

# Package ‘dfoliatR’

May 8, 2026

**Type** Package

**Title** Detection and Analysis of Insect Defoliation Signals in Tree Rings

**Version** 0.3.0

**Description** Tools to identify, quantify, analyze, and visualize growth suppression events in tree rings that are often produced by insect defoliation. Described in Guiterman et al. (2020) <[doi:10.1016/j.dendro.2020.125750](https://doi.org/10.1016/j.dendro.2020.125750)>.

**URL** <https://chguiterman.github.io/dfoliatR/>

**BugReports** <https://github.com/chguiterman/dfoliatR/issues>

**Depends** R (>= 3.2)

**License** GPL (>= 3)

**Suggests** testthat, vdiff, knitr, rmarkdown

**Imports** dplR, dplyr, forcats, ggplot2, ggpubr, magrittr, purrr, rlang, stats, tidyr, glue, tibble

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.3

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Chris Guiterman [aut, cre] (ORCID: <<https://orcid.org/0000-0002-9706-9332>>),  
Ann Lynch [aut] (ORCID: <<https://orcid.org/0000-0002-8043-3855>>),  
Jodi Axelson [aut] (ORCID: <<https://orcid.org/0000-0002-1799-6364>>)

**Maintainer** Chris Guiterman <[christopher.guiterman@noaa.gov](mailto:christopher.guiterman@noaa.gov)>

**Repository** CRAN

**Date/Publication** 2023-08-09 22:10:02 UTC

## Contents

as.defol	2
as.obr	3
as_defol	4
as_obr	4
defol	5
defoliate_trees	6
defol_stats	7
dmj_defol	8
dmj_h	8
dmj_nh	9
dmj_obr	9
efk_defol	10
efk_h	10
efk_nh	11
efk_obr	11
get_defol_events	12
gsi	12
id_defoliation	13
is.defol	14
is.obr	15
obr	15
outbreak	16
outbreak_stats	17
plot.defol	18
plot_defol	19
plot_outbreak	19
sample_depth	20
series_names	20
stack_defoliation	21
<b>Index</b>	<b>22</b>

---

as.defol	<i>Alias to <a href="#">as_defol()</a></i>
----------	--

---

### Description

Alias to [as\\_defol\(\)](#)

### Usage

```
as.defol(x)
```

### Arguments

x A data frame or list-like object to cast. Must have named elements for "year", "series", "gsi", "ngsi", and "defol\_status".

**Value**

x cast to a defol object

**Examples**

```
data(dmj_defol)
example_data <- as.data.frame(dmj_defol)
is.defol(example_data)
back_to_defol <- as_defol(example_data)
is.defol(back_to_defol)
```

---

as.obr

*Alias to [as\\_obr\(\)](#)*

---

**Description**

Alias to [as\\_obr\(\)](#)

**Usage**

```
as.obr(x)
```

**Arguments**

x A data frame or list-like object to cast. Must have named elements for "year", "samp\_depth", "num\_defol", "perc\_defol", "num\_max\_defol", "perc\_max\_defol", "mean\_gsi", "mean\_ngsi", "outbreak\_status".

**Value**

x cast to an obr object

**Examples**

```
data(dmj_obr)
example_data <- as.data.frame(dmj_obr)
is.obr(example_data)
back_to_obr <- as_obr(example_data)
is.obr(back_to_obr)
```

---

as_defol	<i>Cast data frame to list-like defol object</i>
----------	--

---

**Description**

Cast data frame to list-like defol object

**Usage**

```
as_defol(x)
```

**Arguments**

x                    A data frame or list-like object to cast. Must have named elements for "year", "series", "gsi", "ngsi", and "defol\_status".

**Value**

x cast to a defol object

**Examples**

```
data(dmj_defol)
example_data <- as.data.frame(dmj_defol)
is.defol(example_data)
back_to_defol <- as_defol(example_data)
is.defol(back_to_defol)
```

---

as_obr	<i>Cast data frame to list-like obr object</i>
--------	--

---

**Description**

Cast data frame to list-like obr object

**Usage**

```
as_obr(x)
```

**Arguments**

x                    A data frame or list-like object to cast. Must have named elements for "year", "samp\_depth", "num\_defol", "perc\_defol", "num\_max\_defol", "perc\_max\_defol", "mean\_gsi", "mean\_ngsi", "outbreak\_status".

**Value**

x cast to an obr object

**Examples**

```
data(dmj_obr)
example_data <- as.data.frame(dmj_obr)
is.obr(example_data)
back_to_obr <- as_obr(example_data)
is.obr(back_to_obr)
```

---

defol

*Constructor for S3 defol class*

---

**Description**

Constructor for S3 defol class

**Usage**

```
defol(year, series, gsi, ngsi, defol_status)
```

**Arguments**

year	An n-length integer vector of observation years
series	An n-length factor or character vector of series names
gsi	An n-length numeric vector of growth suppression index, such as calculated by <a href="#">gsi()</a>
ngsi	An n-length numeric vector of normalized gsi, such as calculated by <a href="#">gsi()</a> .
defol_status	An n-length factor or character vector denoting the defoliation event status of each year. This uses a controlled vocabulary, see <code>dfoliatR:::make_defol_status</code> for possible values.

**Value**

a tree-level defol object

---

defoliate\_trees      *Identify defoliation events in host trees*

---

## Description

`defoliate_trees()` is the starting point for most analyses of insect defoliation signals preserved in the growth patterns of trees. It requires individual-tree standardized measurements from potential host trees and a tree-ring chronology from a nearby non-host species. First, `defoliate_trees()` combines these tree-ring indices by calling `gsi()` to perform a "correction" of the host-tree indices to remove the climatic influences on tree growth as represented by the non-host chronology. This should isolate a disturbance-related signal. Second, `defoliate_trees()`, runs `id_defoliation()`, which completes a runs analyses to evaluate sequences of negative departures in the host tree growth series (`ngsi`) for potential defoliation events.

## Usage

```
defoliate_trees(
  host_tree,
  nonhost_chron = NULL,
  duration_years = 8,
  max_reduction = -1.28,
  bridge_events = FALSE,
  series_end_event = FALSE,
  list_output = FALSE
)
```

## Arguments

<code>host_tree</code>	A <code>dplR::rwl</code> object containing the tree-level growth series for all host trees to be compared to the non-host chronology.
<code>nonhost_chron</code>	A <code>dplR::rwl</code> object containing a single non-host chronology. If blank, defoliation events will be inferred on the <code>host_tree</code> series as provided. It is incumbent on the user to ensure the <code>host_tree</code> series are properly prepared for analyses when there is no <code>nonhost_chron</code> provided.
<code>duration_years</code>	The minimum length of time in which the tree is considered to be in defoliation.
<code>max_reduction</code>	The minimum value of <code>ngsi</code> required to be considered a defoliation event. If a sequence of negative <code>ngsi</code> values does not reach this threshold, the potential event is rejected. Defaults to -1.28.
<code>bridge_events</code>	Binary, defaults to <code>FALSE</code> . This option allows for successive events that are separated by a single year to be bridged and become one event. It should be used cautiously and closely evaluated to ensure the likelihood that the two (or more) events are actually one long event.
<code>series_end_event</code>	Binary, defaults to <code>FALSE</code> . This option allows the user to identify an event occurring at the time of sampling as a defoliation event, regardless of duration.

Including it will help to quantify periodicity and extent of an outbreak. This should only be used if the user has direct knowledge of an ongoing defoliation event when the trees were sampled.

`list_output` Defaults to FALSE. This option is to output a long list object containing a separate `data.frame` for each series in `host_tree` that includes the input series and the `nonhost_chron`, the corrected series, and the character string identifying the defoliation events.

### Value

By default this returns a long-form data frame of tree-level growth suppression indices and identified defoliation events. If `list_output = TRUE`, it returns a list object with each element containing a `data.frame` `rwl` object of the host and non-host series, plus the outputs from `gsi()`. The list object is useful for assessing the effects of running `gsi()` on the host and nonhost data.

### Note

Other functions in `dfoliatR`, like `outbreak()` and `plot_defol()`, require a long-form data frame identifiable as a `defol()` object. Selecting `list_output = TRUE` will trigger errors in running other functions.

### Examples

```
# Load host and non-host data
data("dmj_h") # Host trees
data("dmj_nh") # Non-host chronology

dmj_defol <- defoliate_trees(dmj_h, dmj_nh)
```

---

`defol_stats`

*Descriptive statistics for defoliation trees*

---

### Description

Descriptive statistics for defoliation trees

### Usage

```
defol_stats(x)
```

### Arguments

`x` A `defol` object after running `defoliate_trees()`.

### Value

A data frame containing tree/series-level statistics.

**Note**

If series-end-events are present, they are omitted from calculations of total event years and mean duration.

**Examples**

```
data("dmj_defol")
defol_stats(dmj_defol)
```

---

dmj_defol	<i>Demi John dfol object</i>
-----------	------------------------------

---

**Description**

Produced by running `defoliate_trees(dmj_h, dmj_nh, series_end_events=TRUE)`

**Usage**

```
dmj_defol
```

**Format**

A defol object with 4267 rows and 5 columns

---

dmj_h	<i>Demi John Douglas-fir</i>
-------	------------------------------

---

**Description**

Western spruce budworm host series

**Usage**

```
dmj_h
```

**Format**

An `rwl` object with 17 tree-level series, standardized in ARSTAN. Dates range from 1620-1997.

---

dmj_nh	<i>Demi John area ponderosa pine</i>
--------	--------------------------------------

---

**Description**

Non-host pair to Demi John Douglas-fir dmj\_h

**Usage**

dmj\_nh

**Format**

An rwl object with 1 series, 1675-1997.

---

dmj_obr	<i>Demi John outbreak object</i>
---------	----------------------------------

---

**Description**

Produced by running outbreak(dmj\_defol)

**Usage**

dmj\_obr

**Format**

An object of class obr (inherits from data.frame) with 323 rows and 9 columns.

**Details**

@format An outbreak object with 323 rows and 9 columns

---

efk\_defol                      *East Fork defol object*

---

**Description**

Produced by running `defoliate_trees(efk_h, efk_nh, series_end_event = TRUE)`

**Usage**

`efk_defol`

**Format**

A defol object with 5142 rows and 5 columns

---

efk\_h                              *East Fork Jemez River Douglas-fir*

---

**Description**

Western spruce budworm host series

**Usage**

`efk_h`

**Format**

An `rwl` object with 37 tree-level series, standardized in ARSTAN. Dates range from 1776-1987.

**References**

Swetnam, T.W., Lynch, A.M., 1993. Multicentury, Regional-Scale Patterns of Western Spruce Budworm Outbreaks. *Ecological Monographs* 63, 399–424. <https://doi.org/10.2307/2937153>

---

efk_nh	<i>Baca ponderosa pine chronology</i>
--------	---------------------------------------

---

**Description**

Non-host pair chronology for East Fork Douglas-fir efk\_h

**Usage**

efk\_nh

**Format**

An rwl chronology object with 1 series, 1612-1987. Standardized in ARSTAN.

**Source**

<https://www.ncei.noaa.gov/access/paleo-search/study/5068>

**References**

Swetnam, T.W., Lynch, A.M., 1993. Multicentury, Regional-Scale Patterns of Western Spruce Budworm Outbreaks. *Ecological Monographs* 63, 399–424. <https://doi.org/10.2307/2937153>

---

efk_obr	<i>East Fork outbreak object</i>
---------	----------------------------------

---

**Description**

Produced by running outbreak(efk\_defo1)

**Usage**

efk\_obr

**Format**

An outbreak object with 221 rows and 9 columns

---

<code>get_defol_events</code>	<i>Defoliation event list</i>
-------------------------------	-------------------------------

---

**Description**

Defoliation event list

**Usage**

```
get_defol_events(x)
```

**Arguments**

<code>x</code>	a defol object
----------------	----------------

---

<code>gsi</code>	<i>Calculate the growth suppression index</i>
------------------	---

---

**Description**

This function removes the nonhost growth signal from a host tree-ring series.

**Usage**

```
gsi(input_series)
```

**Arguments**

<code>input_series</code>	A <code>dplr::rwl</code> object with the host tree series as the first column and the non-host chronology as the second. Years should be the row names. This is specifically created by <code>defoliate_trees()</code> and passed to <code>gsi()</code> .
---------------------------	---

**Details**

The growth suppression index (GSI) is referred to as the "corrected" series in OUTBREAK. It is calculated as:

$$GSI(i) = H(i) - (NH(i) - \text{mean}(NH))sd(H)/sd(NH)$$

where H and NH are the host and nonhost tree-ring series as standardized index values; i is the year, and the functions `mean()` and `sd()` are applied to the common period.

`gsi()` will rarely be run directly by a user. It is called from `defoliate_trees()`.

**Value**

A data frame with the two input columns (host and nonhost series) and 3 added columns:

1. The mean/sd adjusted non-host chronology,
2. The growth suppression index ("gsi") of the host series after subtraction of the adjusted non-host chronology,
3. The normalized growth suppression index ("ngsi") generated by applying `scale()` to the gsi.

---

id_defoliation	<i>Identify defoliation events in a host series</i>
----------------	---

---

**Description**

After calculating the growth suppression index in `gsi()`, `id_defoliation()` performs a runs analysis on the normalized growth suppression index, or `ngsi`, in which sequences of negative departures are assessed as possible defoliation events. `id_defoliation()` is the workhorse for `defoliate_trees()`, performing much of the necessary criteria evaluation used in OUTBREAK. The defaults for most parameters originate from OUTBREAK. Two new features distinguish the `dfoliatR` version: bridging events that occur in close sequence and allowing for the recent end of a series to be evaluated for defoliation regardless of the event duration. See parameter specifics for details.

**Usage**

```
id_defoliation(
  input_series,
  duration_years = 8,
  max_reduction = -1.28,
  bridge_events = FALSE,
  series_end_event = FALSE
)
```

**Arguments**

- |                             |  |
|-----------------------------|--|
| <code>input_series</code>   | A data frame with 5 columns, as generated by <code>gsi()</code> .  |
| <code>duration_years</code> | The minimum length of time in which the tree is considered to be in defoliation.   |
| <code>max_reduction</code>  | The minimum value of <code>ngsi</code> required to be considered a defoliation event. If a sequence of negative <code>ngsi</code> values does not reach this threshold, the potential event is rejected. Defaults to -1.28.  |
| <code>bridge_events</code>  | Binary, defaults to FALSE. This option allows for successive events that are separated by a single year to be bridged and become one event. It should be used cautiously and closely evaluated to ensure the likelihood that the two (or more) events are actually one long event. |

series\_end\_event

Binary, defaults to FALSE. This option allows the user to identify an event occurring at the time of sampling as a defoliation event, regardless of duration. Including it will help to quantify periodicity and extent of an outbreak. This should only be used if the user has direct knowledge of an ongoing defoliation event when the trees were sampled.

### Value

After performing runs analyses, the function adds a column to the input data frame that distinguishes years of defoliation and the maximum defoliation year (ie. the year the greatest negative growth departure within the event).

### Note

`id_defoliation()` is called by `defoliate_trees()`. It might only be used by the user for troubleshooting.

---

is.defol

*Check if object is tree-level defoliation object: defol*

---

### Description

Check if object is tree-level defoliation object: defol

### Usage

```
is.defol(x)
```

### Arguments

x                    Any R object.

### Value

Boolean indicating whether x is a defol object.

---

is.obr	<i>Check if object is outbreak, meaning site-level outbreak object</i>
--------	--

---

**Description**

Check if object is outbreak, meaning site-level outbreak object

**Usage**

```
is.obr(x)
```

**Arguments**

x                   Any R object.

**Value**

Boolean indicating whether x is an outbreak object.

---

obr	<i>Constructor for an obr object.</i>
-----	---------------------------------------

---

**Description**

Constructor for an obr object.

**Usage**

```
obr(  
  year,  
  samp_depth,  
  num_defol,  
  perc_defol,  
  num_max_defol,  
  perc_max_defol,  
  mean_gsi,  
  mean_ngsi,  
  outbreak_status  
)
```

**Arguments**

year	An n-length numeric vector of observed years.
samp_depth	An n-length numeric vector of the number of trees.
num_defol	An n-length numeric vector of the number of trees experiencing defoliation.
perc_defol	An n-length numeric vector of the percent of trees experiencing defoliation.
num_max_defol	An n-length numeric vector of the number of trees experiencing their maximum level of defoliation (i.e., their most extreme negative growth departure).
perc_max_defol	An n-length numeric vector of the percent of trees experiencing their maximum level of defoliation (i.e., their most extreme negative growth departure).
mean_gsi	An n-length numeric vector of the average growth suppression index across all observed trees.
mean_ngsi	An n-length numeric vector of the average normalized (scaled) growth suppression index.
outbreak_status	An n-length factor or character vector that identified whether that year surpasses the designated thresholds for an "outbreak event". Threshold criteria are provided in <code>outbreak()</code> .

**Value**

An obr object with columns matching the input variables.

---

outbreak	<i>Composite defoliation series to determine outbreak events</i>
----------	--

---

**Description**

`outbreak()` takes a `defol` object from `defoliate_trees()` and composites it into a site-level object. Function parameters allow the user to filter the tree-level series in various ways to optimize thresholds of what constitutes an "outbreak" level event recorded by the host trees.

**Usage**

```
outbreak(x, filter_perc = 25, filter_min_series = 3, filter_min_defol = 1)
```

**Arguments**

x	a defol object
filter_perc	the minimum percentage of defoliated trees to be considered an outbreak. Default is 25 percent.
filter_min_series	The minimum number of trees required for an outbreak event. Default is 3 trees.
filter_min_defol	The minimum number of trees recording a defoliation event. Default is 1 tree.

**Value**

A data.frame obr object for the site that includes all trees in the host defol object. Columns in the obr include:

1. year for every year in the set of host trees,
2. num\_defol the number of trees recording a defoliation event,
3. percent\_defol the percent of trees recording a defoliation,
4. num\_max\_defol the number of trees recording a maximum growth suppression (or peak of that event on that tree),
5. perc\_max\_defol the percent of trees at maximum defoliation,
6. mean\_gsi the average of all trees growth suppression index (gsi),
7. mean\_ngsi the average of all trees normalized growth suppression index (ngsi),
8. outbreak\_status whether that year constitutes an outbreak based on the filters applied to the function.

**Examples**

```
data("dmj_defol")
head(outbreak(dmj_defol))
```

---

outbreak_stats	<i>Outbreak statistics</i>
----------------	----------------------------

---

**Description**

Summary statistics for inferred outbreaks

**Usage**

```
outbreak_stats(x)
```

**Arguments**

x                    An [obr](#) object after running [outbreak\(\)](#)

**Value**

A data frame with descriptive statistics for each outbreak event determined by [outbreak\(\)](#), including:

- "start" – first year of outbreak
- "end" – last year of outbreak
- "duration" – length of outbreak (in years)
- "n\_df\_start" – number of trees defoliated at the start

- "perc\_df\_start" – percent of trees defoliated at the start
- "max\_df\_obr" – maximum number of trees in the outbreak during a single year
- "yr\_max\_df" – year with the maximum number of trees defoliated
- "yr\_min\_ngsi" – year with the lowest mean normalized growth suppression index (NGSI)
- "min\_gsi" – minimum growth suppression index
- "min\_ngsi" – minimum normalized gsi

### Note

Certain statistics will be set to NA for the final outbreak event if there was an ongoing defoliation event (in which `series_end_event = TRUE` in `defoliate_trees()`). This is because the end of the outbreak remains unknown, so statistics such as duration cannot be calculated. Statistics pertaining to the start of the event are provided.

### Examples

```
data("dmj_obr")
outbreak_stats(dmj_obr)
```

---

plot.defol

*Plot a defol object*

---

### Description

Plot a defol object

### Usage

```
## S3 method for class 'defol'
plot(...)
```

### Arguments

... arguments passed to `plot_defol()`

---

plot_defol	<i>Produce a Gantt plot of individual tree-ring series to show defoliation events in time</i>
------------	---

---

**Description**

Produce a Gantt plot of individual tree-ring series to show defoliation events in time

**Usage**

```
plot_defol(x, breaks)
```

**Arguments**

x	a defol object produced by <code>defoliate_trees()</code> .
breaks	a vector length two providing threshold (negative) ngsi values to separate minor, moderate, and severe defoliation events. If blank, the mean and 1st quartile are used.

**Examples**

```
data("dmj_defol")
plot_defol(dmj_defol)

## Change the values severity classes
plot_defol(dmj_defol, breaks = c(-1.0, -0.5))
```

---

plot_outbreak	<i>Produce a stacked plot to present composited, site-level insect outbreak chronologies</i>
---------------	--

---

**Description**

Produce a stacked plot to present composited, site-level insect outbreak chronologies

**Usage**

```
plot_outbreak(x, disp_index = c("GSI", "NGSI"), label_defol = "% defoliated")
```

**Arguments**

x	an 'obr' object produced by <code>outbreak()</code>
disp_index	Identify the timeseries index to plot. Defaults to NGSI, the average normalized growth suppression index for the site. The only other option is GSI, the average growth suppression index.
label_defol	Allows users to change the bottom tile of the printed output to read what they'd like. Defaults to "% defoliated"

**Examples**

```
data(dmj_obr)
plot_outbreak(dmj_obr)

# Change middle panel display from the default "NGSI" to "GSI"
plot_outbreak(dmj_obr, disp_index = "GSI")
```

---

sample_depth	<i>Calculate the sample depth of a defol object</i>
--------------	---

---

**Description**

Calculate the sample depth of a defol object

**Usage**

```
sample_depth(x)
```

**Arguments**

x                    A defol object.

**Value**

A data.frame containing the years and number of trees

**Examples**

```
data("dmj_defol")
head(sample_depth(dmj_defol))
```

---

series_names	<i>Extract series names from a defol object</i>
--------------	---

---

**Description**

Extract series names from a defol object

**Usage**

```
series_names(x)
```

**Arguments**

x                    a defol object

---

stack\_defoliation      *Stack a defoliation list*

---

**Description**

Stack a defoliation list

**Usage**

stack\_defoliation(x)

**Arguments**

x                      a list object created by [defoliate\\_trees\(\)](#)

**Value**

a defol object (long-format data frame)

# Index

## \* datasets

- dmj\_defol, 8
- dmj\_h, 8
- dmj\_nh, 9
- dmj\_obr, 9
- efk\_defol, 10
- efk\_h, 10
- efk\_nh, 11
- efk\_obr, 11

as.defol, 2

as.obr, 3

as\_defol, 4

as\_defol(), 2

as\_obr, 4

as\_obr(), 3

defol, 5

defol(), 7

defol\_stats, 7

defoliate\_trees, 6

defoliate\_trees(), 6, 7, 12–14, 16, 18, 19, 21

dmj\_defol, 8

dmj\_h, 8

dmj\_nh, 9

dmj\_obr, 9

efk\_defol, 10

efk\_h, 10

efk\_nh, 11

efk\_obr, 11

get\_defol\_events, 12

gsi, 12

gsi(), 5–7, 12, 13

id\_defoliation, 13

id\_defoliation(), 6, 13, 14

is.defol, 14

is.obr, 15

mean(), 12

obr, 15, 17

outbreak, 16

outbreak(), 7, 16, 17, 19

outbreak\_stats, 17

plot.defol, 18

plot\_defol, 19

plot\_defol(), 7, 18

plot\_outbreak, 19

sample\_depth, 20

scale(), 13

sd(), 12

series\_names, 20

stack\_defoliation, 21