

Package: limpidR (via r-universe)

September 30, 2026

Type Package

Title Reproducible Analysis of Freshwater Microplastic Data

Version 0.1.0

Author Chanikya Naidu [aut, cre]

Maintainer Chanikya Naidu <thefisherieschanikyaneeti@gmail.com>

Description Provides validation, harmonization, descriptive analysis, compositional analysis, transparent risk components, grouped cross-validation, visualization, and predictive modelling tools for freshwater microplastic datasets. The package includes synthetic demonstration data conforming to the LIMPID-India data model and emphasizes explicit units, provenance, percentage closure, non-imputation of missing environmental covariates, and leakage-aware model evaluation.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports ggplot2

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

Config/testthat/edition 3

VignetteBuilder knitr

RoxygenNote 7.3.2

NeedsCompilation no

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

Date/Publication 2026-09-30 13:55:42 UTC

RemoteUrl <https://github.com/cran/limpidR>

RemoteRef HEAD

RemoteSha 8d58a71139ff38b95f36a1218a02811452dd4973

Contents

limpidR-package	2
calculate_risk	2
load_limpid	3
model_mp_abundance	4
plot_lake_map	5
summarise_abundance	6
Index	8

limpidR-package	<i>Reproducible Analysis of Freshwater Microplastic Data</i>
-----------------	--

Description

Tools for loading, validating, harmonizing, analysing, modelling and visualizing freshwater microplastic datasets, with bundled synthetic demonstration tables that follow the LIMPID-India schema.

Details

The package emphasizes explicit units, provenance, composition closure, grouped validation, and transparent assumptions. Missing environmental values are not silently imputed.

Author(s)

Chanikya Naidu <thefisherieschanikyaaneeti@gmail.com>

calculate_risk	<i>Transparent risk components and reproducibility metadata</i>
----------------	---

Description

Calculate assumption-explicit risk components and report reproducibility/citation metadata.

Usage

```
calculate_risk(data, abundance_reference, hazard_scores = NULL,
  weights = NULL, component_max = NULL, thresholds = NULL)

limpid_session_info()

limpid_citation(doi = NULL, creators = "Chanikya Naidu")
```

Arguments

data	A limpid_db or event-level data frame.
abundance_reference	Positive reference abundance in the same unit as MP_Mean.
hazard_scores	Optional named polymer hazard scores supplied by the researcher.
weights	Optional named composite-score weights.
component_max	Explicit positive scaling maxima required when weights are used.
thresholds	Optional ascending numeric composite-score cut points.
doi	Assigned LIMPID-India dataset DOI.
creators	Creator string for citation output.

Details

No universal polymer hazard scores, composite weights or risk classes are hard-coded. Assumptions must be supplied explicitly.

Value

A risk-component data frame, sessionInfo object, or citation string.

load_limpid	<i>Load, clean and validate freshwater microplastic data</i>
-------------	--

Description

Functions for reading LIMPID-India tables, unit harmonization and structural validation.

Usage

```
limpid_tables()

load_limpid(path = NULL, tables = limpid_tables(), validate = TRUE, quiet = FALSE)

check_database(data, strict = FALSE, tolerance = 0.2)

validate_mp_data(data, type = c("abundance", "morphology", "size", "polymer",
  "events", "lake"), tolerance = 0.2)

clean_mp_data(data, abundance_col = "MP_Mean", unit_col = "Unit",
  target_unit = "particles L-1", invalid_action = c("error", "flag", "na"))

convert_mp_units(x, from, to = "particles L-1")
```

Arguments

path	NULL, a compatible CSV directory, or an Excel workbook. Excel input requires readxl .
tables	Character vector of table names.
validate	Logical; validate after loading.
quiet	Logical; suppress load message.
data	A data frame or limpid_db, depending on function.
strict	Logical; stop when a failing database check is found.
tolerance	Allowed percentage deviation from 100 for composition closure.
type	Expected table schema.
abundance_col	Name of abundance column.
unit_col	Name of unit column.
target_unit	Target abundance unit.
invalid_action	Action for invalid abundance values.
x	Numeric abundance vector.
from	Source abundance unit.
to	Target abundance unit.

Value

A loaded database, validation table, cleaned object, or converted numeric vector.

Examples

```
db <- load_limpid()
check_database(db)
convert_mp_units(1000, "particles m^-3", "particles L^-1")
```

model_mp_abundance	<i>Model and validate freshwater microplastic abundance</i>
--------------------	---

Description

Lognormal or Gamma regression with original-scale prediction and grouped cross-validation.

Usage

```
model_mp_abundance(data, formula = NULL,
  method = c("lognormal_lm", "gamma_glm"), na_action = c("omit", "fail"))

predict_mp(model, newdata, interval = c("none", "confidence", "prediction"), level = 0.95)

cross_validate_mp(data, formula = NULL,
  method = c("lognormal_lm", "gamma_glm"), group = "Lake_ID")
```

Arguments

data	A limpid_db or model-ready data frame.
formula	Model formula with untransformed abundance response.
method	Model family.
na_action	Missing-data handling for fitting.
model	A limpid_model.
newdata	Data frame for prediction.
interval	Prediction interval type.
level	Interval confidence level.
group	Grouping column for held-out folds.

Details

The default lognormal model uses a log1p response and Duan smearing for back-transformation. Grouped cross-validation defaults to Lake_ID to reduce spatial leakage.

Value

A limpid_model, prediction data frame, or cross-validation result list.

Examples

```
db <- load_limpid()
m <- model_mp_abundance(db, MP_Mean ~ Season_Global + Lake_Type)
cross_validate_mp(db, MP_Mean ~ Season_Global + Lake_Type)
```

plot_lake_map	<i>Visualize freshwater microplastic patterns</i>
---------------	---

Description

Seasonal, composition, depth and coordinate-based point visualizations.

Usage

```
plot_lake_map(data, value_col = "MP_Mean")

classify_hotspots(data, value_col = "MP_Mean", method = c("quantile", "robust_z"))

map_mp_hotspots(data, value_col = "MP_Mean", method = c("quantile", "robust_z"))

plot_seasonality(data, lake = NULL, value_col = "MP_Mean")

plot_morphology_profile(data, id_col = "Lake_Name")
```

```
plot_polymer_profile(data, id_col = "Lake_Name")

plot_depth_profile(data, depth_col = "Depth_m", value_col = "MP_Mean", group_col = NULL)
```

Arguments

<code>data</code>	A <code>limpid_db</code> or appropriate data frame.
<code>value_col</code>	Numeric indicator column.
<code>method</code>	Relative hotspot classification method.
<code>lake</code>	Optional lake name or identifier filter.
<code>id_col</code>	Identity/group column for profile bars.
<code>depth_col</code>	Depth column in metres.
<code>group_col</code>	Optional depth-profile grouping column.

Details

Hotspot functions provide relative point classification and do not imply spatial interpolation. Depth plotting requires genuinely depth-resolved data and will reject the bundled event-level database.

Value

A **ggplot2** object, except `classify_hotspots()` which returns a data frame.

<code>summarise_abundance</code>	<i>Descriptive and compositional microplastic analysis</i>
----------------------------------	--

Description

Summaries, event-level joins, composition closure, CLR transformation and Aitchison distance.

Usage

```
summarise_abundance(data, by = c("Lake_Name", "Season_Global"),
  value_col = "MP_Mean", conf_level = 0.95)

analyse_morphology(data, group_by = c("Lake_Name", "Season_Global"))

analyse_size_distribution(data, group_by = c("Lake_Name", "Season_Global"))

analyse_polymers(data, group_by = c("Lake_Name", "Season_Global"))

composition_closure(data, cols, tolerance = 0.2,
  action = c("flag", "renormalize", "error"))

clr_transform(data, cols, pseudocount = 1e-06)
```

```
aitchison_distance(data, cols, pseudocount = 1e-06)

build_model_data(data, include_environment = TRUE, primary_eligible_only = FALSE)

provenance_summary(data)
```

Arguments

<code>data</code>	A <code>limpid_db</code> or data frame, depending on function.
<code>by</code>	Grouping columns for abundance summaries.
<code>value_col</code>	Numeric abundance or indicator column.
<code>conf_level</code>	Confidence level for mean confidence intervals.
<code>group_by</code>	Optional composition grouping columns.
<code>cols</code>	Composition component columns.
<code>tolerance</code>	Allowed deviation from 100 percent.
<code>action</code>	Closure action.
<code>pseudocount</code>	Positive zero replacement on the proportion scale.
<code>include_environment</code>	Join environmental context into model data.
<code>primary_eligible_only</code>	Restrict to explicitly primary-model-eligible rows.

Value

A data frame, distance object, or named list, depending on function.

Examples

```
db <- load_limpid()
summarise_abundance(db)
analyse_polymers(db)
```

Index

- * **package**
 - limpidR-package, [2](#)
- aitchison_distance
 - (summarise_abundance), [6](#)
- analyse_morphology
 - (summarise_abundance), [6](#)
- analyse_polymers (summarise_abundance), [6](#)
- analyse_size_distribution
 - (summarise_abundance), [6](#)
- build_model_data (summarise_abundance), [6](#)
- calculate_risk, [2](#)
- check_database (load_limpid), [3](#)
- classify_hotspots (plot_lake_map), [5](#)
- clean_mp_data (load_limpid), [3](#)
- clr_transform (summarise_abundance), [6](#)
- composition_closure
 - (summarise_abundance), [6](#)
- convert_mp_units (load_limpid), [3](#)
- cross_validate_mp (model_mp_abundance), [4](#)
- limpid_citation (calculate_risk), [2](#)
- limpid_session_info (calculate_risk), [2](#)
- limpid_tables (load_limpid), [3](#)
- limpidR (limpidR-package), [2](#)
- limpidR-package, [2](#)
- load_limpid, [3](#)
- map_mp_hotspots (plot_lake_map), [5](#)
- model_mp_abundance, [4](#)
- plot_depth_profile (plot_lake_map), [5](#)
- plot_lake_map, [5](#)
- plot_morphology_profile
 - (plot_lake_map), [5](#)
- plot_polymer_profile (plot_lake_map), [5](#)
- plot_seasonality (plot_lake_map), [5](#)
- predict.limpid_model
 - (model_mp_abundance), [4](#)
- predict_mp (model_mp_abundance), [4](#)
- print.limpid_db (load_limpid), [3](#)
- print.limpid_model
 - (model_mp_abundance), [4](#)
- print.limpid_validation (load_limpid), [3](#)
- provenance_summary
 - (summarise_abundance), [6](#)
- summarise_abundance, [6](#)
- validate_mp_data (load_limpid), [3](#)