Package 'CytoMethIC'

May 5, 2026

Type Package

Title DNA methylation-based machine learning models

Description This package provides model data and functions for easily using machine learning models that use data from the DNA methylome to classify cancer type and phenotype from a sample. The primary motivation for the development of this package is to abstract away the granular and accessibility-limiting code required to utilize machine learning models in R. Our package provides this abstraction for RandomForest, e1071 Support Vector, Extreme Gradient Boosting, and Tensorflow models. This is paired with an ExperimentHub component, which contains models developed for epigenetic cancer classification and predicting phenotypes. This includes CNS tumor classification, Pan-cancer classification, race prediction, cell of origin classification, and subtype classification models. The package links to our models on ExperimentHub. The package currently supports HM450, EPIC, EPICv2, MSA, and MM285.

Version 1.9.0

License Artistic-2.0

Depends R (>= 4.4.0), ExperimentHub

Imports utils, stats, tools, sesame, methods, sesameData, BiocParallel, BiocManager

VignetteBuilder knitr

Suggests tibble, BiocStyle, randomForest, testthat, knitr, rmarkdown, e1071, xgboost, keras, tensorflow

URL https://github.com/zhou-lab/CytoMethIC

BugReports https://github.com/zhou-lab/CytoMethIC/issues

biocViews ExperimentData, MicroarrayData, Genome, ExperimentHub, MethylationArrayData, CancerData, PackageTypeData

NeedsCompilation no

RoxygenNote 7.3.2

Encoding UTF-8

git_url https://git.bioconductor.org/packages/CytoMethIC

git_branch devel

git_last_commit 193109c

git_last_commit_date 2026-04-28

Repository Bioconductor 3.24

Date/Publication 2026-05-05

Author Wanding Zhou [aut] (ORCID: <https://orcid.org/0000-0001-9126-1932>),
Jacob Fanale [aut, cre] (ORCID:
<https://orcid.org/0009-0002-0490-3269>)

Maintainer Jacob Fanale <jfanale@seas.upenn.edu>

Contents

`.optimizeFrac` *Internal function for fraction optimization*

Description

Internal function for fraction optimization

Usage

```
.optimizeFrac(
  frac,
  ref,
  q,
  errFunc,
  temp = 0.5,
  maxIter = 1000,
  delta = 1e-04,
  step.max = 1,
  verbose = FALSE
)
```

Arguments

`frac`	initial fraction
`ref`	reference
`q`	query
`errFunc`	error function
`temp`	annealing temperature
`maxIter`	maximum iteration to stop after converge
`delta`	delta score to reset counter
`step.max`	maximum step, do not adjust
`verbose`	output debug info

Value

a list of fractions and min err

`cmi_checkVersion` *Check CytoMethIC versions*

Description

print package verison of cytomethic and depended packages to help troubleshoot installation issues.

Usage

```
cmi_checkVersion()
```

Value

print the versions of cytomethic and dependencies

Examples

```
cmi_checkVersion()
```

`cmi_deconvolution` *Reference-based cell type deconvolution*

Description

This is a reference-based cell composition estimation. The function takes a reference methylation status matrix (rows for probes and columns for cell types) and a query beta value measurement.

Usage

```
cmi_deconvolution(ref, q, trim = FALSE, ...)
```

Arguments

`ref`	reference methylation
`q`	target measurement: length(q) == nrow(ref)
`trim`	to trim query input beta values. this relieves unclean background subtraction
`...`	extra parameters for optimization.

Details

The length of the target beta values should be the same as the number of rows of the reference Matrix. The function outputs a list containing the estimated cell fraction, the error of optimization.

Value

a list of fraction, min error.

Examples

```
ref = cbind(
  CD4 = c(1,1,1,0,1,0),
  CD19 = c(0,0,1,1,0,1),
  CD14 = c(1,1,1,1,0,1))
rownames(ref) = paste0("cg",1:6)
trueFrac = runif(3)
trueFrac = trueFrac / sum(trueFrac)
q = ref %*% trueFrac
trueFrac
cmi_deconvolution(ref, q)
```

`cmi_deconvolution2` *Reference-based cell type deconvolution (allowing one unknown component)*

Description

This is a reference-based cell composition estimation. The function takes a reference methylation status matrix (rows for probes and columns for cell types) and a query beta value measurement.

Usage

```
cmi_deconvolution2(ref, q, trim = FALSE, ...)
```

Arguments

`ref`	reference methylation
`q`	target measurement: length(q) == nrow(ref)
`trim`	to trim query input beta values. this relieves unclean background subtraction
`...`	extra parameters to .optimizeFrac

Details

The length of the target beta values should be the same as the number of rows of the reference Matrix. The method assumes one unknown component. It outputs a list containing the estimated cell fraction, the error of optimization and methylation status of the unknown component.

Value

a list of fraction, min error and unknown component methylation state

Examples

```
ref = cbind(
  CD4 = c(1,1,1,0,1,0),
  CD19 = c(0,0,1,1,0,1),
  CD14 = c(1,1,1,1,0,1))
rownames(ref) = paste0("cg",1:6)
trueFrac = runif(4)
trueFrac = trueFrac / sum(trueFrac)
```

```
ref_unk = sample(c(0,1), nrow(ref), replace=TRUE)
q = cbind(ref_unk, ref) %*% trueFrac
trueFrac
res = cmi_deconvolution2(ref, q)
res$frac
```

`cmi_models` *Master data frame for all model objects*

Description

This is an internal object which will be updated on every new release

Format

tibble

Value

master sheet of CytoMethIC model objects

Examples

```
print(cmi_models[,c("EHID","Title")])
```

`cmi_predict` *The cmi_predict function takes in a model and a sample, and uses the model to predict it. This function supports randomForest, e1071::svm, xgboost, and keras/tensorflow models. For xgboost and keras models, the features used in classification as well as a label mapping must be provided for output.*

Description

The cmi_predict function takes in a model and a sample, and uses the model to predict it. This function supports randomForest, e1071::svm, xgboost, and keras/tensorflow models. For xgboost and keras models, the features used in classification as well as a label mapping must be provided for output.

Usage

```
cmi_predict(betas, cmi_model, verbose = FALSE, BPPARAM = SerialParam())
```

Arguments

`betas`	DNA methylation beta
`cmi_model`	Cytomethic model downloaded from ExperimentHub
`verbose`	be verbose with warning
`BPPARAM`	use MulticoreParam(n) for parallel processing

Value

predicted cancer type label

Examples

```
library(sesame)
library(ExperimentHub)
library(CytoMethIC)

## Cancer Type
model = ExperimentHub()[["EH8395"]]
betas = openSesame(sesameDataGet("EPICv2.8.SigDF")[[1]])
betas = imputeBetas(mLiftOver(betas, "HM450"))
cmi_predict(betas, model)

betas = openSesame(sesameDataGet('EPIC.1.SigDF'), mask=FALSE)
cmi_predict(betas, model)

betas = sesameDataGet("HM450.1.TCGA.PAAD")$betas
betas = imputeBetas(betas)
cmi_predict(betas, model)
```

Index