

Package: fvfmR (via r-universe)

October 17, 2026

Type Package

Title Chlorophyll Fluorescence (Fv/Fm) Analysis of Leaf Disc Images

Version 1.0.0

Description Semi-automated analysis of chlorophyll fluorescence images of leaf discs taken with a 'Walz' 'IMAGING-PAM' fluorometer. Provides an R interface to the 'fvfmPy' 'Python' package, which detects leaf discs, assigns them to a grid of rows and columns, and computes the maximum quantum efficiency of photosystem II, $Fv/Fm = (Fm - Fo) / Fm$, for each disc. The interactive 'fvfmPy' window is launched from R and the results are returned to R as a data frame.

License MIT + file LICENSE

URL <https://github.com/garenj/fvfmR>, <https://github.com/garenj/fvfmPy>

BugReports <https://github.com/garenj/fvfmR/issues>

Encoding UTF-8

Language en-GB

Imports reticulate, utils

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

SystemRequirements Python (>= 3.10), fvfmPy (>= 1.0.0)

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libpng-dev python3

Repository <https://garenj.r-universe.dev>

Date/Publication 2026-10-07 04:39:59 UTC

RemoteUrl <https://github.com/garenj/fvfmr>

RemoteRef HEAD

RemoteSha 3238b543286ce82064a0c6a21413a16d84aa0ce2

Contents

fvfm_example	2
fvfm_setup	3
run_fvfm	4
Index	6

fvfm_example	<i>Path to the example images</i>
--------------	-----------------------------------

Description

fvfmR ships with two example .pim images from a Walz ImagingPAM. Use them to try out [run_fvfm](#) or to check your installation.

Usage

```
fvfm_example(file = NULL)
```

Arguments

file	Name of an example file. NULL (default) returns the folder containing all example images.
------	---

Details

- example1.pim A full 5 x 9 tray. The default settings detect all 45 ROIs and the correct grid, so it can be analysed straight away.
- example2.pim A 6 x 10 tray with three empty cells (column 10, rows 4 to 6), which needs some adjustment before analysis:
- Fill the three empty cells with dummy ROIs (double-click) so that every ROI is assigned to the correct row and column.
 - Rotating the image slightly helps align the grid.
 - One leaf is split into two ROIs with the default settings; increasing *Watershed segmentation size* slightly (to about 36) merges them.
- Remove any extra ROIs outside the leaves by double-clicking them.

Value

Path to the example folder, or to file within it.

Examples

```
fvfm_example()
list.files(fvfm_example())
fvfm_example("example1.pim")
## Not run:
results <- run_fvfm(fvfm_example())

## End(Not run)
```

fvfm_setup

Set up the Python environment for the fvfmR package

Description

Creates a dedicated virtual environment called fvfm-env and installs the **fvfmPy** Python package and its dependencies into it. Run this once after installing the package, or again to update fvfmPy. The environment is created in reticulate's virtualenv folder (see [virtualenv_root](#)) and needs about 1.5 GB of disk space.

Usage

```
fvfm_setup(python = NULL, source = NULL, ask = interactive())
```

Arguments

python	Path to a Python 3.10+ executable used to create the virtual environment. NULL (default) lets reticulate find Python on your system.
source	Optional path to a local fvfmPy source (.tar.gz) or wheel (.whl) file. NULL (default) installs the latest release from PyPI.
ask	Logical. Ask for confirmation before creating or modifying the Python environment? Default: TRUE in interactive sessions.

Details

By default the latest release of fvfmPy is installed from PyPI. Use source to install from a downloaded .tar.gz or .whl file instead. Running fvfm_setup() again updates fvfmPy to the latest release.

If you already have a Python environment with fvfmPy installed, you can skip this step and point the package at it with options(fvfmR.python = "/path/to/venv/bin/python").

Value

Invisible path to the environment's Python executable, or NULL if setup was cancelled.

Examples

```
## Not run:
fvfm_setup()
fvfm_setup(source = "~/Downloads/fvfmPy-1.0.0.tar.gz")

## End(Not run)
```

run_fvfm

Run the fvfmPy leaf disc analysis GUI

Description

Opens the fvfmPy window, either on a folder of .pim, .tif or .tiff images from a Walz Imaging-PAM or with no folder open (choose one with the *Open folder* button). Use the window to check the detected leaf discs, adjust rotation, cropping, rows/columns and segmentation settings, and press *Analyze* (or Enter) to log each image. See the [fvfmPy User's Guide](#) for details.

Usage

```
run_fvfm(directory = NULL, output = NULL, overwrite = FALSE, python = NULL)
```

Arguments

directory	Optional path to the folder containing the images. NULL (default) opens the window without a folder.
output	Optional path to a CSV file to write the results to.
overwrite	Logical. Overwrite output if it already exists? Default: FALSE.
python	Optional path to a Python executable with fvfmPy installed. See fvfm_setup for how the default is chosen.

Details

R waits while the window is open. When you close it, every observation logged during the session is returned as a data frame, so there is no need to use the *Save results* button (it still works if you want an extra copy).

The GUI runs as a separate Python process, so a problem in the GUI cannot crash your R session.

Value

Invisibly, a data frame with one row per leaf disc and columns filename, row, col, leaf_number, centroid_x, centroid_y, mean_Fo, mean_Fm and FvFm; or NULL if no images were analysed.

Examples

```
## Not run:
results <- run_fvfm() # choose a folder in the window
results <- run_fvfm("~/Dropbox/Experiment1/images")
run_fvfm("~/Dropbox/Experiment1/images", output = "results.csv")

## End(Not run)
```

Index

fvfm_example, [2](#)

fvfm_setup, [3](#), [4](#)

run_fvfm, [2](#), [4](#)

virtualenv_root, [3](#)