Package: riot (via r-universe)

November 26, 2026

Title R Inputs/Outputs for Tractography

Version 1.2.0

Description Provides a unified input-output interface for reading and writing bundles — collections of 3-D polylines (streamlines) reconstructed by fibre-tracking algorithms applied to diffusion MRI data. Supported formats include VTK (.vtk), VTP (.vtp), medInria FDS (.fds), MRtrix TCK/TSF (.tck/.tsf), and TrackVis TRK (.trk), with additional formats accessible via the 'DIPY' Python library through 'reticulate'. Bundles are represented in R using the S7 classes defined in the 'fiber' package: a single tract (class 'streamline') stores its coordinates, per-point scalar attributes, and per-streamline scalar attributes in separate, typed slots; a collection of tracts (class 'bundle') wraps a list of streamlines together with optional bundle-level metadata. VTK support is provided at compile time by the 'rvtk' package, which supplies pre-compiled VTK libraries and headers so that no manual VTK installation is required on most platforms.

License MIT + file LICENSE

Encoding UTF-8

LinkingTo cpp11,

Imports cli, fiber, fs, rlang, rvtk (>= 0.2.0)

Suggests knitr, pkgdown, quarto, reticulate, tinytest

VignetteBuilder quarto

SystemRequirements Python (>= 3.7) with the 'dipy' package is optional; required only for TRX, FIB, and DPY bundle formats.

URL https://github.com/tractoverse/riot, https://tractoverse.github.io/riot/

BugReports https://github.com/tractoverse/riot/issues

Config/roxygen2/version 8.1.1

Config/roxygen2/markdown TRUE

NeedsCompilation yes

Author Aymeric Stamm [aut, cre, cph] (ORCID:
<https://orcid.org/0000-0002-8725-3654>), Jeroen Ooms [ctb],
Lee Thomason [ctb] (Author of the TinyXML2 C++ XML parser), Tim
Schäfer [ctb] (Author of the freesurferformats R package)

Maintainer Aymeric Stamm <aymeric.stamm@cnrs.fr>

Config/pak/sysreqs python3

Repository https://cran.r-universe.dev

Date/Publication 2026-11-26 20:19:03 UTC

RemoteUrl https://github.com/cran/riot

RemoteRef HEAD

RemoteSha a966999e860a3896b21ae060f187a89f968e4217

Contents

read_bundle 2
write_bundle 3

Index **5**

`read_bundle` *Import bundles into R*

Description

This is the go-to function to import bundles into R. Based on both VTK and DIPY, we currently support eight different formats detailed in the documentation of input argument `file`.

Usage

```
read_bundle(file, reference_file = NULL, bundle_data = list())
```

Arguments

`file` A string specifying the path to the file containing the tractography data. Currently supported files are:

- standard VTK formats `.vtk` and `.vtp`,
- medInria `.fds` format,
- MRtrix `.tck`/`.tsf` format,
- TrackVis `.trk` and `.trx` formats,
- DIPY `.dpy` format,
- `.fib` format.

`reference_file` A string specifying the path to a reference image file. This is only required when importing `.trx`, `.fib`, or `.dpy` files, as these formats do not contain spatial information about the image space. The reference image is used to correctly position the bundle in the appropriate space. Default is `NULL`.

`bundle_data` A named list of bundle-level metadata to store in the `@bundle_data` slot of the returned bundle object (e.g. the affine transform used during tracking, subject identifier, or any other scalar/vector annotation). Ignored with a warning when the file contains a single streamline (in which case a streamline is returned). Default is `list()`.

Value

A bundle object when the file contains multiple streamlines, or a streamline object when it contains exactly one. Each `streamline` stores its coordinates as an `n x 3` numeric matrix in the `@points` slot (columns `"X"`, `"Y"`, `"Z"`, one row per point along the tract). Additional per-point scalar attributes, when present in the source file, appear in the `@point_data` slot, and per-streamline attributes in the `@streamline_data` slot. When a `bundle` is returned and `bundle_data` is non-empty, its `@bundle_data` slot is populated with the supplied list.

See Also

`write_bundle()` to export bundles from R.

Examples

```
uf_left_vtk <- read_bundle(system.file("extdata", "UF_left.vtk",  package = "riot"))

# Attach bundle-level metadata
uf_left_vtk2 <- read_bundle(
  system.file("extdata", "UF_left.vtk", package = "riot"),
  bundle_data = list(subject = "sub-01", hemisphere = "left")
)
uf_left_vtk2@bundle_data
```

---

write_bundle *Export bundles from R*

Description

This function exports a bundle stored as a bundle object to a file in one of the supported formats. Supported formats include VTK (`.vtk`), VTP (`.vtp`), FDS (`.fds`), TRK (`.trk`), TCK (`.tck`), TRX (`.trx`), FIB (`.fib`), and DPY (`.dpy`). For formats that require a reference image (such as TRX, FIB, and DPY), the user must provide the path to a reference image file.

Usage

```
write_bundle(x, file, reference_file = NULL)
```

Arguments

x	A bundle object.
file	A string specifying the path to the file containing the tractography data. Currently supported files are: • standard VTK formats .vtk and .vtp, • medInria .fds format, • MRtrix .tck/.tsf format, • TrackVis .trk and .trx formats, • DIPY .dpy format, • .fib format.
reference_file	A string specifying the path to a reference image file. This is only required when importing .trx, .fib, or .dpy files, as these formats do not contain spatial information about the image space. The reference image is used to correctly position the bundle in the appropriate space. Default is NULL.

Details

Warning: we rely on DIPY to provide support to save bundles in .trk, .trx, .tck, .dpy and .fib formats. Among these formats, only .trk and .trx formats are able to keep track of additional attributes assigned to either streamlines or points.

Value

The input bundle (invisibly) so that the function can be used in pipes.

See Also

read_bundle() to import bundles into R.

Examples

```
uf_left  <- read_bundle(system.file("extdata", "UF_left.vtp",  package = "riot"))
## Not run:
out <- fs::file_temp(ext = ".vtp")
write_bundle(uf_left, file = out)

## End(Not run)
```

Index

bundle, *3*, *4*

read_bundle, 2
read_bundle(), *4*

streamline, *3*

write_bundle, 3
write_bundle(), *3*