Package: phenop (via r-universe)

May 17, 2026

Title Multidimensional Analysis of Phenotypic Plasticity in Eco-Evolutionary Contexts

Version 0.1.0

Description Provides novel methods for analyzing phenotypic plasticity across multiple traits and environmental variables, with special focus on insect-fungus host-pathogen systems. Implements original multidimensional plasticity indices and interaction analyses not available in other packages.

License MIT + file LICENSE

URL `https://github.com/leonelstazione/phenop`

BugReports `https://github.com/leonelstazione/phenop/issues`

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

VignetteBuilder knitr

LazyDataCompression xz

Depends R (>= 4.0.0)

Imports dplyr, ggplot2, tidyr, lme4, mgcv, performance, FactoMineR, cluster, patchwork, magrittr, stats

Suggests testthat (>= 3.0.0), knitr, rmarkdown, kableExtra, akima, devtools, MASS, ggridges, plotly, maps, viridis, sf

Config/testthat/edition 3

Config/pak/sysreqs cmake make libicu-dev libuv1-dev

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

Date/Publication 2026-01-27 11:41:06 UTC

RemoteUrl https://github.com/leonelstazione/phenop

RemoteRef HEAD

RemoteSha 1a3371d09fb2974a4402f91f1576c1f38c342a6f

Contents

`analyze_correlations` *Analyze Correlations Between Phenological Parameters*

Description

Calculates correlation matrix between key phenological variables.

Usage

```
analyze_correlations(parameters)
```

Arguments

`parameters`	A data frame containing extracted phenological parameters

Value

A list with correlation results

Examples

```
## Not run:
data(pheno_parameters)
corr_results <- analyze_correlations(pheno_parameters)

## End(Not run)
```

`analyze_gam_relationships` *Analyze GAM relationships*

Description

Fits Generalized Additive Models to capture non-linear relationships.

Usage

```
analyze_gam_relationships(time_series_subset)
```

Arguments

`time_series_subset`	A subset of the time series data

Value

A GAM model object

Examples

```
## Not run:
data(pheno_time_series)
subset <- pheno_time_series[pheno_time_series$species == "Quercus_ilex", ]
gam_model <- analyze_gam_relationships(subset)

## End(Not run)
```

`analyze_mixed_effects` *Analyze Phenology with Mixed Effects Models*

Description

Fits mixed effects models to analyze phenological timing.

Usage

```
analyze_mixed_effects(parameters)
```

Arguments

`parameters`	A data frame containing extracted phenological parameters

Value

A list with model results

Examples

```
## Not run:
data(pheno_parameters)
results <- analyze_mixed_effects(pheno_parameters)

## End(Not run)
```

`analyze_temporal_trends` *Analyze Temporal Trends in Phenology*

Description

Calculates linear trends in phenological parameters over years.

Usage

```
analyze_temporal_trends(parameters)
```

Arguments

`parameters`	A data frame containing extracted phenological parameters

Value

A data frame with trend statistics

Examples

```
## Not run:
data(pheno_parameters)
trends <- analyze_temporal_trends(pheno_parameters)

## End(Not run)
```

`anova_plasticity` *G×E ANOVA for Phenotypic Plasticity Analysis*

Description

Performs ANOVA to test genotype-environment interactions and phenotypic plasticity significance.

Usage

```
anova_plasticity(formula, data, return_type = "anova")
```

Arguments

`formula`	A formula specifying the model (e.g., trait ~ genotype * environment)
`data`	A data frame containing the variables
`return_type`	Type of output: "anova" (default), "model", or "summary"

Value

ANOVA table, model object, or summary depending on return_type

Examples

```
# Using the example data from calcular_ip
example_data <- data.frame(
  genotype = rep(LETTERS[1:5], each = 4),
  environment = rep(c("Low", "High"), each = 2, times = 5),
  trait_value = rnorm(20, 10, 2)
)

anova_plasticity(trait_value ~ genotype * environment, example_data)
```

`create_phenology_dashboard` *Create Phenology Dashboard*

Description

Generates a comprehensive dashboard with multiple phenological visualizations.

Usage

```
create_phenology_dashboard(
  time_series,
  parameters,
  spatial_data,
  selected_year = 2021
)
```

Arguments

`time_series`	Time series data
`parameters`	Extracted parameters data
`spatial_data`	Spatial data
`selected_year`	Year for seasonal curves (default: 2021)

Value

A patchwork object combining multiple plots

`datos_demo` *Datos de demonstration para plasticity analysis*

Description

Dataset simulated que contiene medidas fenotípicas de interacciones insecto-hongo

Usage

```
datos_demo
```

Format

Un data.frame con 200 observaciones y 8 variables:

insect_population Población de insecto (P1, P2, P3, P4)

fungus_strain Cepa de hongo (C1, C2, C3)

temperature Temperatura en °C

humidity Humedad en %

insect_size Tamaño del insecto en mm

resistance Resistencia del insecto

fungus_growth Crecimiento del hongo

virulence Virulencia del hongo

Source

Simulated data para el paquete phenop

Examples

```
data(datos_demo)
head(datos_demo)
```

`demo_data` *Demo Dataset for Plasticity Analysis*

Description

Simulated dataset containing insect and fungus data to demonstrate phenop package functionalities.

Usage

```
demo_data
```

Format

A data.frame with 200 rows and 8 columns:

insect_population Insect population (P1, P2, P3)

fungus_strain Fungus strain (C1, C2, C3)

temperature Temperature in Celsius (15-30)

humidity Relative humidity in percentage (40-80)

insect_size Insect size in mm (mean 10, sd 2)

resistance Insect resistance to infection (mean 5, sd 1)

fungus_growth Fungus growth rate (mean 8, sd 1.5)

virulence Fungus virulence (mean 6, sd 1)

Source

Simulated data for demonstration purposes

Examples

```
data(demo_data)
head(demo_data)
```

`detect_anomalies` *Detect Anomalies in Phenological Parameters*

Description

Identifies anomalous observations based on z-scores and IQR methods.

Usage

```
detect_anomalies(parameters, threshold = 2.5)
```

Arguments

`parameters`	A data frame containing extracted phenological parameters. Must include columns: species, site, sos, eos, los.
`threshold`	Numeric threshold for z-score anomaly detection (default = 2.5). Observations with z-scores beyond ±threshold are flagged as anomalies.

Value

A data frame with anomaly flags including:

`sos_anomaly`	Logical flag for start-of-season anomalies
`eos_anomaly`	Logical flag for end-of-season anomalies
`los_anomaly`	Logical flag for length-of-season anomalies
`anomaly_score`	Numeric score (0-3) of total anomalies detected
`sos_z`, `eos_z`, `los_z`	Z-scores for each parameter

Examples

```
## Not run:
data(pheno_parameters)
anomalies <- detect_anomalies(pheno_parameters)
anomalies_strict <- detect_anomalies(pheno_parameters, threshold = 3.0)

## End(Not run)
```

`fungi_insect_data` *Fungi-Insect Interaction Dataset*

Description

Simulated dataset of phenotypic plasticity in a fungus-insect host-pathogen system. Contains data for 20 fungus strains across 10 insect populations under different temperature and humidity conditions.

Usage

```
fungi_insect_data
```

Format

A data frame with 600 rows and 8 variables:

fungus_strain Fungus strain identifier (H1-H20)

insect_population Insect population identifier (I1-I10)

temperature Environmental temperature in °C (15-35)

humidity Relative humidity in percent (30, 60, 90)

fungus_growth Fungus growth rate

virulence Fungus virulence (0-1 scale)

insect_size Insect body size

resistance Insect resistance to fungus (0-1 scale)

Source

Simulated data for package examples and testing

`host_pathogen_interaction` *Host-Pathogen Interaction Analysis*

Description

Calculates basic plasticity metrics for host-pathogen comparisons. Robust alternative that calculates plasticity metrics without errors.

Usage

```
host_pathogen_interaction(
  data,
  host_trait,
  pathogen_trait,
  environment,
  host_group = "insect_population",
  pathogen_group = "fungus_strain"
)
```

Arguments

`data`	Data frame with both host and pathogen data
`host_trait`	Host trait name
`pathogen_trait`	Pathogen trait name
`environment`	Environmental variable name
`host_group`	Column name for host groups
`pathogen_group`	Column name for pathogen groups

Value

List with plasticity comparison results

Examples

```
data(fungi_insect_data)
result <- host_pathogen_interaction(
  data = fungi_insect_data,
  host_trait = "insect_size",
  pathogen_trait = "fungus_growth",
  environment = "temperature",
  host_group = "insect_population",
  pathogen_group = "fungus_strain"
)
```

`host_pathogen_interaction_extended`
Extended Host-Pathogen Interaction Analysis

Description

Comprehensive analysis of phenotypic plasticity in host-pathogen systems.

Usage

```
host_pathogen_interaction_extended(
  host_data,
  pathogen_data,
  host_traits,
  pathogen_traits,
  interaction_metrics = c("correlation"),
  environments,
  host_group = NULL,
  pathogen_group = NULL,
  infection_data = NULL,
  ...
)
```

Arguments

`host_data`	Data frame with host phenotypic data
`pathogen_data`	Data frame with pathogen phenotypic data
`host_traits`	Character vector of host trait names
`pathogen_traits`	Character vector of pathogen trait names
`interaction_metrics`	Metrics to calculate
`environments`	Environmental variables
`host_group`	Column name for host groups
`pathogen_group`	Column name for pathogen groups
`infection_data`	Optional infection data
`...`	Additional arguments

`multidim_plasticity` *Calculate Multidimensional Plasticity Index (MPI)*

Description

Calculate Multidimensional Plasticity Index (MPI)

Usage

```
multidim_plasticity(
  data,
  traits,
  environments,
  groups,
  weights = NULL,
```

```
    na.action = "omit",
    min_variation = 0.01,
    ...
  )
```

Arguments

`data`	A data frame containing the phenotypic data
`traits`	Character vector of trait column names
`environments`	Character string of environment column name
`groups`	Character string of group/genotype column name
`weights`	Optional numeric vector of weights for each trait
`na.action`	How to handle NAs ("omit", "fail", "warning")
`min_variation`	Minimum variation threshold (default 0.01)
`...`	Additional arguments passed to internal functions

Value

A list with multidimensional_index and individual_plasticity

Examples

```
data(demo_data)
result <- multidim_plasticity(
  data = demo_data,
  traits = c("insect_size", "resistance"),
  environments = "temperature",
  groups = "insect_population"
)
```

---

optimize_multidim_environment

Multidimensional Environmental Optimization

Description

Finds optimal environmental conditions considering multiple traits simultaneously using original optimization algorithms for phenotypic plasticity landscapes.

Usage

```
optimize_multidim_environment(
  data,
  traits,
  environments,
  groups,
```

```
  optimization_goal = "compromise",
  weights = NULL
)
```

Arguments

`data`	Data frame with phenotypic and environmental data
`traits`	Vector of trait names to optimize
`environments`	Vector of environmental variables to optimize
`groups`	Grouping variable(genotypes/populations)
`optimization_goal`	Optimization goal: "maximize", "minimize", "compromise"
`weights`	Optional weights for each trait's importance

Value

List with optimal conditions and optimization landscape

`perform_cluster_analysis`

Perform Cluster Analysis on Phenological Types

Description

Groups populations based on phenological characteristics using k-means.

Usage

```
perform_cluster_analysis(parameters, n_clusters = 4)
```

Arguments

`parameters`	A data frame containing extracted phenological parameters
`n_clusters`	Number of clusters (default: 4)

Value

A list with clustering results

Examples

```
## Not run:
data(pheno_parameters)
clusters <- perform_cluster_analysis(pheno_parameters, n_clusters = 3)

## End(Not run)
```

`perform_pca_analysis` *Perform Principal Component Analysis*

Description

Performs PCA on phenological parameters to identify main axes of variation.

Usage

```
perform_pca_analysis(parameters)
```

Arguments

`parameters`	A data frame containing extracted phenological parameters

Value

A PCA result object

Examples

```
## Not run:
data(pheno_parameters)
pca_results <- perform_pca_analysis(pheno_parameters)

## End(Not run)
```

`pheno_parameters` *Extracted Phenological Parameters*

Description

Derived phenological metrics extracted from time series analysis.

Usage

```
pheno_parameters
```

Format

A data frame with X rows and 44 variables:

parameter_id Unique identifier for each parameter record

site Site name or identifier where observations were made

species Species name or identifier

year Year of observation

population_id Population identifier for the observed group
n_observations Number of individual observations aggregated
sos Start of season (day of year when growth begins)
eos End of season (day of year when growth ends)
pos Peak of season (day of year of maximum growth)
los Length of season (number of days between SOS and EOS)
ndvi_max Maximum Normalized Difference Vegetation Index value
ndvi_min Minimum Normalized Difference Vegetation Index value
ndvi_amplitude Difference between maximum and minimum NDVI
ndvi_mean Mean Normalized Difference Vegetation Index value
ndvi_sd Standard deviation of NDVI values
ndvi_cv Coefficient of variation of NDVI values
ndvi_integral Cumulative NDVI over the growing season
gpp_total Total Gross Primary Production (cumulative)
gpp_max Maximum daily Gross Primary Production
lai_max Maximum Leaf Area Index value
greenup_rate Rate of green-up (rate of increase in vegetation)
senescence_rate Rate of senescence (rate of vegetation decline)
mean_temperature Mean temperature during the growing season (°C)
total_precipitation Total precipitation during growing season (mm)
mean_gdd Mean Growing Degree Days
total_gdd Total Growing Degree Days accumulated
budburst_doy Day of year when budburst occurred
flowering_doy Day of year when flowering occurred
senescence_doy Day of year when senescence began
data_completeness Percentage of complete data records
elevation Elevation above sea level (meters)
latitude Latitude coordinate in decimal degrees
longitude Longitude coordinate in decimal degrees
soil_type Classification of soil type
soil_pH Soil pH measurement
slope Terrain slope in degrees
phenology_type Type of phenology (e.g., early, late, etc.)
productivity_class Productivity classification (low, medium, high)
sos_anomaly Deviation of SOS from long-term average
eos_anomaly Deviation of EOS from long-term average
los_anomaly Deviation of LOS from long-term average
water_use_efficiency Water use efficiency metric
temperature_efficiency Temperature use efficiency metric
sos_trend Temporal trend in SOS over years
eos_trend Temporal trend in EOS over years

`pheno_spatial` *Spatial Phenological Data*

Description

Gridded spatial data for phenological mapping and spatial analysis.

Usage

```
pheno_spatial
```

Format

A data frame with X rows and 24 variables:

site_id	Unique identifier for the spatial site
longitude	Longitude coordinate in decimal degrees
latitude	Latitude coordinate in decimal degrees
elevation	Elevation above sea level (meters)
mean_annual_temp	Mean annual temperature (°C)
annual_precipitation	Total annual precipitation (mm)
soil_type	Classification of soil type
soil_pH	Soil pH measurement
soil_organic_carbon	Soil organic carbon content (g/kg)
sos	Start of season (day of year)
eos	End of season (day of year)
los	Length of season (days)
ndvi_max	Maximum Normalized Difference Vegetation Index value
land_cover	Land cover classification type
forest_type	Forest type classification
protected_area	Indicates if the site is in a protected area
years_since_disturbance	Years since last disturbance event
disturbance_type	Type of disturbance (fire, logging, etc.)
climate_zone	Climate zone classification
phenology_class	Phenology classification category
productivity	Productivity metric value
productivity_class	Productivity classification category
species_richness	Number of species recorded at the site
aboveground_carbon	Aboveground carbon stock (Mg/ha)

`pheno_spatial_sf` *Phenological Spatial Data as Simple Features*

Description

A spatial dataset containing phenological observations with geometry.

Usage

```
pheno_spatial_sf
```

Format

An sf object with 150 features and 12 fields.

site_id Unique identifier for the spatial site

longitude Longitude coordinate in decimal degrees

latitude Latitude coordinate in decimal degrees

elevation Elevation above sea level (meters)

mean_annual_temp Mean annual temperature (°C)

annual_precipitation Total annual precipitation (mm)

soil_type Classification of soil type

soil_pH Soil pH measurement

soil_organic_carbon Soil organic carbon content (g/kg)

sos Start of season (day of year)

eos End of season (day of year)

los Length of season (days)

ndvi_max Maximum Normalized Difference Vegetation Index value

land_cover Land cover classification type

forest_type Forest type classification

protected_area Indicates if the site is in a protected area

years_since_disturbance Years since last disturbance event

disturbance_type Type of disturbance (fire, logging, etc.)

climate_zone Climate zone classification

phenology_class Phenology classification category

productivity Productivity metric value

productivity_class Productivity classification category

species_richness Number of species recorded at the site

aboveground_carbon Aboveground carbon stock (Mg/ha)

`pheno_time_series` *Phenological Time Series Dataset*

Description

A comprehensive synthetic dataset of weekly phenological observations for 15 populations, 5 species, 7 sites, and 6 years (2018-2023).

Usage

```
pheno_time_series
```

Format

A data frame with X rows and 29 variables:

observation_id Unique identifier for each observation

population_id Population identifier

species Species name or identifier

site Site name or identifier

year Year of observation

date Date of observation (YYYY-MM-DD)

doy Day of year (1-365/366)

month Month of observation (1-12)

season Seasonal classification

latitude Latitude coordinate in decimal degrees

longitude Longitude coordinate in decimal degrees

elevation Elevation above sea level (meters)

soil_type Classification of soil type

soil_pH Soil pH measurement

slope Terrain slope in degrees

ndvi Normalized Difference Vegetation Index value

evi Enhanced Vegetation Index value

gpp Gross Primary Production value

lai Leaf Area Index value

temperature Temperature measurement (°C)

precipitation Precipitation measurement (mm)

gdd Growing Degree Days accumulated

vegetation_health Vegetation health index value

sos_adj Adjusted start of season indicator

eos_adj Adjusted end of season indicator

budburst Budburst phenophase indicator

flowering Flowering phenophase indicator

senescence Senescence phenophase indicator

quality_flag Data quality flag

Source

Generated synthetically for the phenop package

Examples

```
data(pheno_time_series)
summary(pheno_time_series$ndvi)
ggplot2::ggplot(pheno_time_series, ggplot2::aes(x = doy, y = ndvi)) +
  ggplot2::geom_line(alpha = 0.3)
```

`plasticidad_ejemplo` *Example dataset para plasticidad fenotipica*

Description

Simulated data to demonstrate las funciones of the package phenop.

Format

Un data.frame con 200 observaciones y 6 variables.

Source

Simulado

`plasticity_meta_analysis` *Plasticity Meta-Analysis*

Description

Performs meta-analysis of phenotypic plasticity studies using original methods for synthesizing effect sizes and exploring publication bias.

Usage

```
plasticity_meta_analysis(
  study_data,
  effect_size = "effect_size",
  variance = "variance",
  moderator = NULL,
  method = "random"
)
```

Arguments

`study_data`	Data frame with study results (plasticity effect sizes)
`effect_size`	Column name containing effect sizes
`variance`	Column name containing variances
`moderator`	Optional moderator variables for analysis
`method`	Meta-analysis method: "fixed", "random", "multilevel"

Value

Meta-analysis results with original plasticity-specific metrics

---

`plasticity_meta_analysis_extended`
Extended Plasticity Meta-Analysis

Description

Comprehensive meta-analysis of phenotypic plasticity studies. This function synthesizes effect sizes from multiple studies, assesses heterogeneity, publication bias, and moderating effects.

Usage

```
plasticity_meta_analysis_extended(
  study_data,
  effect_size = "effect_size",
  variance = "variance",
  study_labels = "study",
  method = "random",
  p_value_threshold = 0.05,
  moderator = NULL,
  ...
)
```

Arguments

`study_data`	Data frame containing study results. Must include columns for effect sizes and their variances.
`effect_size`	Character string specifying the column name for effect sizes
`variance`	Character string specifying the column name for variances
`study_labels`	Character string specifying the column name for study identifiers
`method`	Meta-analysis method: "fixed" (default), "random", "bayesian", "multilevel"
`p_value_threshold`	Threshold for statistical significance (default: 0.05)
`moderator`	Optional character vector of moderator variable names
`...`	Additional arguments passed to meta-analysis functions

Value

A list containing:

`meta_results`	Overall meta-analysis results
`individual_studies`	Study-level statistics
`heterogeneity`	Heterogeneity analysis
`publication_bias`	Publication bias assessment
`moderator_analysis`	Results of moderator analysis (if requested)
`forest_plot_data`	Data for creating forest plots

Examples

```
# Create simulated meta-analysis data
n_studies <- 15
meta_data <- data.frame(
  study = paste("Study", 1:n_studies),
  effect_size = rnorm(n_studies, mean = 0.6, sd = 0.3),
  variance = runif(n_studies, 0.05, 0.2),
  year = 2000 + 1:n_studies,
  taxon = sample(c("Insect", "Fungus", "Plant"), n_studies, replace = TRUE),
  trait_type = sample(c("Morphological", "Physiological", "Life History"),
                     n_studies, replace = TRUE)
)

# Run meta-analysis
meta_result <- plasticity_meta_analysis_extended(
  study_data = meta_data,
  effect_size = "effect_size",
  variance = "variance",
  study_labels = "study",
```

```
  method = "random",
  moderator = c("year", "taxon", "trait_type")
)

print(meta_result)
```

`plasticity_tradeoffs` *Plasticity Trade-off Analysis*

Description

Identifies and quantifies trade-offs between plasticities of different traits using original correlation-based and constraint-based methods.

Usage

```
plasticity_tradeoffs(multidim_result, method = "correlation", threshold = -0.5)
```

Arguments

`multidim_result`	Result from multidim_plasticity() function
`method`	Trade-off detection method: "correlation", "pca", "constraint"
`threshold`	Threshold for significant trade-offs

Value

List with trade-off metrics and visualizations

`plasticity_tradeoffs_extended` *Extended Plasticity Trade-off Analysis*

Description

Comprehensive analysis of trade-offs between different phenotypic plasticity indices. This function identifies and quantifies constraints and trade-offs in multivariate plasticity space.

Usage

```
plasticity_tradeoffs_extended(
  plasticity_result,
  method = "correlation",
  threshold = -0.5,
  p_value_threshold = 0.05,
  visualization = TRUE,
  ...
)
```

Arguments

`plasticity_result`	Result object from multidim_plasticity() or safe_multidim_plasticity()
`method`	Analysis method: "correlation" (default), "pca", "constraint", "regression"
`threshold`	Correlation threshold for identifying trade-offs (default: -0.5)
`p_value_threshold`	Significance threshold for statistical tests (default: 0.05)
`visualization`	Logical, whether to prepare data for visualization (default: TRUE)
`...`	Additional arguments passed to analysis functions

Value

A list containing:

`tradeoff_matrix`	Matrix of correlations between plasticity indices
`significant_tradeoffs`	Data frame of significant trade-off pairs
`pca_results`	PCA results if method includes PCA
`constraint_indices`	Constraint strength indices
`visualization_data`	Data prepared for plotting
`analysis_parameters`	Parameters used in the analysis

Examples

```
data(demo_data)

# First run plasticity analysis
insect_data <- demo_data %>%
  dplyr::select(insect_population, temperature, humidity,
                insect_size, resistance) %>%
  dplyr::distinct()

result <- safe_multidim_plasticity(
  insect_data,
  traits = c("insect_size", "resistance"),
  environments = c("temperature", "humidity"),
  groups = "insect_population"
)

# Then analyze trade-offs
tradeoffs <- plasticity_tradeoffs_extended(result, method = "correlation")
print(tradeoffs)
```

`plot_density_ridges` *Plot Density Ridges of Phenological Parameters*

Description

Creates density ridge plots showing distributions of phenological parameters.

Usage

```
plot_density_ridges(
  parameters,
  parameter = "los",
  group_by = "species",
  title = NULL
)
```

Arguments

`parameters`	A data frame containing extracted phenological parameters
`parameter`	Name of parameter to plot (default: "los")
`group_by`	Grouping variable (default: "species")
`title`	Plot title (optional)

Value

A ggplot object

`plot_internals` *Internal plotting function for multidimensional plasticity*

Description

Internal plotting function for multidimensional plasticity

Usage

```
plot_internals(df, type = "integration", traits = NULL, plot_options = list())
```

Arguments

`df`	Data frame or result object
`type`	Type of plot: "integration" or "radar"
`traits`	Optional vector of traits
`plot_options`	List of plot options

Value

ggplot object

`plot_multidim_plasticity` *Multidimensional Plasticity Visualization*

Description

Creates innovative visualizations for multidimensional plasticity analysis including correlation networks, plasticity landscapes, and trait integration plots.

Usage

```
plot_multidim_plasticity(
  multidim_result,
  type = "network",
  traits = NULL,
  plot_options = list()
)
```

Arguments

`multidim_result`	Result from multidim_plasticity() function
`type`	Type of plot: "network", "landscape", "integration", "radar"
`traits`	Optional: subset of traits to display
`plot_options`	Additional plotting options

Value

ggplot object or list of plots

Examples

```
# Usar demo_data en lugar de fungi_insect_data
data(demo_data)
# Preparar datos para multidim_plasticity
result <- safe_multidim_plasticity(
  demo_data,
  traits = c("insect_size", "resistance"),
  environments = c("temperature", "humidity"),
  groups = "insect_population"
)
plot_multidim_plasticity(result, "network")
```

`plot_multidim_plasticity_extended` *Extended Multidimensional Plasticity Visualization (SAFE)*

Description

Advanced plotting functions for visualizing multidimensional plasticity results. Always returns a ggplot object, even if data are missing or incomplete.

Usage

```
plot_multidim_plasticity_extended(
  multidim_result,
  type = "heatmap",
  traits = NULL,
  plot_options = list(),
  ...
)
```

Arguments

`multidim_result`	Result object from multidim_plasticity() or safe_multidim_plasticity()
`type`	Type of plot to create. Options: "heatmap" (default), "network", "radar", "bar", "scatter", "violin", "density", "pca", "landscape", "integration"
`traits`	Optional character vector of traits to include (subset of all traits)
`plot_options`	List of additional plotting options: • title: Plot title • color_palette: Color palette name ("viridis", "plasma", "magma", "inferno", "cividis") • theme: ggplot2 theme ("minimal", "classic", "bw", "dark", "light") • interactive: Logical, whether to create interactive plot (requires plotly) • save_plot: Logical, whether to save plot to file • filename: Filename for saved plot • width, height: Plot dimensions in inches • alpha: transparency • point_size, text_size, legend_position
`...`	Additional arguments passed to plotting functions

Value

A ggplot object (or plotly object if interactive = TRUE)

`plot_parameter_heatmap` *Plot Phenological Parameter Heatmap*

Description

Creates a heatmap showing variation in phenological parameters across species, sites, and years.

Usage

```
plot_parameter_heatmap(
  parameters,
  parameter = "los",
  title = "Phenological Parameter Variation"
)
```

Arguments

`parameters`	A data frame containing extracted phenological parameters
`parameter`	Name of parameter to plot (default: "los")
`title`	Plot title (optional)

Value

A ggplot object

`plot_productivity_climate` *Plot Productivity vs Climate Relationships*

Description

Shows relationships between productivity (GPP) and climate variables.

Usage

```
plot_productivity_climate(parameters)
```

Arguments

`parameters`	A data frame containing extracted phenological parameters

Value

A ggplot object

`plot_reaction_norm` *Plot Reaction Norms for Phenotypic Plasticity*

Description

Creates reaction norm plots to visualize genotype-environment interactions.

Usage

```
plot_reaction_norm(
  data,
  environment,
  trait,
  genotype,
  add_points = TRUE,
  add_se = TRUE
)
```

Arguments

`data`	A data frame containing the phenotypic data
`environment`	Column name for environment variable(unquoted)
`trait`	Column name for phenotypic trait (unquoted)
`genotype`	Column name for genotype grouping (unquoted)
`add_points`	Whether to add individual data points (default: TRUE)
`add_se`	Whether to add standard error bars (default: TRUE)

Value

A ggplot object with reaction norms

Examples

```
datos <- data.frame(
  gen = rep(c("A", "B"), each = 6),
  env = rep(1:3, each = 2, times = 2),
  trait = c(10, 12, 15, 16, 20, 22, 8, 9, 12, 14, 18, 20)
)
plot_reaction_norm(datos, env, trait, gen)
```

`plot_seasonal_curves` *Plot Seasonal Phenological Curves*

Description

Creates multi-panel plots showing seasonal NDVI dynamics across species and sites.

Usage

```
plot_seasonal_curves(
  time_series,
  selected_species = NULL,
  selected_year = 2021,
  n_populations = 3
)
```

Arguments

`time_series`	A data frame containing phenological time series
`selected_species`	Vector of species to include (NULL for all)
`selected_year`	Year to plot (default: 2021)
`n_populations`	Number of populations to include (default: 3)

Value

A ggplot object

Examples

```
## Not run:
data(pheno_time_series)
plot_seasonal_curves(pheno_time_series, selected_year = 2021)

## End(Not run)
```

`plot_spatial_map` *Plot Spatial Phenology Map*

Description

Creates a spatial map showing phenological patterns across locations.

Usage

```
plot_spatial_map(
  spatial_data,
  variable = "los",
  point_size = "ndvi_max",
  title = "Spatial Phenology Patterns"
)
```

Arguments

`spatial_data`	Spatial data frame (sf object or data.frame with coordinates)
`variable`	Variable to map as color (default: "los")
`point_size`	Variable to map as point size (default: "ndvi_max")
`title`	Plot title (optional)

Value

A ggplot object

`print.plasticity_meta_analysis_extended`
Print method for plasticity_meta_analysis_extended

Description

Print method for plasticity_meta_analysis_extended

Usage

```
## S3 method for class 'plasticity_meta_analysis_extended'
print(x, ...)
```

Arguments

`x`	An object of class plasticity_meta_analysis_extended
`...`	Additional arguments passed to print

`print.plasticity_tradeoffs_extended`

Print method for plasticity_tradeoffs_extended

Description

Print method for plasticity_tradeoffs_extended

Usage

```
## S3 method for class 'plasticity_tradeoffs_extended'
print(x, ...)
```

Arguments

`x`	An object of class plasticity_tradeoffs_extended
`...`	Additional arguments passed to print

`print.safe_multidim_result`

Print method for safe_multidim_result

Description

Print method for safe_multidim_result

Usage

```
## S3 method for class 'safe_multidim_result'
print(x, ...)
```

Arguments

`x`	An object of class safe_multidim_result
`...`	Additional arguments passed to print

`run_complete_analysis` *Run Complete Analysis Pipeline*

Description

Executes a comprehensive analysis pipeline on phenological data.

Usage

```
run_complete_analysis(time_series, parameters)
```

Arguments

`time_series`	Time series data (not always used, but kept for consistency)
`parameters`	A data frame containing extracted phenological parameters

Value

A list with all analysis results

Examples

```
## Not run:
data(pheno_time_series)
data(pheno_parameters)
results <- run_complete_analysis(pheno_time_series, pheno_parameters)

## End(Not run)
```

`safe_multidim_plasticity` *Safe Multidimensional Plasticity Analysis*

Description

A robust version of multidimensional plasticity analysis with error handling and data validation. This function calculates actual plasticity indices based on environmental variation in phenotypic traits.

Usage

```
safe_multidim_plasticity(
  data,
  traits,
  environments,
  groups,
  weights = NULL,
  na.action = "omit",
  ...
)
```

Arguments

`data`	A data frame containing phenotypic and environmental data
`traits`	Character vector of trait names to analyze
`environments`	Character vector of environmental variables
`groups`	Character string specifying the grouping variable (e.g., genotype, population)
`weights`	Optional numeric vector of weights for each trait (ecological importance)
`na.action`	How to handle missing values: "omit" (default), "fail", "impute.mean"
`...`	Additional arguments passed to the plasticity calculation

Value

A list containing:

`individual_plasticity`	Data frame with plasticity indices for each group
`group_plasticity`	Summary statistics by group
`multidimensional_index`	Overall multidimensional plasticity index
`safe_analysis`	List with analysis metadata and success status
`message`	Informative message about the analysis

Examples

```
# Create example data with real variability
set.seed(123)
example_data <- data.frame(
  genotype = rep(paste0("G", 1:10), each = 4),
  environment = rep(c("Cold", "Cool", "Warm", "Hot"), 10),
  trait1 = rnorm(40, mean = 10, sd = 2),
  trait2 = rnorm(40, mean = 5, sd = 1),
  trait3 = rnorm(40, mean = 20, sd = 3)
)

# Run analysis
```

```
result <- safe_multidim_plasticity(
  data = example_data,
  traits = c("trait1", "trait2", "trait3"),
  environments = "environment",
  groups = "genotype"
)

print(result)
```

`simulate_plasticity_data`

Simulate Phenotypic Plasticity Data

Description

Generates realistic simulated data for plasticity studies with customizable parameters for traits, environments, and plasticity patterns. Useful for teaching and method validation.

Usage

```
simulate_plasticity_data(
  n_genotypes = 20,
  n_environments = 5,
  n_traits = 3,
  plasticity_patterns = list("gradient", "optimal", "neutral"),
  trait_correlations = 0.3
)
```

Arguments

`n_genotypes`	Number of genotypes/populations
`n_environments`	Number of environmental conditions
`n_traits`	Number of phenotypic traits
`plasticity_patterns`	List of plasticity patterns: "gradient", "threshold", "optimal"
`trait_correlations`	Correlation structure between traits

Value

Simulated plasticity dataset

Index