

Package: gcf (via r-universe)

October 1, 2026

Title Generalized Covariate Field

Version 0.1.1

Description Generates generalized covariate field (GCF) variables from spatial covariates observed at projected coordinates, and selects a stable subset of them for geospatial prediction. For each input covariate the method builds spatial-pattern features (local indicator of spatial association, local Geary's c, log local variance, rank quantile entropy, geocomplexity, log scale variance, local variogram exponent, and signed z-score and median absolute deviation outlier strengths over a series of buffer radii) and neighbourhood-distribution features (buffer-wise quantiles of the covariate values surrounding each location), reduces the buffer and quantile sweeps to a compact set of interpretable functional summaries, and selects variables by random forest importance combined with spatial-block stability resampling and group voting. The GCF method is positioned as prediction-oriented feature construction: its output feeds any downstream regression learner. Methods are described in Song (2026) [doi:10.1080/13658816.2026.2729719](https://doi.org/10.1080/13658816.2026.2729719).

License GPL-3

Encoding UTF-8

Depends R (>= 4.1.0)

Imports geocomplexity, ranger, sf, spdep (>= 1.1-7), stats, utils

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

VignetteBuilder knitr

Config/testthat/edition 3

LazyData true

LazyDataCompression xz

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Yongze Song [aut, cre, cph] (ORCID:
 <<https://orcid.org/0000-0003-3420-9622>>)
Maintainer Yongze Song <yongze.song@outlook.com>
Repository <https://cran.r-universe.dev>
Date/Publication 2026-09-30 23:59:34 UTC
RemoteUrl <https://github.com/cran/gcf>
RemoteRef HEAD
RemoteSha a5a96740608765b1e7ed1bd716e21c620ba38fba

Contents

bio_grid	2
gcf_blocks	4
gcf_field	5
gcf_psi	7
gcf_reduce	9
gcf_select	10
gcf_zx	12
sim_grid	14
summary.gcf_field	15
Index	16

bio_grid	<i>South-western Australia plant species richness grid (GCF paper case study)</i>
----------	---

Description

The case-study dataset of the GCF paper: vascular plant species richness and twelve environmental covariates over the Southwest Australian Floristic Region (SWAFR), aggregated to a 10-km grid of 6229 cells. Species richness observations (4989 plot records) are aggregated into 958 grid cells; the remaining cells carry the covariates only. The rows with observed = TRUE (equivalently richness > 0) are the observed sample locations used to fit and validate prediction models; richness is recorded as 0 where no sample exists (this is a missing observation, not a true zero richness).

Usage

bio_grid

Format

A data frame with 6229 rows and 19 columns:

GridID grid cell identifier.

lon, lat cell centre longitude/latitude (degrees, GDA94).

xkm, ykm projected cell centre coordinates in kilometres (EPSG:3577, GDA94 Australian Albers, equal-area); use these for GCF buffers and blocks.

richness mean vascular plant species richness of the samples in the cell; 0 where the cell holds no sample.

observed logical; TRUE for the 958 cells with observed richness.

Elevation elevation (m).

Slope terrain slope (degrees).

Precipitation annual average precipitation (mm/year).

Radiation annual average shortwave radiation (W/m^2).

DistWater distance to the nearest water body (km).

DistBuilt distance to artificial land or urban area (km).

SoilN soil total nitrogen content (percent).

SoilC soil organic carbon content (percent).

SoilClay soil clay fraction (percent).

SoilDepth depth to impermeable layer from surface (m).

SoilpH average soil pH value.

SoilBD soil bulk density (g/cm^3).

Details

The paper builds the GCF variables on the projected kilometre coordinates xkm/ykm with buffers 20, 30, ..., 100 km (with $d_{\text{norm}} = 100$), quantile levels $\text{seq}(0, 1, 0.05)$, fine band {20, 30} km and broad band {90, 100} km, and selects variables with spatial blocks of side 132 km.

Source

Case-study data of Song, Y. (2026). Generalized covariate field (GCF): spatial-pattern and neighbourhood-distribution feature expansion improves geospatial prediction. *International Journal of Geographical Information Science*, 40, 1–29. doi:10.1080/13658816.2026.2729719. The richness grid and the covariates are the paper's case-study data, archived with the paper's reproducibility materials at doi:10.6084/m9.figshare.29425739; Table 3 of the paper lists the original source of each variable. The richness observations derive from Mokany, K. et al. (2022). Patterns and drivers of plant diversity across Australia. *Ecography*. doi:10.1111/ecog.06426. Covariates were derived from SRTM elevation, Digital Earth Australia, and CSIRO TERN Landscape soil products.

See Also

[gcf_field\(\)](#), [gcf_select\(\)](#), [sim_grid](#)

Examples

```
data(bio_grid)
table(bio_grid$observed)
summary(bio_grid$richness[bio_grid$observed])
```

gcf_blocks

*Spatial blocks for stability selection and spatial cross-validation***Description**

Assigns each location to a square spatial block of the given side length, on projected coordinates. Block ids are used by [gcf_select\(\)](#) for spatial-block stability resampling, and can also define spatial cross-validation folds.

Usage

```
gcf_blocks(coords, size)
```

Arguments

coords	A two-column matrix or data frame of projected coordinates.
size	Block side length, in coordinate units (the paper's case study uses 132 km, twice the residual variogram range; the simulation uses 6 grid units).

Value

A character vector of block ids, one per location.

Examples

```
data(sim_grid)
blocks <- gcf_blocks(sim_grid[, c("x", "y")], size = 6)
table(blocks)
```

gcf_field

*Generate generalized covariate field (GCF) variables***Description**

The main generator of the GCF method: one or multiple spatial variables with projected coordinates in, their GCF variables out. `gcf_field()` runs the three feature-construction steps in one call:

1. `gcf_psi()` – spatial-pattern features (11 operators over buffer radii);
2. `gcf_zx()` – neighbourhood-distribution features (buffer-wise quantiles);
3. `gcf_reduce()` – functional reduction of the buffer/quantile sweeps to a compact candidate set of X (raw), P (pattern), and D (context) variables.

The resulting candidate variables, together with the raw covariates, form the GCF candidate field that `gcf_select()` screens for a stable subset. No response variable is used at any point of the generation.

Usage

```
gcf_field(
  data,
  coords,
  vars = NULL,
  buffers,
  probs = seq(0, 1, 0.05),
  d_norm = max(buffers),
  theta = 2,
  bins = 10,
  include_vario_exp = TRUE,
  vario_buffers = NULL,
  fine_band = min(buffers),
  broad_band = max(buffers),
  d_mode = "functional"
)
```

Arguments

data	A data frame with one row per location, containing the spatial variables (and optionally the coordinate columns).
coords	Either a length-2 character vector naming the projected coordinate columns of data (for example <code>c("x", "y")</code>), or a two-column matrix or data frame of projected coordinates. Coordinate units must match buffers (for example kilometres).
vars	Character vector of variable names to expand. Defaults to all columns of data except the coordinate columns; make sure the response and any non-numeric columns (which raise an error) are excluded when relying on the default.

buffers	Numeric vector of positive buffer radii, in coordinate units (the paper uses 20–100 km for the case study and 2, 4, 6 grid units for the simulation).
probs	Numeric vector of quantile levels in [0, 1]. Defaults to seq(0, 1, 0.05) (21 levels, as in the paper’s case study).
d_norm	Radius of the local normalization neighbourhood used by the LISA operator. Defaults to max(buffers).
theta	Outlier threshold for the z-score and MAD outlier strengths (default 2, as in the paper).
bins	Number of rank bins for the quantile entropy operator (default 10, as in the paper).
include_vario_exp	Logical; include the local variogram exponent operator (default TRUE, the full 11-operator catalogue).
vario_buffers	Buffer radii for the local variogram exponent. Defaults to buffers.
fine_band, broad_band	Buffer radii forming the fine and broad scale bands of the functional reduction. Default to min(buffers) and max(buffers); the paper’s case study uses c(20, 30) and c(90, 100) km with buffers 20–100 km.
d_mode	Representation of the D layer in the reduction; see gcf_reduce() . Default "functional" (as in the paper).

Value

An object of class "gcf_field": a list with elements

candidates data frame of candidate variables (one row per location): the raw covariates (category "X") plus the GCF variables (categories "P" and "D").

meta data frame describing each candidate column: feature, group (base variable), category ("X", "P", "D"), operator, scale.

vars the retained input variable names.

psi the "gcf_psi" object (full pattern-feature sweep).

zx the "gcf_zx" object (full context-quantile sweep).

params the reduction parameters.

References

Song, Y. (2026). Generalized covariate field (GCF): spatial-pattern and neighbourhood-distribution feature expansion improves geospatial prediction. *International Journal of Geographical Information Science*, 40, 1–29. doi:10.1080/13658816.2026.2729719

See Also

[gcf_select\(\)](#) for the follow-up variable selection; [gcf_psi\(\)](#), [gcf_zx\(\)](#), [gcf_reduce\(\)](#) for the individual steps.

Examples

```
# GCF variables of two simulated covariates on a 10 x 10 grid subset
data(sim_grid)
sub <- sim_grid[sim_grid$x <= 10 & sim_grid$y <= 10, ]
field <- gcf_field(sub, coords = c("x", "y"), vars = c("x1", "x2"),
                  buffers = c(2, 4), probs = seq(0, 1, 0.1), d_norm = 4)

field
head(field$candidates[, 1:6])

# Full simulation grid, paper settings (buffers 2, 4, 6; 11 quantiles)
field <- gcf_field(sim_grid, coords = c("x", "y"),
                  vars = c("x1", "x2", "x3"),
                  buffers = c(2, 4, 6), probs = seq(0, 1, 0.1),
                  d_norm = 4, fine_band = 2, broad_band = 6)

summary(field)
```

gcf_psi

Spatial-pattern features (psi) of spatial variables

Description

Step 1 of the generalized covariate field (GCF) method. For each input variable, `gcf_psi()` computes the 11 spatial-pattern operators of the GCF method over a series of buffer radii: local indicator of spatial association (LISA) on locally normalized values, local Geary's *c*, log local variance, rank-binned quantile entropy, geocomplexity ($k = 23$ nearest neighbours, single scale), log scale-variance (single scale), local variogram exponent, and positive/negative *z*-score and MAD outlier strengths. The computation never uses a response variable.

Usage

```
gcf_psi(
  data,
  coords,
  vars = NULL,
  buffers,
  d_norm = max(buffers),
  theta = 2,
  bins = 10,
  include_vario_exp = TRUE,
  vario_buffers = NULL
)
```

Arguments

<code>data</code>	A data frame with one row per location, containing the spatial variables (and optionally the coordinate columns).
<code>coords</code>	Either a length-2 character vector naming the projected coordinate columns of data (for example <code>c("x", "y")</code>), or a two-column matrix or data frame of projected coordinates. Coordinate units must match buffers (for example kilometres).
<code>vars</code>	Character vector of variable names to expand. Defaults to all columns of data except the coordinate columns; make sure the response and any non-numeric columns (which raise an error) are excluded when relying on the default.
<code>buffers</code>	Numeric vector of positive buffer radii, in coordinate units (the paper uses 20–100 km for the case study and 2, 4, 6 grid units for the simulation).
<code>d_norm</code>	Radius of the local normalization neighbourhood used by the LISA operator. Defaults to <code>max(buffers)</code> .
<code>theta</code>	Outlier threshold for the z-score and MAD outlier strengths (default 2, as in the paper).
<code>bins</code>	Number of rank bins for the quantile entropy operator (default 10, as in the paper).
<code>include_vario_exp</code>	Logical; include the local variogram exponent operator (default TRUE, the full 11-operator catalogue).
<code>vario_buffers</code>	Buffer radii for the local variogram exponent. Defaults to <code>buffers</code> .

Details

Non-finite covariate values are imputed by the column median and zero-variance columns are dropped before feature construction.

Value

An object of class `"gcf_psi"`: a list with elements

features data frame of pattern features (one row per location).

map data frame describing each feature column (base variable, group id, operator category, buffer).

vars the retained variable names.

params the parameters used.

References

Song, Y. (2026). Generalized covariate field (GCF): spatial-pattern and neighbourhood-distribution feature expansion improves geospatial prediction. *International Journal of Geographical Information Science*, 40, 1–29. doi:10.1080/13658816.2026.2729719

See Also

`gcf_field()` for the full GCF variable generation pipeline, `gcf_zx()` for the neighbourhood-distribution features.

Examples

```
data(sim_grid)
sub <- sim_grid[sim_grid$x <= 10 & sim_grid$y <= 10, ]
psi <- gcf_psi(sub, coords = c("x", "y"), vars = c("x1", "x2"),
              buffers = c(2, 4), d_norm = 4)

psi
head(psi$features[, 1:4])
```

gcf_reduce

Reduce psi and Zx layers to the GCF candidate variables

Description

Step 3a of the generalized covariate field (GCF) method. The buffer and quantile sweeps produced by `gcf_psi()` and `gcf_zx()` are collinear; `gcf_reduce()` collapses them into a compact set of interpretable functionals over two scale bands (a leakage-free per-row transform):

- **D (context):** per variable and band, from the band-averaged quantile curve, five functionals: med (q0.50), iqr (q0.75 - q0.25), lotail (q0.10), hitail (q0.90), and skew (q0.90 + q0.10 - 2 q0.50).
- **P (pattern):** per variable and operator, the band-averaged buffered operators; the single-scale operators (geocomplexity, scale variance) are kept as-is.
- **X:** the raw covariates passed through unchanged.

Usage

```
gcf_reduce(
  data,
  psi,
  zx,
  fine_band = NULL,
  broad_band = NULL,
  d_mode = "functional"
)
```

Arguments

data	The data frame that was passed to <code>gcf_psi()</code> and <code>gcf_zx()</code> (supplies the raw covariate columns).
psi	A "gcf_psi" object from <code>gcf_psi()</code> .
zx	A "gcf_zx" object from <code>gcf_zx()</code> .
fine_band	Numeric vector of buffer radii forming the fine scale band. Defaults to the smallest buffer used; the paper's case study uses <code>c(20, 30)</code> km.
broad_band	Numeric vector of buffer radii forming the broad scale band. Defaults to the largest buffer used; the paper's case study uses <code>c(90, 100)</code> km.

d_mode Representation of the D layer: "functional" (default, the five functionals per band as in the paper), "functional_allscale" (functionals at every buffer), "qgrid" (raw quantile columns on a coarse grid), or "full" (all quantile columns unchanged).

Value

An object of class "gcf_field"; see [gcf_field\(\)](#) for its structure.

See Also

[gcf_field\(\)](#), which runs [gcf_psi\(\)](#), [gcf_zx\(\)](#), and [gcf_reduce\(\)](#) in one call.

Examples

```
data(sim_grid)
sub <- sim_grid[sim_grid$x <= 10 & sim_grid$y <= 10, ]
psi <- gcf_psi(sub, coords = c("x", "y"), vars = c("x1", "x2"),
  buffers = c(2, 4), d_norm = 4)
zx <- gcf_zx(sub, coords = c("x", "y"), vars = c("x1", "x2"),
  buffers = c(2, 4), probs = seq(0, 1, 0.1))
field <- gcf_reduce(sub, psi, zx, fine_band = 2, broad_band = 4)
field
```

gcf_select

Select stable GCF variables

Description

Step 3b of the generalized covariate field (GCF) method: screens the candidate variables of a [gcf_field\(\)](#) for a stable subset. The raw covariates (category "X") are always kept; the derived P/D variables are selected by the `rf_imp` kernel with spatial-block stability and medium group voting:

1. **Per-subsample selector (`rf_imp`).** On each subsample, fit a random forest (**ranger**, `num_trees` trees) and keep the top `k` top derived variables by impurity importance.
2. **Spatial-block stability.** Repeat `B` times on subsamples drawn as `floor(subsample_frac * M)` of the `M` spatial blocks, recording which variables are kept each time.
3. **Medium group voting.** Group the derived variables by (variable x category); a group qualifies if any member is kept in at least `pi_thr` of the subsamples, and contributes its most frequently kept member as representative.

Group voting de-dilutes collinear blocks: sibling variables split the selection frequency so that none clears the threshold individually, while the block as a whole fires reliably.

Usage

```
gcf_select(
  x,
  y,
  blocks,
  train = NULL,
  B = 80,
  subsample_frac = 0.7,
  pi_thr = 0.6,
  ktop = 20,
  num_trees = 200,
  seed = NULL
)
```

Arguments

x	A "gcf_field" object from gcf_field() or gcf_reduce() (or a list with elements candidates and meta in the same format).
y	Numeric response vector, one value per location of the field. The response is used for selection only; gcf_field() never sees it.
blocks	Vector of spatial block ids, one per location; see gcf_blocks() .
train	Optional integer vector of training row indices to run the selection on (for example the training part of a cross-validation fold). Defaults to all rows.
B	Number of stability resamples (default 80, as in the paper).
subsample_frac	Fraction of spatial blocks drawn per resample (default 0.7).
pi_thr	Group fire-frequency threshold (default 0.6).
ktop	Number of top derived variables kept per subsample (default 20).
num_trees	Number of trees of the random forest importance kernel (default 200).
seed	Optional integer seed for the block resampling and the random forests. NULL (default) leaves the random number generator untouched; the paper uses seed = 1. When supplied, the generator state is restored on exit.

Details

The random forests always run single-threaded. When seed is supplied the selection is fully reproducible: the seed governs both the block resampling and the ranger forests, and the caller's random number generator state is restored on exit. When seed = NULL (the default) no seed is set and the procedure follows the current random number generator stream; set a seed yourself before the call for reproducibility.

Value

An object of class "gcf_selection": a list with elements

selected character vector of selected variable names (forced raw covariates first, then the group representatives).

forced the raw covariate names (always kept).

derived the selected derived (P/D) variable names.

freq named numeric vector: per-variable selection frequency across the B resamples (derived variables, sorted decreasing).

group_fire named numeric vector: per-group fire frequency.

params the selection parameters.

References

Song, Y. (2026). Generalized covariate field (GCF): spatial-pattern and neighbourhood-distribution feature expansion improves geospatial prediction. *International Journal of Geographical Information Science*, 40, 1–29. doi:10.1080/13658816.2026.2729719

See Also

[gcf_field\(\)](#), [gcf_blocks\(\)](#).

Examples

```
data(sim_grid)
sub <- sim_grid[sim_grid$x <= 12 & sim_grid$y <= 12, ]
field <- gcf_field(sub, coords = c("x", "y"), vars = c("x1", "x2", "x3"),
  buffers = c(2, 4), probs = seq(0, 1, 0.1), d_norm = 4)
blocks <- gcf_blocks(sub[, c("x", "y")], size = 6)
# B reduced from the paper's 80 to keep the example fast
sel <- gcf_select(field, y = sub$y1, blocks = blocks, B = 5, seed = 1)
sel

# Full simulation grid with the paper's field settings; B = 20 resamples
# here to keep the example short (the paper uses B = 80; see the vignette)
field <- gcf_field(sim_grid, coords = c("x", "y"),
  vars = c("x1", "x2", "x3"),
  buffers = c(2, 4, 6), probs = seq(0, 1, 0.1),
  d_norm = 4, fine_band = 2, broad_band = 6)
blocks <- gcf_blocks(sim_grid[, c("x", "y")], size = 6)
sel <- gcf_select(field, y = sim_grid$y1, blocks = blocks, B = 20, seed = 1)
sel$selected
```

Description

Step 2 of the generalized covariate field (GCF) method. For each location v , variable x and buffer radius b , `gcf_zx()` summarises the distribution of the variable over the locations within the buffer by quantile levels τ :

$$Z_x(v; b, \tau) = Q_\tau(\{x(u) : u \in N(v, b)\}).$$

The buffer neighbourhood includes the location itself. The computation never uses a response variable.

Usage

```
gcf_zx(data, coords, vars = NULL, buffers, probs = seq(0, 1, 0.05))
```

Arguments

<code>data</code>	A data frame with one row per location, containing the spatial variables (and optionally the coordinate columns).
<code>coords</code>	Either a length-2 character vector naming the projected coordinate columns of data (for example <code>c("x", "y")</code>), or a two-column matrix or data frame of projected coordinates. Coordinate units must match buffers (for example kilometres).
<code>vars</code>	Character vector of variable names to expand. Defaults to all columns of data except the coordinate columns; make sure the response and any non-numeric columns (which raise an error) are excluded when relying on the default.
<code>buffers</code>	Numeric vector of positive buffer radii, in coordinate units (the paper uses 20–100 km for the case study and 2, 4, 6 grid units for the simulation).
<code>probs</code>	Numeric vector of quantile levels in $[0, 1]$. Defaults to <code>seq(0, 1, 0.05)</code> (21 levels, as in the paper’s case study).

Details

Non-finite covariate values are imputed by the column median and zero-variance columns are dropped before feature construction. A buffer that contains no locations yields NA.

Value

An object of class `"gcf_zx"`: a list with elements

features data frame of context-quantile features (one row per location; columns named `<var>_b<buffer>_q<prob>`).

map data frame describing each feature column (base variable, group id, buffer, quantile level).

vars the retained variable names.

params the parameters used.

References

Song, Y. (2026). Generalized covariate field (GCF): spatial-pattern and neighbourhood-distribution feature expansion improves geospatial prediction. *International Journal of Geographical Information Science*, 40, 1–29. doi:10.1080/13658816.2026.2729719

See Also

[gcf_field\(\)](#) for the full GCF variable generation pipeline, [gcf_psi\(\)](#) for the spatial-pattern features.

Examples

```
data(sim_grid)
sub <- sim_grid[sim_grid$x <= 10 & sim_grid$y <= 10, ]
zx <- gcf_zx(sub, coords = c("x", "y"), vars = c("x1", "x2"),
            buffers = c(2, 4), probs = c(0.1, 0.5, 0.9))
zx
head(zx$features)
```

sim_grid

Simulated spatial grid (GCF paper simulation)

Description

The 30 x 30 simulation grid of the GCF paper: a response surface and three spatially structured covariates on a 900-cell regular grid with unit spacing. The paper builds the GCF variables with buffers 2, 4, 6 (with `d_norm = 4`), quantile levels `seq(0, 1, 0.1)`, fine band {2} and broad band {6}, and evaluates prediction of `y1` from the covariates by random and spatial-block five-fold cross-validation (spatial blocks of side 6).

Usage

```
sim_grid
```

Format

A data frame with 900 rows and 6 columns:

y1 simulated response.

x1, x2, x3 simulated spatial covariates.

x, y projected grid coordinates (column and row, 1–30, unit spacing).

Source

Simulation data of Song, Y. (2026). Generalized covariate field (GCF): spatial-pattern and neighbourhood-distribution feature expansion improves geospatial prediction. *International Journal of Geographical Information Science*, 40, 1–29. doi:[10.1080/13658816.2026.2729719](https://doi.org/10.1080/13658816.2026.2729719)

See Also

[gcf_field\(\)](#), [gcf_select\(\)](#), [bio_grid](#)

Examples

```
data(sim_grid)
head(sim_grid)
```

summary.gcf_field	<i>Summarise a generalized covariate field</i>
-------------------	--

Description

Summarise a generalized covariate field

Usage

```
## S3 method for class 'gcf_field'
summary(object, ...)
```

Arguments

object	A "gcf_field" object from gcf_field() or gcf_reduce() .
...	Unused.

Value

object, invisibly. Prints a per-variable and per-category breakdown of the candidate variables.

Index

* datasets

- bio_grid, [2](#)
- sim_grid, [14](#)

bio_grid, [2](#), [14](#)

gcf_blocks, [4](#)
gcf_blocks(), [11](#), [12](#)
gcf_field, [5](#)
gcf_field(), [3](#), [8](#), [10–12](#), [14](#), [15](#)
gcf_psi, [7](#)
gcf_psi(), [5](#), [6](#), [9](#), [10](#), [14](#)
gcf_reduce, [9](#)
gcf_reduce(), [5](#), [6](#), [11](#), [15](#)
gcf_select, [10](#)
gcf_select(), [3–6](#), [14](#)
gcf_zx, [12](#)
gcf_zx(), [5](#), [6](#), [8–10](#)

sim_grid, [3](#), [14](#)
summary.gcf_field, [15](#)