Package 'imageFeatureTCGA'

May 18, 2026

Title Import features from hovernet, provgigapath into a MultiAssayExperiment

Version 1.0.0

Description The package imports data from HoverNet, and ProvGigaPath pipelines. Pipeline output data are hosted in a self-owned online repository. Package functionality conveniently incorporates pipeline data into existing MultiAssayExperiment instances from curatedTCGAData.

Depends R (>= 4.5.0)

Imports BiocBaseUtils, BiocFileCache, BiocIO, BumpyMatrix, dplyr, httr2, IRanges, methods, readr, rjsoncons, S4Vectors, SingleCellExperiment, SpatialExperiment, SummarizedExperiment, TCGAutils, TENxIO, tibble, utils

Suggests AnnotationDbi, anndataR, BiocStyle, BiocParallel, cowplot, curatedTCGAData, curl, ggplot2, imageTCGAutils, knitr, png, RColorBrewer, rhdf5, rmarkdown, SpatialFeatureExperiment, tinytest, yaml

VignetteBuilder knitr

biocViews Software, Infrastructure, DataImport, DataRepresentation

BugReports https://github.com/waldronlab/imageFeatureTCGA/issues

URL https://github.com/waldronlab/imageFeatureTCGA

License Artistic-2.0

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Date 2026-03-19

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

git_branch RELEASE_3_23

git_last_commit 39ba8ff

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-05-17

Author Marcel Ramos [aut] (ORCID: <https://orcid.org/0000-0002-3242-0582>, affiliation: CUNY Graduate School of Public Health and Health Policy, New York, NY USA),
Ilaria Billato [aut, cre] (ORCID: <https://orcid.org/0000-0002-3335-3254>, affiliation: Department of Biology, University of Padova),
Eslam Abousamra [aut] (affiliation: CUNY Graduate School of Public Health and Health Policy, New York, NY USA),
Sehyun Oh [aut] (ORCID: <https://orcid.org/0000-0002-9490-3061>, affiliation: CUNY Graduate School of Public Health and Health Policy, New York, NY USA)

Maintainer Ilaria Billato <ilaria.billato@phd.unipd.it>

Contents

`embeddingStack` *Create an embedding stack from a ProvGigaList object*

Description

This function imports and combines embeddings from multiple ProvGigaPath files contained within a ProvGigaList object. It supports both slide-level and tile-level embeddings.

Usage

```
embeddingStack(
  con,
  levels,
  layer = "last_layer_embed",
  redownload = FALSE,
  ...
)
```

Arguments

con	ProvGigaList object containing multiple ProvGigaPath files or URLs.
levels	character() vector specifying the data level for each file in the ProvGigaList. If not provided, levels will be inferred from the ProvGiga objects within the list.
layer	character() vector specifying the embedding layer to import from each file. Default is "last_layer_embed" for "slide_level" and ignored for "tile_level".
redownload	logical(1) indicating whether to re-download cached files.
...	Additional arguments passed to the slide and tile import functions.

Value

A named list of tibbles, where each list element corresponds to a specific level (e.g., "slide_level", "tile_level") and contains the combined embeddings from all files at that level.

Examples

```
slide_urls <- getCatalog("provgigapath") |>
    dplyr::filter(level == "slide_level", Project.ID == "TCGA-UVM") |>
    dplyr::slice(1:3) |>
    getFileURLs()

ProvGigaList(slide_urls) |>
   embeddingStack(redownload = FALSE)
```

---

getCatalog *Download the catalog of available HoVerNet and ProvGigaPath files*

Description

The getCatalog function retrieves a catalog of all available HoVerNet and ProvGigaPath files, including filenames, sizes, pipelines used, tumor types, and data levels.

Usage

```
getCatalog(
  pipeline = c("hovernet", "provgigapath"),
  format = c("csv", "thumb", "h5ad", "geojson", "json"),
  redownload = FALSE
)

getFileURLs(catalog)
```

Arguments

pipeline	character() One or both "hovernet" and/or "provgigapath" specifying which pipeline(s) to include in the catalog. Default includes both.
format	character() One or more of "csv", "thumb", "h5ad", "geojson", or "json" specifying which file formats to include in the catalog. Default includes all.
redownload	logical(1L) Whether to redownload the catalog file even if it is already cached locally. Default is FALSE.
catalog	A tibble as returned by getCatalog().

Value

getCatalog: A tibble containing the full catalog of available files for the specified pipeline(s).

getFileURLs: A character() vector of full URLs for the files listed in the provided catalog.

Examples

```
## Get the full catalog of available files
getCatalog(pipeline = c("hovernet", "provgigapath"), format = "h5ad")


## Get file URLs from the catalog
getCatalog(pipeline = "hovernet", format = "h5ad") |>
    dplyr::slice(1:10) |>
    getFileURLs()
```

HoverNet *Import Hovernet JSON, H5AD, and PNG files into Bioconductor classes*

Description

The HoverNet virtual class and its subclasses represent different file formats used in the HoverNet cell segmentation and classification pipeline for histopathology images. The HoverNet constructor function creates instances of the appropriate subclass based on the file format of the provided resource. The import methods for each subclass read the respective file formats and represent the data as either a SpatialExperiment or SpatialFeatureExperiment object, depending on the specified output class.

The HoverNetJSON constructor function creates an instance of the HoverNetJSON class. The resource argument can be either a file path or URL to a Hovernet JSON file. The contours parameter is optional and can be used to include cell contours in the metadata. The outClass parameter specifies the output class when importing the data, either SpatialExperiment or SpatialFeatureExperiment.

Usage

```
HoverNet(
  resource,
  contours = FALSE,
  outClass = c("SpatialExperiment", "SpatialFeatureExperiment")
)

## S4 method for signature 'HoverNetJSON'
show(object)

## S4 method for signature 'HoverNetJSON,ANY,ANY'
import(con, format, text, ...)

## S4 method for signature 'HoverNetH5AD,ANY,ANY'
import(con, format, text, ...)

## S4 method for signature 'HoverNetPNG,ANY,ANY'
import(con, format, text, ...)
```

Arguments

resource	character(1) the file path or URL to the Hovernet JSON file.
contours	logical(1) whether to include cell contours in the metadata of the resulting SpatialExperiment or SpatialFeatureExperiment object. Default is FALSE and used only when importing "JSON" data.
outClass	character(1) specifying the output class when importing either "JSON" or "H5AD" data into a Bioconductor class object. Must be one of "SpatialExperiment" (default) or "SpatialFeatureExperiment".
object	An object of class HoverNetJSON.
con	The connection from which data is loaded or to which data is saved. If this is a character vector, it is assumed to be a file name and a corresponding file connection is created and then closed after exporting the object. If it is a BiocFile derivative, the data is loaded from or saved to the underlying resource. If missing, the function will return the output as a character vector, rather than writing to a connection.
format	The format of the output. If missing and con is a file name, the format is derived from the file extension. This argument is unnecessary when con is a derivative of BiocFile.
text	If con is missing, this can be a character vector directly providing the string data to import.
...	Parameters to pass to the format-specific method.

Details

The HoverNetJSON class represents Hovernet JSON files used for cell segmentation and classification in histopathology images. It extends the TENxFile class from the TENxIO package, allowing

for efficient handling of large JSON files. The class includes a slot to indicate whether cell contours should be included in the metadata when importing the data. As well as a slot to specify the output class when importing the data, either SpatialExperiment or SpatialFeatureExperiment. The HoverNetH5AD class represents Hovernet H5AD files, which contain similar data but in a different format. The HoverNetPNG class represents PNG thumbnail images of the whole-slide images used in HoverNet.

The HoverNetJSON constructor function can import file paths and URLs. Remote files are automatically cached using BiocFileCache when the import method is called. This allows for efficient handling of large JSON files without the need to download them manually.

Value

- HoverNetJSON: An object of class HoverNetJSON
- import,HoverNetJSON-method: An object of class SpatialExperiment or SpatialFeatureExperiment containing the cell data and spatial coordinates extracted from the Hovernet JSON file
- import,HoverNetH5AD-method: An object of class SpatialExperiment or SpatialFeatureExperiment containing the cell data and spatial coordinates extracted from the Hovernet H5AD file
- import,HoverNetPNG-method: A PNG image as an RGB array as given by png::readPNG.

Slots

`contours` `logical(1)` indicating whether to include cell contours in the metadata of the resulting SpatialExperiment or SpatialFeatureExperiment object.

`outClass` `character(1)` specifying the output class when importing the data. One of "SpatialExperiment" or "SpatialFeatureExperiment".

`is_url` `logical(1)` indicating whether the resource is a URL.

show

The show method for HoverNetJSON objects displays the resource, contours, and outClass slots and vaules.

import

The import method for HoverNetJSON reads the JSON file and represents the data as either a SpatialExperiment or SpatialFeatureExperiment object. It extracts cell centroid coordinates, cell types, and type probabilities, and optionally includes cell contours in the metadata. The resulting SpatialExperiment object contains the cell data in the colData slot and spatial coordinates in the spatialCoords slot of the object.

The import method for HoverNetH5AD reads the H5AD file and represents the data as either a SpatialExperiment or SpatialFeatureExperiment object. It extracts cell centroid coordinates, cell types, mean intensity, and nearest neighbor distance. The resulting SpatialExperiment object contains the cell data in the colData slot and spatial coordinates in the spatialCoords slot of the object.

Author(s)

Ilaria B., Marcel R.

Sehyun O.

Examples

```
## Manual download and local file input
hov_json_file <- paste0(
    "https://store.cancerdatasci.org/hovernet/json/",
    "TCGA-VG-A8LO-01A-01-DX1.B39A4D64-82A1-4A04-8AB6-918F3058B83B.json.gz"
)
dest_json <- file.path(tempdir(), basename(hov_json_file))
download.file(hov_json_file, destfile = dest_json)

HoverNet(dest_json, outClass = "SpatialExperiment") |>
    import()

## Direct URL input (with caching)
HoverNet(hov_json_file, outClass = "SpatialExperiment") |>
    import()

## Import as SpatialFeatureExperiment
library(SpatialFeatureExperiment)
HoverNet(dest_json, outClass = "SpatialFeatureExperiment") |>
    import()


hov_h5ad_file <- paste0(
    "https://store.cancerdatasci.org/hovernet/h5ad/",
    "TCGA-VG-A8LO-01A-01-DX1.B39A4D64-82A1-4A04-8AB6-918F3058B83B.h5ad.gz"
)
dest_h5ad <- file.path(tempdir(), basename(hov_h5ad_file))
download.file(hov_h5ad_file, destfile = dest_h5ad)

HoverNet(dest_h5ad, outClass = "SpatialExperiment") |>
    import()

## Import HoverNetPNG thumbnail from URL
hov_png_url <- paste0(
   "https://store.cancerdatasci.org/hovernet/thumb/",
   "TCGA-VG-A8LO-01A-02-DX2.9B58474C-DAC0-4D45-B13C-0A1EA9E1BC32.png"
)
HoverNet(hov_png_url) |>
  import()
```

`linkTCGA` *Link MultiAssayExperiment object to TCGA data*

Description

Link MultiAssayExperiment object to TCGA data

Usage

```
linkTCGA(MultiAssayExperiment, catalog, redownload = FALSE, parallel = TRUE)
```

Arguments

MultiAssayExperiment	A MultiAssayExperiment object containing sample metadata with TCGA barcodes.
catalog	A data.frame from getCatalog() containing file metadata.
redownload	logical(1) whether to re-download cached files. Default is FALSE.
parallel	logical(1) whether to use parallel processing to instantiate multiple ProvGiga objects. This can speed up the import process when importing multiple files, but requires the BiocParallel package to be installed. Default is TRUE.

Value

A MultiAssayExperiment object with two additional assays:

slide_assay A SummarizedExperiment containing slide-level ProvGigaPath embeddings.

tile_assay A SummarizedExperiment containing tile-level ProvGigaPath embeddings stored as a BumpyMatrix.

Examples

```
library(curatedTCGAData)
coad <- curatedTCGAData(
    diseaseCode = "COAD",
    assays = "RNASeq2GeneNorm",
    version = "2.1.1",
    dry.run = FALSE
)
coad_sub <- coad[, 1:3, ]
catalog <- getCatalog(pipeline = "provgigapath", format = "csv")
linkTCGA(coad_sub, catalog, parallel = FALSE)
```

ProvGiga *Import ProvGiga slide-level data into a Bioconductor class object*

Description

The ProvGiga class represents ProvGiga slide-level CSV files containing embeddings for histopathology images. It extends the TENxFile class from the TENxIO package, allowing for efficient handling of large CSV files. The class includes slots to specify the output class when importing the data, either SpatialExperiment or SpatialFeatureExperiment, the tumor type, and whether the resource is a URL.

The ProvGiga constructor function creates an instance of the ProvGiga class. The resource argument can be either a file path or URL to a ProvGiga CSV file. The tumorType parameter specifies the tumor type associated with the ProvGiga data.

Usage

```
ProvGiga(
  resource,
  level = c("slide_level", "tile_level"),
  is_url = TRUE,
  tumorType = NA_character_
)

## S4 method for signature 'ProvGiga'
show(object)

## S4 method for signature 'ProvGigaCSV,ANY,ANY'
import(con, format, text, ...)
```

Arguments

resource	character(1) the file path or URL to the ProvGiga CSV file, or a TENxFile object.
level	character(1) specifying the level of ProvGiga data to import. Must be one of "slide_level" or "tile_level". If not provided, the level is inferred from the file path or URL.
is_url	logical(1) indicating whether the resource is a URL. If not provided, it is inferred from the resource value.
tumorType	character(1) specifying the tumor type associated with the ProvGiga data. Required if resource is a local file (file path).
object	An object of class ProvGiga.
con	The connection from which data is loaded or to which data is saved. If this is a character vector, it is assumed to be a file name and a corresponding file connection is created and then closed after exporting the object. If it is a BiocFile derivative, the data is loaded from or saved to the underlying resource.

If missing, the function will return the output as a character vector, rather than writing to a connection.

`format` The format of the output. If missing and `con` is a file name, the format is derived from the file extension. This argument is unnecessary when `con` is a derivative of BiocFile.

`text` If `con` is missing, this can be a character vector directly providing the string data to import.

`...` Parameters to pass to the format-specific method.

Details

The `ProvGiga` constructor function can import file paths, URLs, and `TENxFile` objects. If a local file path is provided, the `tumorType` parameter must be specified to indicate the tumor type associated with the ProvGiga data. If a URL is provided, the tumor type is inferred from the URL structure.

Value

- `ProvGiga`: An object of class `ProvGiga`.
- `import`: A `tibble` containing slide-level embeddings along with slide names and tumor type.

Slots

`tumorType character(1)` specifying the tumor type associated with the `ProvGiga` data.

`level character(1)` specifying the level of ProvGiga data to import. Must be one of `"slide_level"` or `"tile_level"`. If not provided, the level is inferred from the file path or URL.

`is_url logical(1)` indicating whether the resource is a URL.

show

The `show` method for `ProvGiga` objects displays information about the object, including the resource path and tumor type.

import

The `import` method for `ProvGiga` objects reads the ProvGiga CSV file and extracts slide-level embeddings along with the slide names and tumor type. The embeddings are returned as a `tibble` with columns for slide names, tumor type, and embedding values.

Author(s)

Ilaria B., Marcel R.

Examples

```
## Importing a slide_level ProvGiga CSV file from a local path
slide_prov_url <- paste0(
    "https://store.cancerdatasci.org/provgigapath/slide_level/",
    "TCGA-OR-A5JJ-01Z-00-DX1.459B5DFE-47B1-426F-B009-7664C1B6FEEC.csv.gz"
)
slide_file <- file.path(tempdir(), basename(slide_prov_url))
download.file(slide_prov_url, destfile = slide_file)

ProvGiga(slide_file, tumorType = "TCGA_ACC") |>
    import()

## Importing a slide_level ProvGiga CSV file from a URL
ProvGiga(slide_prov_url, tumorType = "TCGA_ACC") |>
    import()

## Import tile_level ProvGiga CSV file from a URL
tile_prov_url <- paste0(
   "https://store.cancerdatasci.org/provgigapath/tile_level/",
   "TCGA-AA-3556-01Z-00-DX1.63a74b91-44e8-4ffd-8737-bcf6992183c3.csv.gz"
)

ProvGiga(tile_prov_url, tumorType = "TCGA_COAD") |>
    import()
```

ProvGigaList *ProvGigaList class for handling lists of ProvGiga objects*

Description

The `ProvGigaList` class is a container for multiple `ProvGiga` objects, allowing for efficient management and manipulation of collections of ProvGiga data. It extends the `SimpleList` class from the `S4Vectors` package.

The `ProvGigaList` constructor function creates an instance of the `ProvGigaList` class. It accepts multiple `ProvGiga` objects, a vector of file paths or URLs, or a list of these elements.

Usage

```
ProvGigaList(..., is_url = TRUE, levels = "slide_level", parallel = FALSE)

## S4 method for signature 'ProvGigaList'
path(object, ...)

## S4 method for signature 'ProvGigaList,ANY,ANY'
import(con, format, text, ...)
```

Arguments

...	Multiple ProvGiga objects, a vector of file paths or URLs, or a list of these elements. For import, futher arguments are passed to the import method for the individual ProvGiga objects.
is_url	logical(1L) whether the input resources are URLs. Default is TRUE.
levels	character() the data levels for each resource. Default is "slide_level".
parallel	logical(1L) whether to use parallel processing to instantiate multiple ProvGiga objects. Default is FALSE.
object	A ProvGigaList object.
con	The connection from which data is loaded or to which data is saved. If this is a character vector, it is assumed to be a file name and a corresponding file connection is created and then closed after exporting the object. If it is a BiocFile derivative, the data is loaded from or saved to the underlying resource. If missing, the function will return the output as a character vector, rather than writing to a connection.
format	The format of the output. If missing and con is a file name, the format is derived from the file extension. This argument is unnecessary when con is a derivative of BiocFile.
text	If con is missing, this can be a character vector directly providing the string data to import.

Value

- A ProvGigaList object containing multiple ProvGiga objects.
- import-ProvGigaList: Either a single SummarizedExperiment (if all objects are the same level) or a list of SummarizedExperiment objects (if levels differ).
- getEmbeddings: A matrix of embeddings extracted from all slide-level ProvGiga objects in the list.

path

The path method for ProvGigaList objects retrieves the file paths or URLs of all contained ProvGiga objects.

import

The import method for ProvGigaList objects imports the data from all contained ProvGiga objects and returns a list of tibbles.

Examples

```
## slide level imports
slide_urls <- getCatalog("provgigapath") |>
    dplyr::filter(level == "slide_level", Project.ID == "TCGA-UVM") |>
    dplyr::slice(1:3) |>
    getFileURLs()
```

```
## set a temporary BiocFileCache cache location
old <- options(BiocFileCache.cache = tempdir())
on.exit(options(BiocFileCache.cache = old))

ProvGigaList(slide_urls) |>
   import(redownload = FALSE)

## tile level imports
tile_urls <- getCatalog("provgigapath") |>
   dplyr::filter(level == "tile_level", Project.ID == "TCGA-GBM") |>
   dplyr::slice(1:2) |>
   getFileURLs()

ProvGigaList(tile_urls) |>
   import(redownload = FALSE)
```

Index