_2.1-1
#>  [88] GenomeInfoDbData_1.2.13  restfulr_0.0.15          cli_3.6.3
#>  [91] rappdirs_0.3.3           fansi_1.0.6              S4Arrays_1.7.1
#>  [94] dplyr_1.1.4              gtable_0.3.6             R.methodsS3_1.8.2
#>  [97] digest_0.6.37            SparseArray_1.7.2        rjson_0.2.23
#> [100] farver_2.1.2             R.oo_1.27.0              memoise_2.0.1
#> [103] htmltools_0.5.8.1        lifecycle_1.0.4          httr_1.4.7
#> [106] mime_0.12                bit64_4.5.2
```