Package 'aracne.networks'

November 12, 2026

Type Package

Title ARACNe-inferred gene networks from TCGA tumor datasets

Version 1.39.1

Description This package provides access to ARACNe-inferred gene regulatory networks (regulons) from TCGA tumor datasets. The networks are hosted on Zenodo, downloaded on demand and cached locally with BiocFileCache. The package also contains a function to export them into plain-text format.

License file LICENSE

URL https://github.com/federicogiorgi/aracne.networks

BugReports https://github.com/federicogiorgi/aracne.networks/issues

biocViews ExperimentData, Genome, Homo_sapiens_Data, CancerData

NeedsCompilation no

Depends R (>= 4.0), viper

Imports BiocFileCache, tools, utils

Suggests BiocStyle, knitr, rmarkdown, testthat

VignetteBuilder knitr

Encoding UTF-8

git_url https://git.bioconductor.org/packages/aracne.networks

git_branch devel

git_last_commit d000659

git_last_commit_date 2026-09-23

Repository Bioconductor 3.24

Date/Publication 2026-11-12

Author Federico M. Giorgi [aut, cre] (ORCID: <https://orcid.org/0000-0002-7325-9908>), Mariano J. Alvarez [aut]

Maintainer Federico M. Giorgi <federico.giorgi@gmail.com>

Contents

aracne.networks-package 2
getRegulon 3
regulons 4
write.regulon 5

Index **7**

`aracne.networks-package`

ARACNe-inferred gene networks from TCGA tumor datasets

Description

This package provides access to ARACNe-inferred networks from TCGA tumor datasets and a function to export them into text form. The networks are hosted on Zenodo (https://doi.org/10.5281/zenodo.22918956), downloaded on demand with `getRegulon` and cached locally. The available networks are listed by `listRegulons` and described in regulons.

Details

Package:	aracne.networks
Type:	Package
License:	file LICENSE

Author(s)

Federico M. Giorgi, Mariano J. Alvarez

Maintainer: Federico M. Giorgi <federico.giorgi@gmail.com>

References

Giorgi,F.M. et al. (2016) ARACNe-AP: Gene Network Reverse Engineering through Adaptive Partitioning inference of Mutual Information. Bioinformatics 2016 Alvarez, M.J. et al. (2016) Functional characterization of somatic mutations in cancer using network-based inference of protein activity. Nature Genetics 2016

Examples

```
### Create a random regulon with two hubs
# The first hub will have 100 targets
# The second hub will have 67 targets
regulon<-list()
regulon[["hub1"]]<-list(
    tfmode=setNames(runif(100,-1,1),paste0("target",sample(1:1000,100))),
    likelihood=runif(100,0,1)
)
regulon[["hub2"]]<-list(
```

```
    tfmode=setNames(runif(67,-1,1),paste0("target",sample(1:1000,67))),
    likelihood=runif(67,0,1)
)
class(regulon)<-"regulon"
write.regulon(regulon,file=tempfile())


### List the available networks
listRegulons()

### Print the Bladder Carcinoma (blca) network to standard output
# The gene ids are in Entrez format
regulonblca<-getRegulon("blca")
write.regulon(regulonblca,file="",n=10)
```

`getRegulon` *Retrieve ARACNe networks from the online repository*

Description

The ARACNe networks are stored on Zenodo (https://doi.org/10.5281/zenodo.22918956). `listRegulons` returns the table of available networks. `getRegulon` downloads one network, stores it in a local `BiocFileCache` cache and returns it as a `regulon` object. Each network is downloaded only once; subsequent calls read it from the cache and do not require an internet connection.

Usage

```
listRegulons()

getRegulon(network, force = FALSE, verbose = TRUE)
```

Arguments

`network`	Character string. The network to retrieve, either the short name in the `network` column of `listRegulons()` (e.g. `"blca"`) or the name of the object that was distributed with previous versions of the package (e.g. `"regulonblca"`).
`force`	Logical. If TRUE, the network is downloaded again even if it is already present in the cache.
`verbose`	Logical. If TRUE, a message is printed when a network is downloaded.

Details

Downloaded files are checked against their MD5 checksums before being loaded. The cache is located in `tools::R_user_dir("aracne.networks", "cache")`; a different location can be set with `options(aracne.networks.cache = "path")`. The cache can be inspected and cleaned with the functions of the `BiocFileCache` package, e.g. `BiocFileCache::bfcinfo(BiocFileCache::BiocFileCache(tools::R_user_dir("aracne.networks", "cache")))`.

Value

`listRegulons` returns a data.frame with one row per network and the columns `network` (short name), `object` (name of the object in previous versions of the package), `tumor`, `tcga` (TCGA project code), `regulators`, `interactions` and `bytes` (size of the download).

`getRegulon` returns an object of class regulon (regulon-class); see regulons.

See Also

regulons, write.regulon

Examples

```
listRegulons()

### Retrieve the Bladder Carcinoma network (downloaded once, then cached)
regulonblca <- getRegulon("blca")
length(regulonblca)
write.regulon(regulonblca, n = 10)
```

---

regulons *Context-specific ARACNe interactomes of human tumors*

Description

The package provides 25 human context-specific transcriptional regulatory networks (interactomes) reverse engineered by the ARACNe-AP algorithm from tumor RNA-Seq expression profiles, mostly from TCGA. Each interactome is a list object of S3 class 'regulon' where each element represents a transcriptional regulator and contains two vectors: (1) a named numeric vector indicating the mode of regulation (MoR) for each target gene, whose ID is indicated by the names attribute of the vector; (2) a numeric vector indicating the confidence score (likelihood) for the regulator-target interaction. Genes are identified by Entrez Gene IDs.

Up to version 1.39.0 the networks were bundled in the package as data sets (e.g. `data(regulonblca)`). They are now hosted on Zenodo and retrieved with getRegulon, e.g. `regulonblca <- getRegulon("blca")`.

Format

Objects of class regulon (regulon-class).

Details

network	tumor	regulators	interactions
"blca"	Bladder Carcinoma	6,054	489,101
"brca"	Breast Carcinoma	6,054	331,919
"cesc"	Cervical Squamous Carcinoma	6,056	583,961
"coad"	Colon Adenocarcinoma	6,056	413,789
"esca"	Esophageal Carcinoma	5,951	529,286
"gbm"	Glioblastoma	6,056	563,850
"hnsc"	Head and Neck Squamous Carcinoma	6,055	423,104
"kirc"	Kidney Renal Clear Cell Carcinoma	6,054	350,478
"kirp"	Kidney Papillary Carcinoma	6,055	452,653
"laml"	Acute Myeloid Leukemia	6,007	531,535
"lihc"	Liver Hepatocellular Carcinoma	6,056	469,922
"luad"	Lung Adenocarcinoma	6,055	399,513
"lusc"	Lung Squamous Carcinoma	6,054	455,032
"net"	Neuroendocrine Tumor	6,129	666,241
"ov"	Ovarian Carcinoma	6,007	647,358
"paad"	Pancreas Carcinoma	6,056	520,756

"pcpg"	Pheochromocytoma and Paraganglioma	6,056	603,617
"prad"	Prostate Carcinoma	6,053	330,922
"read"	Rectal Adenocarcinoma	6,056	557,911
"sarc"	Sarcoma	6,112	526,591
"stad"	Stomach Adenocarcinoma	6,056	561,858
"tgct"	Testicular Cancer	6,056	432,621
"thca"	Thyroid Carcinoma	6,053	317,582
"thym"	Thymoma	6,056	387,923
"ucec"	Uterine Corpus Endometrial Carcinoma	6,055	469,845

Source

https://doi.org/10.5281/zenodo.22918956. The network files are released under the CC BY-NC-ND 4.0 license (https://creativecommons.org/licenses/by-nc-nd/4.0/); commercial use is not permitted.

References

Giorgi, F.M. et al. (2016) ARACNe-AP: Gene Network Reverse Engineering through Adaptive Partitioning inference of Mutual Information. Bioinformatics 32(14):2233-2235. doi: 10.1093/bioinformatics/btw216.

Alvarez, M.J. et al. (2016) Functional characterization of somatic mutations in cancer using network-based inference of protein activity. Nature Genetics 48(8):838-847. doi: 10.1038/ng.3593.

See Also

getRegulon, listRegulons, write.regulon

Examples

```
regulonblca <- getRegulon("blca")
write.regulon(regulonblca, regulator = "399")
```

---

write.regulon *Print a regulon object into a text file*

Description

This function will print the network into an output stream. Four columns will be printed: the Regulator id, the Target id, the Mode of Action (MoA, based on Spearman correlation that indicates the sign of the connection and ranges between -1 and +1), the Likelihood (essentially an edge weight that indicates how strong the mutual information for an edge is when compared to the maximum observed MI in the network, it ranges between 0 and 1).

Usage

```
write.regulon(
    regulon,
    file="",
    sep="\t",
    header=TRUE,
    n=Inf,
    regulator=NULL
)
```

Arguments

regulon	An object of class regulon
file	File name where the network will be printed
sep	String, a separator for the fields (default = "\t")
header	Logical. If a header should be printed. Default is TRUE
n	Numeric. How many interactions to print. Default is Inf
regulator	String. A particular regulator. Default is NULL

Value

Text output containing the network in tabular format.

Examples

```
### Create a random regulon with two hubs
# The first hub will have 100 targets
# The second hub will have 67 targets
regulon<-list()
regulon[["hub1"]]<-list(
    tfmode=setNames(runif(100,-1,1),paste0("target",sample(1:1000,100))),
    likelihood=runif(100,0,1)
)
regulon[["hub2"]]<-list(
    tfmode=setNames(runif(67,-1,1),paste0("target",sample(1:1000,67))),
    likelihood=runif(67,0,1)
)
class(regulon)<-"regulon"
write.regulon(regulon,file=tempfile())


### Print the Bladder Carcinoma (blca) network to standard output
# The gene ids are in Entrez format
regulonblca<-getRegulon("blca")
write.regulon(regulonblca,file="",n=10)
```

Index

∗ **datasets**
regulons, 4

aracne.networks
*(*aracne.networks-package*)*, 2
aracne.networks-package, 2

getRegulon, *2*, 3, *4, 5*

listRegulons, *2*, *5*
listRegulons *(*getRegulon*)*, 3

regulon-class, *3, 4*
regulonblca *(*regulons*)*, 4
regulonbrca *(*regulons*)*, 4
reguloncesc *(*regulons*)*, 4
reguloncoad *(*regulons*)*, 4
regulonesca *(*regulons*)*, 4
regulongbm *(*regulons*)*, 4
regulonhnsc *(*regulons*)*, 4
regulonkirc *(*regulons*)*, 4
regulonkirp *(*regulons*)*, 4
regulonlaml *(*regulons*)*, 4
regulonlihc *(*regulons*)*, 4
regulonluad *(*regulons*)*, 4
regulonlusc *(*regulons*)*, 4
regulonnet *(*regulons*)*, 4
regulonov *(*regulons*)*, 4
regulonpaad *(*regulons*)*, 4
regulonpcpg *(*regulons*)*, 4
regulonprad *(*regulons*)*, 4
regulonread *(*regulons*)*, 4
regulons, *2–4*, 4
regulonsarc *(*regulons*)*, 4
regulonstad *(*regulons*)*, 4
regulontgct *(*regulons*)*, 4
regulonthca *(*regulons*)*, 4
regulonthym *(*regulons*)*, 4
regulonucec *(*regulons*)*, 4

write.regulon, *4, 5*, 5