

# Package: hakis (via r-universe)

March 6, 2025

**Title** Analysis of Plant Pathogen Pathotype Complexities, Distributions and Diversity

**Version** 4.0.0

**Description** Analysis of plant pathogen pathotype survey data.  
Functions provided calculate distribution of susceptibilities, distribution of complexities with statistics, pathotype frequency distribution, as well as diversity indices for pathotypes. This package is meant to be a direct replacement for Herrmann, Löwer and Schachtel's (1999) [doi:10.1046/j.1365-3059.1999.00325.x](https://doi.org/10.1046/j.1365-3059.1999.00325.x) Habgood-Gilmour Spreadsheet, 'HaGiS', previously used for pathotype analysis.

**License** MIT + file LICENSE

**URL** <https://github.com/openplantpathology/hakis>,  
<https://openplantpathology.github.io/hakis/>

**BugReports** <https://github.com/openplantpathology/hakis/issues>

**Depends** R (>= 3.4.0)

**Imports** data.table, ggplot2, pander, stats, utils

**Suggests** ape, dplyr, knitr, rmarkdown, roxygen2, testthat, vdiff, vegan

**VignetteBuilder** knitr

**Config/Needs/build** moodymudskipper/devtag

**Config/roxygen2/filename** globals.R

**Config/roxygen2/unique** FALSE

**Encoding** UTF-8

**Language** en-US

**LazyData** true

**Roxygen** list(markdown = TRUE, roclets = c("`collate", "`namespace", "`rd", "`roxygen2::global\_roclet", "`devtag::dev\_roclet"))

**RoxygenNote** 7.3.2

**X-schema.org-applicationCategory** Tools

**X-schema.org-isPartOf** <https://openplantpathology.org>  
**X-schema.org-keywords** plant-pathology, pathotype, pathogen-survey,  
virulence analysis, differential set, assessment scale  
**Repository** <https://openplantpathology.r-universe.dev>  
**RemoteUrl** <https://github.com/openplantpathology/hagis>  
**RemoteRef** HEAD  
**RemoteSha** a515edcccd12e312a6b4d7268d9093bb24ceafd6

Contents

autoplot.hagis.complexities . . . . .	2
autoplot.hagis.gene.summary . . . . .	3
calculate_complexities . . . . .	4
calculate_diversities . . . . .	5
create_binary_matrix . . . . .	7
diversities_table . . . . .	8
individual_pathotypes . . . . .	9
P_sojae_survey . . . . .	10
sample_meta . . . . .	11
summarize_gene . . . . .	12
<b>Index</b>	<b>13</b>

---

autoplot.hagis.complexities
<i>Plot hagis Complexities Objects</i>

---

Description

Creates a **ggplot2** object of the frequency of complexity (percent per complexity) or a **ggplot2** object of the distribution (number per complexity) calculated by [calculate\\_complexities\(\)](#).

Usage

```
## S3 method for class 'hagis.complexities'  
autoplot(object, type, color = NULL, order = NULL, ...)
```

Arguments

- object      a **hagis** complexities object generated by [calculate\\_complexities\(\)](#). Character.
- type        a vector of values for which the bar plot is desired. Specify whether to return a graph of the frequency of complexities as a percentage, "percentage", or as the count, "count". Character.
- color       a named or hexadecimal color value to use for the bar color

order	sort the x-axis of the bar chart by ascending or descending order of frequency. Accepts ascending or descending input values. Defaults to complexity value. Character.
...	passed to the chosen geom(s)

## Value

A **ggplot2** object

## Examples

```
# Using the built-in data set, `P_sojae_survey`
data(P_sojae_survey)

# calculate susceptibilities with a 60 % cutoff value
complexities <- calculate_complexities(
  x = P_sojae_survey,
  cutoff = 60,
  control = "susceptible",
  sample = "Isolate",
  gene = "Rps",
  perc_susc = "perc.susc"
)

# Visualize the distribution (count or actual values)
autoplot(complexities, type = "count")

# Visualize the frequency (percentages)
autoplot(complexities, type = "percentage")
```

---

autoplot.hagis.gene.summary

*Plot hagis Summary Objects*

---

## Description

Creates a **ggplot2** object of the gene summaries calculated by `summarize_gene()`

## Usage

```
## S3 method for class 'hagis.gene.summary'
autoplot(object, type, color = NULL, order = NULL, ...)
```

**Arguments**

object	a hagsis.gene.summary object generated by <code>summarize_gene()</code> . Character.
type	a vector of values for which the bar plot is desired. Specify whether to return a graph of the percent pathogenic isolates, percentage, or as the count, count. Character.
color	a named or hexadecimal color value to use for the bar color
order	sort the x-axis of the bar chart by ascending or descending order of <code>N_virulent_isolates</code> or <code>percent_pathogenic</code> . Accepts ascending or descending input values. Defaults to gene name. Character.
...	passed to the chosen <code>geom(s)</code>

**Value**

A **ggplot2** plot

**Examples**

```
# Using the built-in data set, `P-sojae_survey`
data(P-sojae_survey)

# calculate susceptibilities with a 60 % cutoff value
susc <- summarize_gene(
  x = P-sojae_survey,
  cutoff = 60,
  control = "susceptible",
  sample = "Isolate",
  gene = "Rps",
  perc_susc = "perc.susc"
)

# Visualize the summary of genes
autoplot(susc, type = "percentage")
```

---

calculate\_complexities

*Calculate Distribution of Complexities by Sample*

---

**Description**

Calculate the distribution of susceptibilities by sample id.

**Usage**

```
calculate_complexities(x, cutoff, control, sample, gene, perc_susc)
```

**Arguments**

x	a data.frame containing the data.
cutoff	value for percent susceptible cutoff. Numeric.
control	value used to denote the susceptible control in the gene column. Character.
sample	column providing the unique identification for each sample being tested. Character.
gene	column providing the gene(s) being tested. Character.
perc_susc	column providing the percent susceptible reactions. Character.

**Value**

An object of class `hagis.complexities`.

An object of class `hagis.complexities` is a list containing the following components

**grouped\_complexities** a `data.table::data.table()` object of grouped complexities

**individual\_complexities** a `data.table::data.table()` object of individual complexities

**Examples**

```
# Using the built-in data set, `P-sojae_survey`  
data(P-sojae_survey)  
  
P-sojae_survey  
  
# calculate susceptibilities with a 60 % cutoff value  
complexities <- calculate_complexities(  
  x = P-sojae_survey,  
  cutoff = 60,  
  control = "susceptible",  
  sample = "Isolate",  
  gene = "Rps",  
  perc_susc = "perc.susc"  
)  
complexities  
  
summary(complexities)
```

## Description

Calculate five pathogen diversity indices.

Diversity indices include:

- Simple diversity index, which will show the proportion of unique pathotypes to total samples. As the values gets closer to 1, there is greater diversity in pathotypes within the population. Simple diversity is calculated as:

$$D = \frac{Np}{Ns}$$

where  $Np$  is the number of pathotypes and  $Ns$  is the number of samples.

- Gleason diversity index, an alternate version of Simple diversity index, is less sensitive to sample size than the Simple index.

$$D = \frac{(Np - 1)}{\log(Ns)}$$

Where  $Np$  is the number of pathotypes and  $Ns$  is the number of samples.

- Shannon diversity index is typically between 1.5 and 3.5, as richness and evenness of the population increase, so does the Shannon index value.

$$D = - \sum_{i=1}^R p_i \log p_i$$

Where  $p_i$  is the proportional abundance of species  $i$ .

- Simpson diversity index values range from 0 to 1, 1 represents high diversity and 0 represents no diversity. Where diversity is calculated as:

$$D = \sum_{i=1}^R p_i^2$$

- Evenness ranges from 0 to 1, as the Evenness value approaches 1, there is a more even distribution of each pathotype's frequency within the population. Where Evenness is calculated as:

$$D = \frac{H'}{\log(Np)}$$

where  $H'$  is the Shannon diversity index and  $Np$  is the number of pathotypes.

## Usage

```
calculate_diversities(x, cutoff, control, sample, gene, perc_susc)
```

## Arguments

<code>x</code>	a <code>data.frame</code> containing the data.
<code>cutoff</code>	value for percent susceptible cutoff. Numeric.
<code>control</code>	value used to denote the susceptible control in the gene column. Character.
<code>sample</code>	column providing the unique identification for each sample being tested. Character.
<code>gene</code>	column providing the gene(s) being tested. Character.
<code>perc_susc</code>	column providing the percent susceptible reactions. Character.

**Value**

A hgis.diversities object.

A hgis.diversities object is a list containing:

- Number of Samples
- Number of Pathotypes
- Simple Diversity Index
- Gleason Diversity Index
- Shannon Diversity Index
- Simpson Diversity Index
- Evenness Diversity Index

**Examples**

```
# Using the built-in data set, P_sojae_survey
data(P_sojae_survey)

P_sojae_survey

# calculate susceptibilities with a 60 % cutoff value
diversities <- calculate_diversities(
  x = P_sojae_survey,
  cutoff = 60,
  control = "susceptible",
  sample = "Isolate",
  gene = "Rps",
  perc_susc = "perc.susc"
)

diversities
```

---

create\_binary\_matrix    *Create Binary Data Matrix From Pathotype Data*

---

**Description**

Creates a binary data matrix from pathotype data representing the pathotype of each isolate. This binary data matrix can be used to visualize beta-diversity of pathotypes using **vegan** and **ape**.

**Usage**

```
create_binary_matrix(x, cutoff, control, sample, gene, perc_susc)
```

**Arguments**

<code>x</code>	a <code>data.frame</code> containing the data.
<code>cutoff</code>	value for percent susceptible cutoff. Numeric.
<code>control</code>	value used to denote the susceptible control in the gene column. Character.
<code>sample</code>	column providing the unique identification for each sample being tested. Character.
<code>gene</code>	column providing the gene(s) being tested. Character.
<code>perc_susc</code>	column providing the percent susceptible reactions. Character.

**Value**

a binary matrix of pathotype data

**Examples**

```
# Using the built-in data set, `P_sojae_survey`
data(P_sojae_survey)

P_sojae_survey

# calculate susceptibilities with a 60 % cutoff value
final_matrix <- create_binary_matrix(x = P_sojae_survey,
                                     cutoff = 60,
                                     control = "susceptible",
                                     sample = "Isolate",
                                     gene = "Rps",
                                     perc_susc = "perc.susc")

final_matrix
```

---

<code>diversities_table</code>	<i>Custom Print for hagsis Diversities Tables</i>
--------------------------------	---

---

**Description**

Print the frequency table of diversities from a `hagsis.diversities` object The resulting object is a [pander](#) table (a text object for Markdown) for ease of use in reporting and viewing in the console.

**Usage**

```
diversities_table(x, ...)
```

**Arguments**

<code>x</code>	a <code>hagsis.diversities</code> object generated by <a href="#">calculate_diversities()</a>
<code>...</code>	other arguments passed to <a href="#">pander::panderOptions()</a>



**Value**

A [pander](#) object of diversities

**See Also**

[calculate\\_diversities\(\)](#), [individual\\_pathotypes\(\)](#)

**Examples**

```
# Using the built-in data set, P_sojae_survey
data(P_sojae_survey)

P_sojae_survey

# calculate susceptibilities with a 60 % cutoff value
diversities <- calculate_diversities(
  x = P_sojae_survey,
  cutoff = 60,
  control = "susceptible",
  sample = "Isolate",
  gene = "Rps",
  perc_susc = "perc.susc"
)

# print the diversities table
diversities_table(diversities)
```

---

`individual_pathotypes` *Prints Individual Pathotypes for Each Sample*

---

**Description**

Print an object from a `hagis.diversities` object with individual pathotypes, *i.e.* each sample's pathotype. The resulting object is a [pander](#) table (a text object for Markdown) for ease of use in reporting and viewing in the console.

**Usage**

```
individual_pathotypes(x, ...)
```

**Arguments**

`x` a `hagis.diversities` object generated by [calculate\\_diversities\(\)](#)  
`...` other arguments passed to [pander::panderOptions\(\)](#)

**Value**

A [pander](#) object of individual pathotypes

See Also

```
calculate_diversities(), diversities_table()
```

Examples

```
# Using the built-in data set, P_sojae_survey
data(P_sojae_survey)

P_sojae_survey

# calculate susceptibilities with a 60 % cutoff value
diversities <- calculate_diversities(
  x = P_sojae_survey,
  cutoff = 60,
  control = "susceptible",
  sample = "Isolate",
  gene = "Rps",
  perc_susc = "perc.susc"
)

# print the diversities table
individual_pathotypes(diversities)
```

---

P_sojae_survey	Phytophthora sojae Survey Example Data
----------------	--

---

Description

Data from a *Phytophthora sojae* survey

Usage

```
data(P_sojae_survey)
```

Format

An object of class `data.table` with 294 observations of 12 variables

- Isolate** *P. sojae* isolate identifier
- Line** Soybean cultivar
- Rps** *Rps* gene identifier
- Total** Total number of plants inoculated
- HR (1)** Number of plants that exhibit a hypersensitive response after inoculation
- Lesion (2)** Number of plants that develop a lesion at inoculation site
- Lesion to cotyledon (3)** Number of plants that develop a lesion, which advances to the hypocotyl of the seedling after infection

**Dead (4)** Number of dead plants that are observed after inoculation

**total.susc** The total number of susceptible plants (Lesion+Lesion to cotyledon+Dead)

**total.resis** The total number of resistant plants (equal to HR value)

### Source

Data from an ongoing 2017 *Phytophthora sojae* survey in Michigan, conducted by A. G. McCoy *et al.*.

### Examples

```
data(P_sojae_survey)
P_sojae_survey
```

---

sample_meta	Phytophthora sojae Survey Example Metadata
-------------	--

---

### Description

Metadata to accompany internal data from a *Phytophthora sojae* survey used for an example in the "Beta-diversity Analyses" vignette.

### Usage

```
data(sample_meta)
```

### Format

An object of class `data.frame` with 21 observations of 2 variables

**Sample** *P. sojae* sample identifier

**Locale** US State where sample was collected, limited to Michigan

### Source

Data from an ongoing 2017 *Phytophthora sojae* survey in Michigan, conducted by A. G. McCoy *et al.*.

### Examples

```
data(sample_meta)
sample_meta
```

---

summarize\_gene

---

*Calculate and Summarize Distribution of Susceptibilities by Gene*


---

## Description

Calculate the distribution of susceptibilities by gene.

## Usage

```
summarize_gene(x, cutoff, control, sample, gene, perc_susc)
```

## Arguments

x	a data.frame containing the data.
cutoff	value for percent susceptible cutoff. Numeric.
control	value used to denote the susceptible control in the gene column. Character.
sample	column providing the unique identification for each sample being tested. Character.
gene	column providing the gene(s) being tested. Character.
perc_susc	column providing the percent susceptible reactions. Character.

## Value

a hgis.gene.summary object.

An object of class hgis.gene.summary is a

[data.table::data.table\(\)](#) containing the following components columns

**gene** the gene

**N\_virulent\_isolates** the total number virulent isolates for a given gene in the gene column

**percent\_pathogenic** the frequency with which a gene is pathogenic

## Examples

```
# Using the built-in data set, `P_sojae_survey`
data(P_sojae_survey)

P_sojae_survey

# calculate susceptibilities with a 60 % cutoff value
susc <- summarize_gene(
  x = P_sojae_survey,
  cutoff = 60,
  control = "susceptible",
  sample = "Isolate",
  gene = "Rps",
  perc_susc = "perc.susc"
)
susc
```

# Index

## \* datasets

P\_sojae\_survey, [10](#)

sample\_meta, [11](#)

autoplot.hagis.complexities, [2](#)

autoplot.hagis.gene.summary, [3](#)

calculate\_complexities, [4](#)

calculate\_complexities(), [2](#)

calculate\_diversities, [5](#)

calculate\_diversities(), [8–10](#)

create\_binary\_matrix, [7](#)

data.table::data.table(), [5](#), [12](#)

diversities\_table, [8](#)

diversities\_table(), [10](#)

individual\_pathotypes, [9](#)

individual\_pathotypes(), [9](#)

P\_sojae\_survey, [10](#)

pander, [8](#), [9](#)

pander::panderOptions(), [8](#), [9](#)

sample\_meta, [11](#)

summarize\_gene, [12](#)

summarize\_gene(), [3](#), [4](#)