Package 'HPAanalyze'

April 6, 2026

Type Package

Title Retrieve and analyze data from the Human Protein Atlas

Date 2025-05-29

Version 1.28.0

Description Provide functions for retrieving, exploratory analyzing and visualizing the Human Protein Atlas data. HPAanalyze is designed to fullfill 3 main tasks: (1) Import, subsetting and export downloadable datasets; (2) Visualization of downloadable datasets for exploratory analysis; and (3) Working with the individual XML files. This package aims to serve researchers with little programming experience, but also allow power users to use the imported data as desired.

Depends R (>= 3.5.0)

License GPL-3 + file LICENSE

RoxygenNote 7.3.2

Encoding UTF-8

Imports dplyr, openxlsx, ggplot2, tibble, xml2, stats, utils, gridExtra

Suggests knitr, rmarkdown, devtools, BiocStyle

VignetteBuilder knitr

biocViews Proteomics, CellBiology, Visualization, Software

URL https://github.com/anhtr/HPAanalyze

BugReports https://github.com/anhtr/HPAanalyze/issues

LazyData true

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

git_branch RELEASE_3_22

git_last_commit 6d7d09e

git_last_commit_date 2025-10-29

Repository Bioconductor 3.22

Date/Publication 2026-04-05

Author Anh Nhat Tran [aut, cre]

Maintainer Anh Nhat Tran <trannhatanh89@gmail.com>

Contents

`hpaDownload` *Download datasets*

Description

Download the latest version of HPA datasets and import them in R. It is recommended to only download the datasets you need, as some of them may be very big.

Usage

```
hpaDownload(downloadList = "histology", version = "latest")
```

Arguments

`downloadList`	A vector or string indicate which datasets to download. Common values: • `'normal_tissue'` • `'pathology'` • `'subcellular_location'` For the full list of possible values for a specific version, set downloadList as NULL: `hpaDownload(downloadList = NULL, version = <version>)` You can also use the following shortcuts: • `'all'`: download everything (not recommended!!!) • `'histology'`: same as `c('normal_tissue', 'pathology', 'subcellular_location')` See https://www.proteinatlas.org/about/download for more information.
`version`	A string indicate which version to be downloaded. Possible value: • `'latest'`: Download latest version. Due to the constantly changing nature of the API, this will download the latest version known to work with this package. Require Internet connection. This is the default option. • `'example'` or `'built-in'`: Load the built-in histology dataset from 'HPA-analyze' ('hpa_histology_data'). Do not require internet connection. • `'v23'`: version 23 • `'v24'`: version 24

Value

This function will return a list of tibbles corresponding to requested datasets.

See Also

hpaDownload hpa_histology_data

Other downloadable datasets functions: hpaExport(), hpaSubset()

Examples

```
histologyData <- hpaDownload(downloadList='histology', version='example')
# tissueTranscriptData <- hpaDownload('RNA transcript tissue')
```

---

hpaExport *Export the subset data*

Description

Export the list object generated by hpaSubset() into xlsx format. Due to the size of some HPA datasets, as well as the limitation of the output format, exporting the full datasets generated by hpaDownload() is not recommended. This is a convenient wrapper for 'write.' functions.

Usage

```
hpaExport(data, fileName, fileType = "xlsx")
```

Arguments

data	Input the list object generated by hpaSubset()
fileName	A string indicate the desired output file name. Do not include file extension such as '.xlsx'.
fileType	The format as which the data will be exported. Choose one of these options: 'xlsx', 'csv' and 'tsv'.

Value

- 'xlsx': return one .xlsx file named 'fileName.xlsx'. One individual sheet for each dataset in the input list object.
- 'csv': return .csv files, one for each dataset in the input list object, named 'fileName_datasetName.csv'
- 'tsv': return .tsv files, one for each dataset in the input list object, named 'fileName_datasetName.tsv'

See Also

Other downloadable datasets functions: hpaDownload(), hpaSubset()

Examples

```
downloadedData <- hpaDownload(downloadList='histology', version='example')
geneList <- c('TP53', 'EGFR')
tissueList <- c('breast', 'cerebellum', 'skin 1')
cancerList <- c('breast cancer', 'glioma', 'melanoma')

subsetData <- hpaSubset(data=downloadedData,
                        targetGene=geneList,
                        targetTissue=tissueList,
                        targetCancer=cancerList)
hpaExport(data=subsetData,
          fileName='TP53_EGFR_in_tissue_cancer.xlsx',
          fileType='xlsx')
```

`hpaSubset` *Subset downloaded data*

Description

`hpaSubset()` subsets data by gene name, tissue, cell type, cancer and/or cell line. The input is the list object generated by `hpaDownload()` or as the output of another `hpaSubset()`. Use `hpaListParam()` to see the list of available parameters for a specific list object. This is a convenient wrapper for 'lapply/filter' and works on any table which contain 'gene', 'tissue', 'cell_type', 'cancer', and 'cell_line' columns.

`hpaListParam()` list available variables in downloaded data that can be used as parameters to subset the data via `hpaSubset()`. This function work with the data object generated by `hpaDownload()` or a previous call of `hpaSubset()`. This is a convenient wrapper for 'lapply/unique' and works on any table which contain 'tissue', 'cell_type', 'cancer', and 'cell_line' columns.

Usage

```
hpaSubset(
  data = NULL,
  targetGene = NULL,
  targetTissue = NULL,
  targetCellType = NULL,
  targetCancer = NULL,
  targetCellLine = NULL
)

hpaListParam(data = NULL)
```

Arguments

`data`	Input the list object generated by `hpaDownload()` or `hpaSubset()`
`targetGene`	Vector of strings of HGNC gene symbols. It will be used to subset every dataset in the list object. You can also mix HGNC gene symbols and ensemnbl ids (start with ENSG) and they will be converted to HGNC gene symbols.
`targetTissue`	Vector of strings of normal tissues. Will be used to subset the `normal_tissue` and `rna_tissue` dataset.

targetCellType	Vector of strings of normal cell types. Will be used to subset the normal_tissue dataset.
targetCancer	Vector of strings of cancer types. Will be used to subset the pathology dataset.
targetCellLine	Vector of strings of cell lines. Will be used to subset the rna_cell_line dataset.

Value

hpaSubset will return a list of tibbles as the result of subsetting, depending on the input data.

The output of hpaListParam() is a list of vectors containing all subset parameter for the downloaded data.

See Also

Other downloadable datasets functions: hpaDownload(), hpaExport()

Examples

```
downloadedData <- hpaDownload(downloadList='histology', version='example')
geneList <- c('TP53', 'EGFR')
tissueList <- c('breast', 'cerebellum', 'skin 1')
cancerList <- c('breast cancer', 'glioma', 'melanoma')

subsetData <- hpaSubset(data=downloadedData,
                        targetGene=geneList,
                        targetTissue=tissueList,
                        targetCancer=cancerList)

downloadedData <- hpaDownload(downloadList='histology', version='example')
params <- hpaListParam(data=downloadedData)
params$normal_tissue
```

---

hpaVis *Visualize data in one function*

Description

This function is an universal visualization function that allow calling other hpaVis functions via a single function call. By default, this function will use the dataset bundled with HPAanalyze, and provide a grid of all available plots. The types of plots in the output can be specified via the visType argument. If only one plot type is specified, this function will return the exact same output as the specific hpaVis function used to create the plot.

Usage

```
hpaVis(
  data = NULL,
  targetGene = NULL,
  targetTissue = NULL,
  targetCellType = NULL,
  targetCancer = NULL,
  visType = c("Tissue", "Patho", "Subcell"),
```

```
  color = c("#FCFDBF", "#FE9F6D", "#DE4968", "#8C2981"),
  customTheme = FALSE,
  ...
)
```

Arguments

data	Input the list object generated by hpa_download() or hpa_subset(). By default this function use the example dataset bundled with HPAanalyze.
targetGene	Vector of strings of HGNC gene symbols. By default it is set to c('TP53', 'EGFR', 'CD44', 'PTEN', 'IDH1'). You can also mix HGNC gene symbols and ensemnl ids (start with ENSG) and they will be converted to HGNC gene symbols.
targetTissue	Vector of strings of normal tissue names. By default it is set to the first available.
targetCellType	Vector of strings of normal cell types. By default inludes all available cell types in the target tissues.
targetCancer	Vector of strings of normal tissues. By default it is set to the first available.
visType	Vector of strings indicating which plots will be generated. Currently available values are "all", "Tissue", "Patho", "Cancer", "Subcell".
color	Vector of 4 colors used to depict different expression levels.
customTheme	Logical argument. If TRUE, the function will return a barebone ggplot2 plot to be customized further.
...	Additional arguments to be passed downstream to other hpaVis functions being called behind the scene. These arguments includes targetTissue, targetCellType, targetCancer. See documentation for individual hpaVis functions for more information.

Value

If multiple visType is chosen, this function will return multiple graphs in one panel. If only one visType is chosen, this function will return a ggplot2 plot object, which can be further modified if desirable. See help file for each of the hpaVis function for more information about individual graphs.

See Also

hpaDownload, hpaSubset

Other visualization functions: hpaVisPatho(), hpaVisSubcell(), hpaVisTissue()

Examples

```
hpaVis()
```

hpaVisPatho *Visualize pathology data*

Description

Visualize the expression of genes of interest in each cancer.

Usage

```
hpaVisPatho(
  data = NULL,
  targetGene = NULL,
  targetCancer = NULL,
  facetBy = "cancer",
  color = c("#FCFDBF", "#FE9F6D", "#DE4968", "#8C2981"),
  customTheme = FALSE
)
```

Arguments

data	Input the list object generated by hpa_download() or hpa_subset(). Require the pathology dataset. Use HPA histology data (built-in) by default.
targetGene	Vector of strings of HGNC gene symbols. By default it is set to c('TP53', 'EGFR', 'CD44', 'PTEN'). You can also mix HGNC gene symbols and ensemnl ids (start with ENSG) and they will be converted to HGNC gene symbols.
targetCancer	Vector of strings of normal tissues. The function will plot all available cancer by default.
facetBy	Determine how multiple graphs would be faceted. Either cancer (default) or gene.
color	Vector of 4 colors used to depict different expression levels.
customTheme	Logical argument. If TRUE, the function will return a barebone ggplot2 plot to be customized further.

Value

This function will return a ggplot2 plot object, which can be further modified if desirable. The pathology data is visualized as multiple bar graphs, one for each type of cancer. For each bar graph, x axis contains the inquired protein and y axis contains the proportion of patients.

See Also

Other visualization functions: hpaVis(), hpaVisSubcell(), hpaVisTissue()

Examples

```
data("hpa_histology_data")
geneList <- c('TP53', 'EGFR', 'CD44', 'PTEN', 'IDH1', 'IDH2', 'CYCS')
cancerList <- c('breast cancer', 'glioma', 'melanoma')

## A typical function call
```

```
hpaVisPatho(data=hpa_histology_data,
                targetGene=geneList)
```

`hpaVisSubcell` *Visualize subcellular location data*

Description

Visualize the the confirmed subcellular locations of genes of interest.

Usage

```
hpaVisSubcell(
  data = NULL,
  targetGene = NULL,
  reliability = c("enhanced", "supported", "approved", "uncertain"),
  color = c("#FCFDBF", "#8C2981"),
  customTheme = FALSE
)
```

Arguments

`data`	Input the list object generated by `hpa_download()` or `hpa_subset()`. Require the `subcellular_location` dataset. Use HPA histology data (built-in) by default.
`targetGene`	Vector of strings of HGNC gene symbols. By default it is set to `c('TP53', 'EGFR', 'CD44', 'PTEN')`. You can also mix HGNC gene symbols and ensemnl ids (start with ENSG) and they will be converted to HGNC gene symbols.
`reliability`	Vector of string indicate which reliability scores you want to plot. The default is everything `c("enhanced", "supported", "approved","uncertain")`.
`color`	Vector of 2 colors used to depict if the protein expresses in a location or not.
`customTheme`	Logical argument. If `TRUE`, the function will return a barebone ggplot2 plot to be customized further.

Value

This function will return a ggplot2 plot object, which can be further modified if desirable. The subcellular location data is visualized as a tile graph, in which the x axis includes the inquired proteins and the y axis contain the subcellular locations.

See Also

Other visualization functions: `hpaVis()`, `hpaVisPatho()`, `hpaVisTissue()`

Examples

```
data("hpa_histology_data")
geneList <- c('TP53', 'EGFR', 'CD44', 'PTEN', 'IDH1', 'IDH2', 'CYCS')

## A typical function call
hpaVisSubcell(data=hpa_histology_data,
                targetGene=geneList)
```

`hpaVisTissue` *Visualize tissue data*

Description

Visualize the expression of protein of interest in each target tissue by cell types.

Usage

```
hpaVisTissue(
  data = NULL,
  targetGene = NULL,
  targetTissue = NULL,
  targetCellType = NULL,
  color = c("#FCFDBF", "#FE9F6D", "#DE4968", "#8C2981"),
  customTheme = FALSE
)
```

Arguments

`data`	Input the list object generated by `hpa_download()` or `hpa_subset()`. Require the `normal_tissue` dataset. Use HPA histology data (built-in) by default.
`targetGene`	Vector of strings of HGNC gene symbols. By default it is set to `c('TP53', 'EGFR', 'CD44', 'PTEN')`. You can also mix HGNC gene symbols and ensemnl ids (start with ENSG) and they will be converted to HGNC gene symbols.
`targetTissue`	Vector of strings of normal tissues. Default to all.
`targetCellType`	Vector of strings of normal cell types. Default to all.
`color`	Vector of 4 colors used to depict different expression levels.
`customTheme`	Logical argument. If `TRUE`, the function will return a barebone ggplot2 plot to be customized further.

Value

This function will return a ggplot2 plot object, which can be further modified if desirable. The tissue data is visualized as a heatmap: x axis contains inquired protein and y axis contains tissue/cells of interest.

See Also

Other visualization functions: `hpaVis()`, `hpaVisPatho()`, `hpaVisSubcell()`

Examples

```
data("hpa_histology_data")
geneList <- c('TP53', 'EGFR', 'CD44', 'PTEN', 'IDH1', 'IDH2', 'CYCS')
tissueList <- c('breast', 'cerebellum', 'skin 1')

## A typical function call
hpaVisTissue(data=hpa_histology_data,
             targetGene=geneList,
             targetTissue=tissueList)
```

`hpaXml` *Extract details about an individual protein from XML file in one function*

Description

This function is the umbrella function for the hpaXml function family. It take the input of either one Ensembl gene id or a imported XML object resulting from a `hpaXmlGet()` function call. By default, it will extract all information available for HPAanalyze user from the XML file by calling every hpaXml function and put all results into a list.

Usage

```
hpaXml(
  inputXml,
  extractType = c("ProtClass", "TissueExprSum", "Antibody", "TissueExpr"),
  ...
)
```

Arguments

`inputXml`	Input can be either one Ensembl gene id (start with ENSG) or a imported XML object resulting from a `hpaXmlGet()` function call. You can also use HGNC gene symbol and it will be converted to ensembl id.
`extractType`	A vector of strings indicate which information is desired for extraction. By default this function will call all `hpaXml` functions available. Other options are `'ProtClass'`, `'TissueExprSum'`, `'Antibody'`, `'TissueExpr'`.
`...`	Additional arguments to be passed downstream to other hpaXml functions being called behind the scene. See help files of other hpaXml functions for more information.

Value

This function returns a list. Each element of the list is information extracted from the XML file specified using other hpaXml functions. See help file for each XML function for more information.

See Also

Other xml functions: `hpaXmlAntibody()`, `hpaXmlGet()`, `hpaXmlProtClass()`, `hpaXmlTissueExpr()`, `hpaXmlTissueExprSum()`

Examples

```
hpaXml(inputXml='ENSG00000131979', extractType=c('ProtClass', 'TissueExprSum', 'Antibody'))
```

---

hpaXmlAntibody *Extract antibody information*

Description

Extract information about the antibodies used for a specific protein. It is important to note that the data that HPA provides on their website and through xml files are not one-to-one equivalents.

Usage

```
hpaXmlAntibody(importedXml)
```

Arguments

importedXml	Input an xml document object resulted from a hpaXmlGet() call.

Value

This function returns a tibble of 4 columns, containing information about the antibodies used in the project for the inquired protein: id, releaseDate, releaseVersion, and RRID.

See Also

Other xml functions: hpaXml(), hpaXmlGet(), hpaXmlProtClass(), hpaXmlTissueExpr(), hpaXmlTissueExprSum()

Examples

```
GCH1xml <- hpaXmlGet('ENSG00000131979')
hpaXmlAntibody(GCH1xml)
```

---

hpaXmlGet *Download and import xml file*

Description

Download and import individual xml file for a specified protein. This function calls xml2::read_xml() under the hood. It is important to note that the data that HPA provides on their website and through xml files are not one-to-one equivalents.

Usage

```
hpaXmlGet(targetEnsemblId, version = "latest")
```

Arguments

targetEnsemblId	A string of one ensembl ID, start with ENSG. For example 'ENSG00000131979'. You can also use HGNC gene symbol and it will be converted to ensembl id.
version	A string indicate which version to be downloaded. Possible value: • 'latest': Download latest version. • 'v?' with '?' is a integer: Download a specific version of the dataset. For example: 'v18' download version 18. Currently support version 13 and above.

Value

This function return an object of class "xml_document" "xml_node" containing the content of the imported XML file. (See documentations for package xml2 for more information.)

See Also

Other xml functions: hpaXml(), hpaXmlAntibody(), hpaXmlProtClass(), hpaXmlTissueExpr(), hpaXmlTissueExprSum()

Examples

```
GCH1xml <- hpaXmlGet('ENSG00000131979')
```

---

hpaXmlProtClass *Extract protein classes*

Description

Extract protein class information from imported xml document resulted from hpaXmlGet(). It is important to note that the data that HPA provides on their website and through xml files are not one-to-one equivalents.

Usage

```
hpaXmlProtClass(importedXml)
```

Arguments

importedXml	Input an xml document object resulted from a hpaXmlGet() call.

Value

This function return a tibble of 4 columns.

See Also

Other xml functions: hpaXml(), hpaXmlAntibody(), hpaXmlGet(), hpaXmlTissueExpr(), hpaXmlTissueExprSum()

Examples

```
GCH1xml <- hpaXmlGet('ENSG00000131979')
hpaXmlProtClass(GCH1xml)
```

`hpaXmlTissueExpr` *Extract tissue expression details*

Description

Extract tissue expression information for each sample and url to download images from imported xml document resulted from `hpaXmlGet()`. It is important to note that the data that HPA provides on their website and through xml files are not one-to-one equivalents. For example, xml files usually only provide one of the two histology image for each patient.

Usage

```
hpaXmlTissueExpr(importedXml)
```

Arguments

`importedXml`	Input an xml document object resulted from a `hpaXmlGet()` call.

Value

This function returns a list of tibbles, each for an antibody. Each tibble contains information about all individual samples and their staining. Due to the variation in amount of information available for these samples, the number of columns differs, but the tibble essentially includes: patientId, age, sex, staining, intensity, quantity, location, imageUrl, snomedCode, and tissueDescription. The last two items may have more than one column each.

See Also

Other xml functions: `hpaXml()`, `hpaXmlAntibody()`, `hpaXmlGet()`, `hpaXmlProtClass()`, `hpaXmlTissueExprSum()`

Examples

```
GCH1xml <- hpaXmlGet('ENSG00000131979')
hpaXmlTissueExpr(GCH1xml)
```

hpaXmlTissueExprSum *Extract tissue expression and download images*

Description

Extract tissue expression information and url to download images from imported xml document resulted from `hpaXmlGet()`. It is important to note that the data that HPA provides on their website and through xml files are not one-to-one equivalents.

Usage

```
hpaXmlTissueExprSum(importedXml, downloadImg = FALSE)
```

Arguments

`importedXml`	Input an xml document object resulted from a `hpaXmlGet()` call.
`downloadImg`	Logical argument. The function will download all image from the extracted urls into the working folder.

Value

This function return a list consists of a summary string, which is a very brief description of the protein, and a tibble of 2 columns: tissue (name of tissue available) and imageUrl (link to download the perspective image)

See Also

Other xml functions: `hpaXml()`, `hpaXmlAntibody()`, `hpaXmlGet()`, `hpaXmlProtClass()`, `hpaXmlTissueExpr()`

Examples

```
GCH1xml <- hpaXmlGet('ENSG00000131979')
hpaXmlTissueExprSum(GCH1xml)
```

hpa_histology_data *HPA histology dataset*

Description

Dataset downloaded with `hpaDownload('histology', version = 'latest')`. This should be the most updated dataset at the time of generation. Check metadata for more information.

Usage

```
hpa_histology_data
```

Format

A list of 3 tibbles

normal_tissue Normal tissue IHC data

pathology Cancer IHC data

subcellular_location Subcellular location IF data

See Also

hpaDownload

Examples

```
# load data
data("hpa_histology_data")

# access data frames
normal_tissue_data <- hpa_histology_data$normal_tissue
cancer_data <- hpa_histology_data$pathology
subcell_location_data <- hpa_histology_data$subcellular_location

# see metadata
hpa_histology_data$metadata
```

Index