

# Package ‘vol2birdR’

February 21, 2025

**Type** Package

**Title** Vertical Profiles of Biological Signals in Weather Radar Data

**Version** 1.0.7

**Date** 2025-02-21

**Description** 'R' implementation of the 'vol2bird' software for generating vertical profiles of birds and other biological signals in weather radar data. See Dokter et al. (2011) <[doi:10.1098/rsif.2010.0116](https://doi.org/10.1098/rsif.2010.0116)> for a paper describing the methodology.

**License** LGPL (>= 3)

**Language** en-US

**URL** <https://github.com/adokter/vol2birdR/>,  
<https://adriaandokter.com/vol2bird/>

**BugReports** <https://github.com/adokter/vol2birdR/issues>

**Depends** R (>= 4.0.0)

**Imports** assertthat, methods, pkgbuild, Rcpp (>= 1.0.4), rlang, utils,  
withr

**LinkingTo** Rcpp, RcppGSL

**RoxygenNote** 7.3.2

**VignetteBuilder** knitr

**SystemRequirements** GNU make, GSL, HDF5, PROJ

**NeedsCompilation** yes

**RcppModules** RaveIO, PolarVolume, Vol2Bird, Vol2BirdConfig

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Copyright** file COPYRIGHTS

**Encoding** UTF-8

**Author** Anders Henja [aut] (vol2birdR package author and author of RAVE and HLHDF libraries),  
Adriaan M. Dokter [aut, cre] (<<https://orcid.org/0000-0001-6573-066X>>),

Alexander Tedeschi [ctb] (<<https://orcid.org/0000-0003-0772-6931>>),  
 Tsung-Yu Lin [ctb] (contributor MistNet segmentation model),  
 Subrانشu Maji [ctb] (contributor MistNet segmentation model),  
 Daniel Sheldon [ctb] (contributor MistNet segmentation model),  
 Bart Kranstauber [ctb] (<<https://orcid.org/0000-0001-8303-780X>>),  
 Jurriaan H. Spaaks [ctb] (contributor to vol2bird library),  
 Lourens Veen [ctb] (contributor to vol2bird library),  
 Iwan Holleman [ctb] (contributor to vol2bird library),  
 Hidde Leijnse [ctb] (contributor to vol2bird library),  
 John H. Merritt [ctb, cph] (author of RSL library),  
 Bart Kelley [ctb] (contributor and maintainer of RSL library),  
 Mark Couture [ctb] (author of iris2odim add-on to RAVE library),  
 Daniel Falbel [ctb] (contributor of original idea for building with  
 torch support),  
 Swedish Meteorological and Hydrological Institute, SMHI [cph]  
 (copyright holder of HLHDF and RAVE libraries),  
 GloBAM [fnd] (<https://globam.science>)

**Maintainer** Adriaan M. Dokter <vol2birdr@cornell.edu>

**Repository** CRAN

**Date/Publication** 2025-02-21 18:50:01 UTC

## Contents

get_install_urls . . . . .	2
install_mistnet . . . . .	3
install_mistnet_from_file . . . . .	4
install_mistnet_model . . . . .	6
mistnet_exists . . . . .	7
nexrad_station_file . . . . .	7
rsl2odim . . . . .	8
torch_install_path . . . . .	9
vol2bird . . . . .	10
vol2bird_config . . . . .	11
vol2bird_version . . . . .	15

**Index** **16**

---

get\_install\_urls *List of installation files to download*

---

## Description

List the 'LibTorch' and 'MistNet' files to download as local files in order to proceed with `install_mistnet_from_file()`.

## Usage

```
get_install_urls(version = "1.10.2", type = install_type(version = version))
```

**Arguments**

version	The 'LibTorch' version to install.
type	The installation type for 'LibTorch'. Valid value is currently "cpu".

**Value**

a named list with character urls

**See Also**

- [install\\_mistnet\\_from\\_file\(\)](#)

---

install_mistnet	<i>Install 'MistNet' libraries</i>
-----------------	------------------------------------

---

**Description**

Installs libraries and dependencies for using 'MistNet'.

**Usage**

```
install_mistnet(
  version = "1.12.1",
  reinstall = FALSE,
  path = install_path(),
  timeout = 360,
  ...
)
```

**Arguments**

version	The 'LibTorch' version to install.
reinstall	Re-install 'MistNet' even if its already installed?
path	Optional path to install or check for an already existing installation.
timeout	Optional timeout in seconds for large file download.
...	other optional arguments (like `load` for manual installation).

**Details**

By default libraries are installed in the 'tools::R\_user\_dir("vol2birdR", "data")' directory.

When using path to install in a specific location, make sure the MISTNET\_HOME environment variable is set to this same path to reuse this installation.

The TORCH\_INSTALL environment variable can be set to 0 to prevent auto-installing 'LibTorch' and TORCH\_LOAD set to 0 to avoid loading dependencies automatically. These environment variables are meant for advanced use cases and troubleshooting only.

When timeout error occurs during library archive download, or length of downloaded files differ from reported length, an increase of the timeout value should help.

**Value**

no value returned. Installs libraries into the package

**See Also**

- [install\\_mistnet\\_from\\_file\(\)](#)

**Examples**

```
install_mistnet()
```

---

```
install_mistnet_from_file
```

*Install 'MistNet' libraries from files*

---

**Description**

Installs 'LibTorch' and 'MistNet' dependencies from files.

**Usage**

```
install_mistnet_from_file(  
  version = "1.12.1",  
  libtorch,  
  libmistnet,  
  mistnet_model = NULL,  
  ...  
)
```

**Arguments**

version	The 'LibTorch' version to install.
libtorch	The installation archive file to use for 'LibTorch'. Shall be a "file://" URL scheme.
libmistnet	The installation archive file to use for 'MistNet'. Shall be a "file://" URL scheme.
mistnet_model	The installation archive file to use for the model. Shall be a "file://" URL scheme. Is optional!
...	other parameters to be passed to <code>install_torch()</code>

**Details**

When `install_mistnet()` initiated download is not possible, but installation archive files are present on local filesystem, `install_mistnet_from_file()` can be used as a workaround to installation issues. "libtorch" is the archive containing all 'LibTorch' modules, and "libmistnet" is the 'C' interface to 'LibTorch' that is used for the 'R' package. Both are highly platform dependent, and should be checked through `get_install_urls()`

```
> get_install_urls()
$libtorch
[1] "https://download.pytorch.org/libtorch/cpu/libtorch-cxx11-abi-shared-with-deps-1.10.2%2Bcpu.zip"

$libmistnet
[1] "https://s3.amazonaws.com/vol2bird-builds/vol2birdr/refs/heads/main/latest/Linux-cpu.zip"

$mistnet_model
[1] "http://mistnet.s3.amazonaws.com/mistnet_nexrad.pt"
```

In a terminal, download above zip-files.

```
%> mkdir /tmp/myfiles
%> cd /tmp/myfiles
%> wget https://download.pytorch.org/libtorch/cpu/libtorch-cxx11-abi-shared-with-deps-1.10.2%2Bcpu.zip
%> wget https://s3.amazonaws.com/vol2bird-builds/vol2birdr/refs/heads/main/latest/Linux-cpu.zip
%> wget http://mistnet.s3.amazonaws.com/mistnet_nexrad.pt
```

Then in R, type:

```
> install_mistnet_from_file(libtorch="file:///tmp/myfiles/libtorch-cxx11-abi-shared-with-deps-1.10.2%2Bcpu.zip",
  libmistnet="file:///tmp/myfiles/Linux-cpu.zip",
  mistnet_model="file:///tmp/myfiles/mistnet_nexrad.pt")
```

**Value**

a list with character urls

**See Also**

- [install\\_mistnet\(\)](#)

**Examples**

```
# get paths to files to be downloaded
get_install_urls()
# download the files to a directory on disk, e.g. to /tmp/myfile,
# then install with:
## Not run:
install_mistnet_from_file(
  libtorch="file:///tmp/myfiles/libtorch-cxx11-abi-shared-with-deps-1.10.2+cpu.zip",
  libmistnet="file:///tmp/myfiles/Linux-cpu.zip",
```

```
mistnet_model="file:///tmp/myfiles/mistnet_nexrad.pt")
## End(Not run)
```

---

install\_mistnet\_model *Install 'MistNet' model file*

---

### Description

Installs the 'MistNet' model file in 'PyTorch' format

### Usage

```
install_mistnet_model(
  reinstall = FALSE,
  path = file.path(torch_install_path(), "data", "mistnet_nexrad.pt"),
  timeout = 1800,
  from_url = "http://mistnet.s3.amazonaws.com/mistnet_nexrad.pt",
  method = "libcurl",
  ...
)
```

### Arguments

reinstall	Re-install the model even if its already installed
path	Optional path to install or check for an already existing installation.
timeout	Optional timeout in seconds for large file download.
from_url	From where the 'MistNet' model file should be downloaded.
method	The download method to use, see <a href="#">download.file</a>
...	other optional arguments (like <code>`load`</code> for manual installation).

### Details

Download and install the 'MistNet' model file. By default the library is downloaded to `data/mistnet_nexrad.pt` in the `'tools::R_user_dir("vol2birdR", "data")'` directory.

Alternatively, the model file can be downloaded to a different location, which has the advantage that it doesn't have to be redownloaded after a reinstall of 'vol2birdR'.

'vol2birdR' will automatically detect the model file if it is downloaded to `/opt/vol2bird/etc/mistnet_nexrad.pt`, which can be done as follows

```
install_mistnet_model(path="/opt/vol2bird/etc/mistnet_nexrad.pt")
```

### Value

No value returned, this function downloads a file

**Examples**

```
install_mistnet_model()
```

---

mistnet_exists	<i>Checks if the 'LibTorch' and 'MistNet' libraries have been installed or not.</i>
----------------	---

---

**Description**

Checks if the 'LibTorch' and 'MistNet' libraries have been installed or not.

**Usage**

```
mistnet_exists()
```

**Value**

TRUE if both 'LibTorch' and 'MistNet' libraries can be found, otherwise FALSE

---

nexrad_station_file	<i>Retrieve or set the nexrad location file</i>
---------------------	---

---

**Description**

Retrieves and sets the path of the RSL nexrad location file

**Usage**

```
nexrad_station_file(file)
```

**Arguments**

file	A string containing the path of a location file. Do not specify to retrieve path of current location file.
------	--

**Details**

The RSL library stores the locations and names of NEXRAD stations in a fixed-width text file. This function retrieves the path of the location file, and allows one to set the path to a different location file.

**Value**

The path of the nexrad location file

## Examples

```
# return current location file
nexrad_station_file()
# store nexrad station file path
file_path <- nexrad_station_file()
# set station location file
nexrad_station_file(file_path)
```

---

rsl2odim

*Convert a NEXRAD polar volume file to an ODIM polar volume file*


---

## Description

Convert a NEXRAD polar volume file to an ODIM polar volume file

## Usage

```
rsl2odim(
  file,
  config,
  pvolfile_out = "",
  verbose = TRUE,
  update_config = FALSE
)
```

## Arguments

file	Character (vector). Either a path to a single radar polar volume (pvol) file containing multiple scans/sweeps, or multiple paths to scan files containing a single scan/sweep. The file data format should be either 1) <b>ODIM</b> format, which is the implementation of the OPERA data information model in the <b>HDF5</b> format, 2) NEXRAD format supported by the ' <b>RSL</b> ' library or 3) Vaisala IRIS (IRIS RAW) format. IRIS format is not available on CRAN, see vol2birdR development version on Github.
config	optional configuration object of class Rcpp_Vol2BirdConfig, typically output from <a href="#">vol2bird_config</a>
pvolfile_out	Character. File name. When provided, writes a polar volume (pvol) file in the ODIM HDF5 format to disk. Useful for converting 'RSL' formats to ODIM, and for adding 'MistNet' segmentation output.
verbose	logical. When TRUE print profile output to console.
update_config	logical. When TRUE processing options that are determined based on input file characteristics are returned and updated in the object specified by the config argument. Do not set to TRUE when vol2bird() is used in loops like lapply() or in parallel processes.



**Value**

No value returned, creates a file specified by pvolfile\_out argument.

**See Also**

- [vol2bird\\_config\(\)](#)

**Examples**

```
# define filenames
nextrad_file <- paste0(tempdir(),"KBGM20221001_000243_V06")
odim_file <- paste0(tempdir(),"KBGM20221001_000243_V06.h5")
# download NEXRAD file:
download.file("https://noaa-nexrad-level2.s3.amazonaws.com/2022/10/01/KBGM/KBGM20221001_000243_V06",
destfile = nextrad_file, mode="wb")
# convert NEXRAD file to ODIM hdf5 format:
rsl2odim(nextrad_file, pvolfile_out = odim_file)
# clean up
file.remove(nextrad_file)
file.remove(odim_file)
```

---

torch\_install\_path      *Returns the 'LibTorch' installation path.*

---

**Description**

Returns the directory where the LibTorch library has been downloaded

**Usage**

```
torch_install_path()
```

**Value**

a character path

**Examples**

```
torch_install_path()
```

vol2bird

*Calculate a vertical profile (vp) from a polar volume (pvol) file***Description**

Calculates a vertical profile of biological scatterers (vp) from a polar volume (pvol) file using the algorithm [vol2bird](#) (Dokter et al. 2011 [doi:10.1098/rsif.2010.0116](#)).

**Usage**

```
vol2bird(
  file,
  config,
  vpfile = "",
  pvolfile_out = "",
  verbose = TRUE,
  update_config = FALSE
)
```

**Arguments**

file	Character (vector). Either a path to a single radar polar volume (pvol) file containing multiple scans/sweeps, or multiple paths to scan files containing a single scan/sweep. The file data format should be either 1) <b>ODIM</b> format, which is the implementation of the OPERA data information model in the <b>HDF5</b> format, 2) NEXRAD format supported by the ' <b>RSL</b> ' library or 3) Vaisala IRIS (IRIS RAW) format. IRIS format is not available on CRAN, see vol2birdR development version on Github.
config	optional configuration object of class Rcpp_Vol2BirdConfig, typically output from <a href="#">vol2bird_config</a>
vpfile	Character. File name. When provided with .csv extension, writes a vertical profile in <b>VPTS CSV format</b> . Provided with another or no extension, writes a vertical profile in the ODIM HDF5 format to disk.
pvolfile_out	Character. File name. When provided, writes a polar volume (pvol) file in the ODIM HDF5 format to disk. Useful for converting 'RSL' formats to ODIM, and for adding 'MistNet' segmentation output.
verbose	logical. When TRUE print profile output to console.
update_config	logical. When TRUE processing options that are determined based on input file characteristics are returned and updated in the object specified by the config argument. Do not set to TRUE when vol2bird() is used in loops like lapply() or in parallel processes.

**Value**

No value returned, creates a file specified by file argument

**See Also**

- [vol2bird\\_config\(\)](#)

**Examples**

```
# Locate the polar volume example file
pvolfiler <- system.file("extdata", "volume.h5", package = "vol2birdR")

# Create a configuration instance:
conf <- vol2bird_config()

# Define output file
output_file <- paste0(tempdir(), "/vp.h5")

# Calculate the profile:
vol2bird(file = pvolfiler, config = conf, vpfiler = output_file)
```

---

vol2bird_config	<i>Create a 'vol2bird' configuration instance</i>
-----------------	---

---

**Description**

Creates or copies a 'vol2bird' configuration instance of class Rcpp\_Vol2BirdConfig

**Usage**

```
vol2bird_config(config)
```

**Arguments**

config            a configuration instance to be copied.

**Details****Copying configuration instances:**

All processing options for [vol2bird\(\)](#) are set using a configuration instance of class Rcpp\_Vol2BirdConfig. In some cases it might be necessary to copy and modify configuration instance, for example when processing polar volume files with different settings. In these cases you can't copy the instance like:

```
config<-vol2bird_config()
extra_config<-config
```

In the above example, the config and extra\_config instances will both refer to the same object. (copy by reference). To avoid this (and make a copy by value), use:

```
config<-vol2bird_config()
# create a copy identical to object config:
extra_config<-vol2bird_config(config)
```

**User configuration options:**

The Rcpp\_Vol2BirdConfig class object sets the following 'vol2bird' processing options:

- `azimMax`: Numeric. The minimum azimuth (0-360 degrees) used for constructing the bird density profile
- `azimMin`: Numeric. The maximum azimuth (0-360 degrees) used for constructing the bird density profile
- `birdRadarCrossSection`: Numeric. Radar cross section in  $\text{cm}^2$
- `clutterMap`: Character. clutter map path and filename
- `clutterValueMin`: Numeric. sample volumes in the static cluttermap with a value above this threshold will be considered clutter-contaminated. Default 0.1
- `dbzType`: Character. Reflectivity factor quantity to use. Default DBZH
- `dualPol`: Logical. Whether to use dual-pol moments for filtering meteorological echoes. Default TRUE
- `elevMax`: Numeric. The minimum scan elevation in degrees used for constructing the bird density profile
- `elevMin`: Numeric. The maximum scan elevation in degrees used for constructing the bird density profile
- `layerThickness`: Numeric. The width/thickness of an altitude layer in m. Default 200
- `mistNetPath`: Character. Path of 'MistNet' segmentation model in pytorch (.pt) format
- `nLayers`: Integer. The number of layers in an altitude profile. Default 25
- `radarWavelength`: Numeric. The radar wavelength in cm to assume when unavailable as an attribute in the input file. Default 5.3
- `rangeMax`: Numeric. The maximum range in m used for constructing the bird density profile. Default 35000
- `rangeMin`: Numeric. The minimum range in m used for constructing the bird density profile. Default 5000
- `rhohvThresMin`: Numeric. Correlation coefficients higher than this threshold will be classified as precipitation. Default 0.95
- `singlePol`: Logical. Whether to use single-pol moments for filtering meteorological echoes. Default TRUE
- `stdDevMinBird`: Numeric. VVP Radial velocity standard deviation threshold. Default 2 m/s.
- `useClutterMap`: Logical. Whether to use a static clutter map. Default FALSE
- `useMistNet`: Logical. Whether to use the 'MistNet' segmentation model. Default FALSE.

**Advanced configuration options:**

Changing these settings is rarely needed.

- `cellEtaMin`: Numeric. Maximum mean reflectivity in  $\text{cm}^2/\text{km}^3$  for cells containing birds
- `cellStdDevMax`: Numeric. When analyzing precipitation cells, only cells for which the `stddev` of `vrad` (aka the texture) is less than `cellStdDevMax` are considered in the rest of the analysis
- `dbzThresMin`: Numeric. Minimum reflectivity factor of a gate to be considered for inclusion in a weather cell. Default 0 dBZ
- `dealiasRecycle`: Logical. Whether we should dealias all data once (default TRUE), or dealias for each profile individually (FALSE)

- dealiasVrad: Logical. Whether we should dealias the radial velocities. Default TRUE.
- etaMax: Numeric. Maximum reflectivity in  $\text{cm}^2/\text{km}^3$  for single gates containing birds. Default 36000
- exportBirdProfileAsJSONVar: Logical. Deprecated, do not use. Default FALSE
- fitVrad: Logical. Whether or not to fit a model to the observed vrad. Default TRUE
- maxNyquistDealias: Numeric. When all scans have nyquist velocity higher than this value, dealiasing is suppressed. Default 25 m/s.
- minNyquist: Numeric. Scans with Nyquist velocity lower than this value are excluded. Default 5 m/s.
- mistNetElevs: Numeric vector of length 5. Elevations to use in Cartesian projection for 'MistNet'. Default  $c(0.5, 1.5, 2.5, 3.5, 4.5)$
- mistNetElevsOnly: Logical. When TRUE (default), use only the specified elevation scans for 'MistNet' to calculate profile, otherwise use all available elevation scans
- requireVrad: Logical. For a range gate to contribute it should have a valid radial velocity. Default FALSE
- resample: Logical. Whether to resample the input polar volume. Downsampling speeds up the calculation. Default FALSE
- resampleNbins: Numeric. Resampled number of range bins. Ignored when resample is FALSE. Default 100
- resampleNrays: Numeric. Resampled number of azimuth bins. Ignored when resample is FALSE. Default 360
- resampleRscale: Numeric. Resampled range gate length in m. Ignored when resample is FALSE. Default 500 m.

#### Algorithm constants:

Changing any of these constants is not recommended

- constant\_absVDifMax: Numeric. After fitting the radial velocity data, throw out any VRAD observations that are more than absVDifMax away from the fitted value as outliers. Default 10
- constant\_areaCellMin: Numeric. When analyzing cells, areaCellMin determines the minimum size of a cell to be considered in the rest of the analysis. in  $\text{km}^2$ . Default 0.5
- constant\_cellClutterFractionMax: Cells with clutter fractions above this value are likely not birds. Default 0.5
- constant\_chisqMin: Minimum standard deviation of the VVP fit. Default  $1e-05$
- constant\_fringeDist: Each identified weather cell is grown by a distance equal to 'fringeDist' using a region-growing approach. Default 5000
- constant\_nAzimNeighborhood: vrad's texture is calculated based on the local neighborhood. The neighborhood size in the azimuth direction is equal to this value. Default 3
- constant\_nBinsGap: When determining whether there are enough vrad observations in each direction, use nBinsGap sectors. Default 8
- constant\_nCountMin: The minimum number of neighbors for the texture value to be considered valid, as used in calcTexture(). Default 4
- constant\_nNeighborsMin: the minimum number of direct neighbors with dbz value above dbzThresMin as used in findWeatherCells(). Default 5
- constant\_nObsGapMin: there should be at least this many vrad observations in each sector. Default 5

- `constant_nPointsIncludedMin`: when calculating the altitude-layer averaged dbz, there should be at least this many valid data points. Default 25
- `constant_nRangNeighborhood`: `vrad`'s texture is calculated based on the local neighborhood. The neighborhood size in the range direction is equal to this value. Default 3
- `constant_refracIndex`: Refractive index of the scatterers. Default equal to water 0.964
- `constant_vradMin`: When analyzing cells, radial velocities lower than `vradMin` are treated as clutter. Default 1 m/s.

### Debug printing options:

Enable these printing options only for debugging purposes in a terminal, since large amounts of data will be dumped into the console.

- `printCell`: Logical. Print precipitation cell data to stderr. Default FALSE
- `printCellProp`: Logical. Print precipitation cell properties to stderr. Default FALSE
- `printClut`: Logical. Print clutter data to stderr. Default FALSE
- `printDbz`: Logical. Print reflectivity factor data to stderr. Default FALSE
- `printDealias`: Logical. FALSE
- `printOptions`: Logical. Print options to stderr. Default FALSE
- `printPointsArray`: Logical. Print the 'points' array to stderr. Default FALSE
- `printProfileVar`: Logical. Print profile data to stderr. Default FALSE
- `printRhoHV`: Logical. Print correlation coefficient data to stderr. Default FALSE
- `printTex`: Logical. Print radial velocity texture data to stderr. Default FALSE
- `printVrad`: Logical. Print radial velocity data to stderr. Default FALSE

### Value

an object of class `Rcpp_Vol2BirdConfig`

### See Also

- [vol2bird\(\)](#)

### Examples

```
# create a configuration instance
config <- vol2bird_config()
# list the the configuration elements:
config
# change the maximum range included in the profile generation to 40 km:
config$rangeMax <- 40000
# make a copy of the configuration instance:
config_copy <- vol2bird_config(config)
```

---

vol2bird_version	<i>Return 'vol2bird' version</i>
------------------	----------------------------------

---

**Description**

Return version of the 'vol2bird' algorithm

**Usage**

```
vol2bird_version()
```

**Value**

an object of class [numeric\\_version](#)

**Examples**

```
# check installed 'vol2bird' version:  
vol2bird_version()
```

# Index

`download.file`, [6](#)

`get_install_urls`, [2](#)

`get_install_urls()`, [5](#)

`install_mistnet`, [3](#)

`install_mistnet()`, [5](#)

`install_mistnet_from_file`, [4](#)

`install_mistnet_from_file()`, [2-5](#)

`install_mistnet_model`, [6](#)

`mistnet_exists`, [7](#)

`nexrad_station_file`, [7](#)

`numeric_version`, [15](#)

`rsl2odim`, [8](#)

`torch_install_path`, [9](#)

`vol2bird`, [10](#)

`vol2bird()`, [11](#), [14](#)

`vol2bird_config`, [8](#), [10](#), [11](#)

`vol2bird_config()`, [9](#), [11](#)

`vol2bird_version`, [15](#)