Package 'iDOS'

October 13, 2022

Type Package

Title Integrated Discovery of Oncogenic Signatures

Version 1.0.0

Date 2016-07-27

Author Syed Haider, Francesca M Buffa

Maintainer Syed Haider <Syed.Haider@oncology.ox.ac.uk>

Depends R (>= 2.15.0), VennDiagram (>= 1.6.5)

Description Integrate molecular profiles to discover candidate oncogenic drivers.

License GPL-2

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2016-07-27 20:25:35

R topics documented:

iDOS-package *Integrated Discovery of Oncogenic Signatures*

Description

A method to identify correlated changes on mRNA and DNA level. For details, see PMID: 27358048

Details

Package:	iDOS
Type:	Package
Version:	0.0.1
Date:	2014-11-14
License:	GPL-2

Author(s)

Syed Haider, Francesca M Buffa
Maintainer: Syed Haider <Syed.Haider@oncology.ox.ac.uk>
MRC Weatherall Institute of Molecular Medicine
Department of Oncology
University of Oxford.

Examples

```
# load test data
x <- get.test.data(data.types = c("mRNA.T", "mRNA.N", "CNA", "ann"));

# list of features to be assessed for differential expression
feature.ids <- rownames(x$mRNA.T$BLCA);

# get differentially expressed features
DE.results <- find.DE.features(
  exp.data.T = x$mRNA.T,
  exp.data.N = x$mRNA.N,
  feature.ids = feature.ids,
  test.name = "t.test"
  );

# get top features
top.features <- get.top.features(
  DE.features = cbind("FC" = DE.results[, 1], "P" = DE.results[, 2]),
  cna.data.fractions = x$CNA.fractions$BLCA,
  mRNA.FC.up = 0.25,
```

```
  mRNA.FC.down = 0.25,
  mRNA.p = 0.05,
  mRNA.top.n = NULL,
  cna.fractions.gain = 0.2,
  cna.fractions.loss = 0.2
  );

# temporary output directory
tmp.output.dir <- tempdir();

# estimate mRNA and CNA correlation using the pre-selected top features
correlated.features <- estimate.expression.cna.correlation(
  exp.data = x$mRNA.T$BLCA,
  cna.data.log2 = x$CNA.log2$BLCA,
  corr.threshold = 0.3,
  corr.direction = "two.sided",
  subtypes.metadata = list(
    "subtype.samples.list" = list("All" = colnames(x$mRNA.T$BLCA))
    ),
  feature.ids = top.features,
  cancer.type = "BLCA",
  data.dir = paste(tmp.output.dir, "/data/BLCA/", sep = ""),
  graphs.dir = paste(tmp.output.dir, "/graphs/BLCA/", sep = "")
  );
```

`create.counts.table` *create.counts.table*

Description

Summary function to collapse the counts of selected (e.g. correlated) features per cancer type into counts table

Usage

```
create.counts.table(corr.summary = NULL)
```

Arguments

`corr.summary`	A list object containing subtype specific selected (e.g. correlated) features. This is the list object returned by `estimate.expression.cna.correlation`

Value

A matrix of cancer type specific counts

Author(s)

Syed Haider

See Also

estimate.expression.cna.correlation

Examples

```
# load test data
x <- get.test.data(data.types = c("mRNA.T", "CNA"));

# temporary output directory
tmp.output.dir <- tempdir();

# go through each cancer type iteratively and perform mRNA-CNA correlation analysis
correlated.features <- list();
for (cancer.type in names(x$mRNA.T)) {

  # estimate mRNA and CNA correlation for each cancer/disease type
  correlated.features[[cancer.type]] <- estimate.expression.cna.correlation(
    exp.data = x$mRNA.T[[cancer.type]],
    cna.data.log2 = x$CNA.log2[[cancer.type]],
    corr.threshold = 0.3,
    corr.direction = "two.sided",
    subtypes.metadata = list(
      "subtype.samples.list" = list("All" = colnames(x$mRNA.T[[cancer.type]]))
  ),
    feature.ids = rownames(x$mRNA.T[[cancer.type]]),
    cancer.type = cancer.type,
    data.dir = paste(tmp.output.dir, "/data/", cancer.type, sep = ""),
    graphs.dir = paste(tmp.output.dir, "/graphs/", cancer.type, sep = "")
    );
  }

# create counts table across cancer types
counts.table <- create.counts.table(corr.summary = correlated.features);
```

`create.training.validation.split`

create.training.validation.split

Description

Utility function to create random partitions of a dataset into training and validation sets. If samples are < 200, 66:34; otherwise 50:50 partitions are generated between training and validation sets respectively

Usage

```
create.training.validation.split(
  exp.data = NULL, ann.data = NULL, seed.number = 51214
  )
```

Arguments

exp.data	Feature by sample mRNA abundance matrix
ann.data	Sample by clinical attribute matrix
seed.number	Random seed for sampling

Value

A list of four matrices expression and two associated clinical matrices (exp.T, ann.T, exp.V and ann.V). One set for training and one for validation

Author(s)

Syed Haider

Examples

```
# load test data
x <- get.test.data(data.types = c("mRNA.T", "ann"));

# create training and validation sets
partitioned.datasets <- create.training.validation.split(
  exp.data = x$mRNA.T$BLCA,
  ann.data = x$ann$BLCA,
  seed.number = 51214
  );
```

---

estimate.expression.cna.correlation

estimate.expression.cna.correlation

Description

Estimate subtype specific correlation between mRNA and CNA profiles

Usage

```
estimate.expression.cna.correlation(
  exp.data = NULL,
  cna.data.log2 = NULL,
  corr.threshold = 0.3,
  corr.direction = "two.sided",
  subtypes.metadata = NULL,
  feature.ids = NULL,
  cancer.type = NULL,
  data.dir = NULL,
  graphs.dir = NULL
  )
```

Arguments

exp.data	Feature by sample mRNA abundance matrix
cna.data.log2	Feature by sample CNA log ratio matrix
corr.threshold	Threshold for Spearman's Rho to consider a feature as candidate driver
corr.direction	Whether to include positively (greater), negatively (less) or both (two.sided) correlated features. Defaults to two.sided
subtypes.metadata	Subtypes metadata list of lists. Must contain at least one subtype specific samples using list subtype.samples.list. If no subtypes are present, specify list element "All" with all samples
feature.ids	Vector of features to be used to estimate correlation
cancer.type	Name of the cancer type or dataset
data.dir	Path to output directory where mRNA and CNA correlation statistics will be stored
graphs.dir	Path to graphs directory

Value

A list of lists containing correlated features per cancer subtype

Author(s)

Syed Haider

Examples

```
# load test data
x <- get.test.data(data.types = c("mRNA.T", "CNA"));

# temporary output directory
tmp.output.dir <- tempdir();
```

```
# estimate mRNA and CNA correlation
correlated.features <- estimate.expression.cna.correlation(
  exp.data = x$mRNA.T$BLCA,
  cna.data.log2 = x$CNA.log2$BLCA,
  corr.threshold = 0.3,
  corr.direction = "two.sided",
  subtypes.metadata = list(
    "subtype.samples.list" = list("All" = colnames(x$mRNA.T$BLCA))
    ),
  feature.ids = rownames(x$mRNA.T$BLCA),
  cancer.type = "BLCA",
  data.dir = paste(tmp.output.dir, "/data/BLCA/", sep = ""),
  graphs.dir = paste(tmp.output.dir, "/graphs/BLCA/", sep = "")
  );
```

`estimate.null.distribution.correlation`
estimate.null.distribution.correlation

Description

Function to estimate probability of observing correlations as high as observed using a feature list of interest

Usage

```
estimate.null.distribution.correlation(
  exp.data = NULL,
  cna.data.log2 = NULL,
  corr.threshold = 0.3,
  corr.direction = "two.sided",
  subtypes.metadata = NULL,
  feature.ids = NULL,
  observed.correlated.features = NULL,
  iterations = 50,
  cancer.type = NULL,
  data.dir = NULL
  )
```

Arguments

`exp.data`	Feature by sample mRNA abundance matrix
`cna.data.log2`	Feature by sample CNA log ratio matrix
`corr.threshold`	Threshold for Spearman's Rho to consider a feature as candidate driver
`corr.direction`	Whether to include positively (greater), negatively (less) or both (two.sided) correlated features. Defaults to `two.sided`

`subtypes.metadata`
: Subtypes metadata list. Contains at least subtype specific samples

`feature.ids`
: Vector of features to be used to estimate correlation

`observed.correlated.features`
: List of features that were found to be correlated for subtypes of a given cancer type

`iterations`
: Number of random permutations for estimating p value

`cancer.type`
: Name of the cancer type or dataset

`data.dir`
: Path to output directory where the randomisation results will be stored

Value

1 if successful

Author(s)

Syed Haider

See Also

`estimate.expression.cna.correlation`

Examples

```
# load test data
x <- get.test.data(data.types = c("mRNA.T", "CNA"));

# temporary output directory
tmp.output.dir <- tempdir();

# estimate mRNA and CNA correlation for each cancer/disease type
correlated.features <- estimate.expression.cna.correlation(
  exp.data = x$mRNA.T$BLCA,
  cna.data.log2 = x$CNA.log2$BLCA,
  corr.threshold = 0.3,
  corr.direction = "two.sided",
  subtypes.metadata = list(
    "subtype.samples.list" = list("All" = colnames(x$mRNA.T$BLCA))
    ),
  feature.ids = rownames(x$mRNA.T$BLCA),
  cancer.type = "BLCA",
  data.dir = paste(tmp.output.dir, "/data/BLCA/", sep = ""),
  graphs.dir = paste(tmp.output.dir, "/graphs/BLCA/", sep = "")
  );

# estimate NULL distribution
estimate.null.distribution.correlation(
  exp.data = x$mRNA.T$BLCA,
  cna.data.log2 = x$CNA.log2$BLCA,
  corr.threshold = 0.3,
```

```
  corr.direction = "two.sided",
  subtypes.metadata = list(
    "subtype.samples.list" = list("All" = colnames(x$mRNA.T$BLCA))
    ),
  feature.ids = rownames(x$mRNA.T$BLCA),
  observed.correlated.features = correlated.features$correlated.genes.subtypes,
  iterations = 50,
  cancer.type = "BLCA",
  data.dir = paste(tmp.output.dir, "/data/BLCA/", sep = "")
  );
```

`find.DE.features` *find.DE.features*

Description

Funtion to identify differentially expressed/variable features between Tumour (T) and Normal (N) profiles

Usage

```
find.DE.features(
  exp.data.T = NULL,
  exp.data.N = NULL,
  feature.ids = NULL,
  test.name = "t.test"
  )
```

Arguments

`exp.data.T`	Feature by sample mRNA abundance matrix; tumour samples
`exp.data.N`	Feature by sample mRNA abundance matrix; normal/baseline samples
`feature.ids`	Vector of features to be used to estimate correlation
`test.name`	Specify the statistical test name (exactly as it appears in R). Supported tests are `t.test`, `wilcox.test`, `var.test`

Value

Feature by cancer type matrix of log2 fold change (T vs N) and adjusted P values. P values are estimated through `test.name`

Author(s)

Syed Haider

See Also

t.test, wilcox.test, var.test

Examples

```
# load test data
x <- get.test.data(data.types = c("mRNA.T", "mRNA.N"));

# list of features to be assessed for differential expression
feature.ids <- rownames(x$mRNA.T$BLCA);

DE.results <- find.DE.features(
  exp.data.T = x$mRNA.T,
  exp.data.N = x$mRNA.N,
  feature.ids = feature.ids,
  test.name = "t.test"
  );
```

---

get.program.defaults *get.program.defaults*

Description

Get default datasets bundled with package for test runs

Usage

```
get.program.defaults()
```

Value

A list with `program.data.dir` containing path to example program directory and `test.data.dir` containing path to example datasets directory

Author(s)

Syed Haider

Examples

```
x <- get.program.defaults();
```

get.test.data *get.test.data*

Description

Function to load test data

Usage

```
get.test.data(data.types = c("mRNA.T", "ann"))
```

Arguments

data.types	Datatypes to be read Valid datatypes are: mRNA.T, mRNA.N, CNA (includes: log2, calls and fractions), annotations

Value

List of lists containing datasets and respective molecular profiles as matrices

Author(s)

Syed Haider

Examples

```
x <- get.test.data(data.types = c("mRNA.T", "mRNA.N", "ann"));
```

get.top.features *get.top.features*

Description

Prioritise top features satisfying the criteria specified by various parameters described below

Usage

```
get.top.features(
  DE.features = NULL,
  cna.data.fractions = NULL,
  mRNA.FC.up = 0,
  mRNA.FC.down = 0,
  mRNA.p = 0.05,
  mRNA.top.n = NULL,
```

```
  cna.fractions.gain = 0.2,
  cna.fractions.loss = 0.2
  )
```

Arguments

DE.features	Matrix containing differentially expressed features with two columns: FC and P. P may contain adjusted P or raw
cna.data.fractions	Feature by cancer type matrix with CNA fractions
mRNA.FC.up	Log2 fold change threshold for selecting over-expressed features
mRNA.FC.down	Log2 fold change threshold for selecting under-expressed features
mRNA.p	P value threshold for selecting significantly differentially expressed features. Mutually exclusive to mRNA.top.n
mRNA.top.n	Top n differentially expressed features satisfying each of the fold change criteria. Mutually exclusive to mRNA.p
cna.fractions.gain	Threshold for selecting copy number gain/amplifications
cna.fractions.loss	Threshold for selecting copy number losses

Value

Vector of top features

Author(s)

Syed Haider

Examples

```
# load test data
x <- get.test.data(data.types = c("mRNA.T", "mRNA.N", "CNA"));

# list of features to be assessed for differential expression
feature.ids <- rownames(x$mRNA.T$BLCA);

# get differentially expressed features
DE.results <- find.DE.features(
  exp.data.T = x$mRNA.T,
  exp.data.N = x$mRNA.N,
  feature.ids = feature.ids,
  test.name = "t.test"
  );

# get top features
top.features <- get.top.features(
  DE.features = cbind("FC" = DE.results[, 1], "P" = DE.results[, 2]),
```

```
  cna.data.fractions = x$CNA.fractions$BLCA,
  mRNA.FC.up = 0.25,
  mRNA.FC.down = 0.25,
  mRNA.p = 0.05,
  mRNA.top.n = NULL,
  cna.fractions.gain = 0.2,
  cna.fractions.loss = 0.2
  );
```

`load.datasets` *load.datasets*

Description

Function to load and systemise molecular datasets

Usage

```
load.datasets(
  data.dir = "./",
  metadata = NULL,
  data.types = c("mRNA.T", "ann")
  )
```

Arguments

`data.dir`	Path to base data directory or directory containing molecular profiles
`metadata`	Dataset by profile metadata matrix containing file names of the molecular profiles for different datasets
`data.types`	Datatypes to be read Valid datatypes are: mRNA.T, mRNA.N, CNA (includes: log2, calls and fractions), annotations

Value

List of lists containing datasets and respective molecular profiles as matrices

Author(s)

Syed Haider

Examples

```
# locate test data directory which comes with the package
data.dir <- paste(system.file("programdata/testdata/", package = "iDOS"), "/", sep = "");

# read meta data file
```

```
metadata <- read.table(
  file = paste(data.dir, "metadata.txt", sep = ""),
  row.names = 1,
  header = TRUE,
  sep = "\t",
  stringsAsFactors = FALSE
  );

x <- load.datasets(
  data.dir = data.dir,
  metadata = metadata,
  data.types = c("mRNA.T", "mRNA.N", "ann")
  );
```

Index