

Package: cvr (via r-universe)

October 15, 2026

Title Polygonal Coverage Operations via GEOS

Version 0.0.0.9000

Description Bindings to the GEOS coverage API: topology-preserving simplification of polygonal coverages, coverage validation, and coverage union. These are set-level operators that consume a whole vector of polygons as one coverage and return a result parallel to it, so shared edges are simplified once and stay edge-matched in both neighbours. Geometry is exchanged as well-known binary, so any 'wk' compatible vector is accepted and returned. Requires GEOS 3.12.0 or later.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.0)

Imports wk

SystemRequirements GEOS (>= 3.12.0), GNU make

NeedsCompilation yes

Author Michael Sumner [aut, cre]

Maintainer Michael Sumner <mdsumner@gmail.com>

Config/roxygen2/version 8.1.0

Config/pak/sysreqs libgeos-dev make

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

Date/Publication 2026-09-19 22:14:16 UTC

RemoteUrl <https://github.com/hypertidy/cvr>

RemoteRef HEAD

RemoteSha e8a8b82afa8e32d95db94e8b8867b87ec4bf2779

Contents

cvr_coverage	2
cvr_geos_version	4

Index	5
--------------	----------

cvr_coverage

*Polygonal coverage operations***Description**

Bindings to the GEOS coverage API (GEOS $\geq$ 3.12). A polygonal coverage is a set of valid polygons that are non-overlapping and edge-matched: vertices along a shared edge are identical in both neighbours.

Usage

```
cvr_simplify(x, tolerance, boundary = TRUE)
```

```
cvr_is_valid(x, gap_width = 0)
```

```
cvr_invalid_edges(x, gap_width = 0)
```

```
cvr_union(x)
```

```
cvr_n_coord(x)
```

```
cvr_simplify_each(x, tolerance)
```

```
cvr_area(x)
```

```
cvr_gap(x)
```

Arguments

<code>x</code>	A vector of polygons: anything [wk::as_wkb()] accepts, such as a 'wk_wkb', 'wk_wkt', 'sfc' or 'sf' object. All members must be POLYGON or MULTI-POLYGON.
<code>tolerance</code>	Simplification tolerance in the units of 'x', roughly the square root of the area of the triangles removed. Length 1.
<code>boundary</code>	Simplify the outer boundary of the coverage as well as the inner shared edges? 'FALSE' simplifies inner edges only.
<code>gap_width</code>	The maximum width of gaps to report as invalid.

Details

These are set-level operators, not row-wise ones. They consume the whole vector as a single coverage and return a result parallel to it. This is why PostGIS exposes 'ST_CoverageSimplify' as a window function rather than an ordinary one, and why the result of 'cvr_simplify()' cannot be computed feature by feature.

'cvr_simplify()' applies a Visvalingam-Whyatt simplification to the coverage edges. Shared edges are simplified once and stay identical in both neighbours, so no gaps or overlaps open up. Node

points (inner vertices shared by three or more polygons, boundary vertices shared by two or more) are never moved. Features never disappear, though they may be reduced to a triangle.

With `'boundary = FALSE'` only the inner, shared edges are simplified and the outer boundary of the coverage is left untouched. A subset simplified this way still fits exactly against the rest of the coverage, which is what makes chunked or incremental generalization possible.

Value

`'cvr_simplify()'` a `'wk_wkb'` the same length as `'x'`, in the same order, carrying the CRS of `'x'`.

`'cvr_is_valid()'` a single `'TRUE'` or `'FALSE'` for the coverage as a whole.

`'cvr_invalid_edges()'` a `'wk_wkb'` the same length as `'x'`: a `MULTILINESTRING` of the offending edges for each invalid member, and an empty geometry where that member is a valid participant.

`'cvr_union()'` a length-1 `'wk_wkb'`.

`'cvr_n_coord()'` a numeric vector of vertex counts, one per member.

`'cvr_area()'` a numeric vector of planar areas, one per member.

`'cvr_simplify_each()'` a `'wk_wkb'` the same length as `'x'`, each member simplified independently with `'GEOSTopologyPreserveSimplify'`. The control case, not the one to use: same library, same algorithm family, but no knowledge of neighbours. It is here so the cost of ignoring them can be measured rather than asserted.

`'cvr_gap()'` a single number: the absolute difference between the summed member areas and the area of their union. Zero for a coverage. Grows as soon as shared edges drift apart, so it is the direct test of whether a simplification stayed edge-matched.

Examples

```
library(wk)

# The four-polygon coverage from the PostGIS ST_CoverageSimplify docs
cov <- as_wkb(wkt(c(
  paste0("POLYGON ((160 150, 110 130, 90 100, 90 70, 60 60, 50 10, 30 30,",
    " 40 50, 25 40, 10 60, 30 100, 30 120, 20 170, 60 180, 90 190,",
    " 130 180, 130 160, 160 150)),",
    " (40 160, 50 140, 66 125, 60 100, 80 140, 90 170, 60 160, 40 160))"),
  "POLYGON ((40 160, 60 160, 90 170, 80 140, 60 100, 66 125, 50 140, 40 160))",
  "POLYGON ((110 130, 160 50, 140 50, 120 33, 90 30, 50 10, 60 60, 90 70, 90 100, 110 130))",
  "POLYGON ((160 150, 150 120, 160 90, 160 50, 110 130, 160 150))"
)))

cvr_is_valid(cov)
sum(cvr_n_coord(cov))

simp <- cvr_simplify(cov, 30)
sum(cvr_n_coord(simp))

# still a coverage: shared edges moved together
cvr_is_valid(simp)

# inner-only simplification pins the outer boundary
```

```
inner <- cwr_simplify(cov, 30, boundary = FALSE)
cwr_is_valid(inner)
```

cwr_geos_version	<i>GEOS version cwr was built against</i>
------------------	---

Description

GEOS version cwr was built against

Usage

```
cwr_geos_version()
```

Value

A version string.

Examples

```
cwr_geos_version()
```

Index

`cvr_area (cvr_coverage)`, [2](#)
`cvr_coverage`, [2](#)
`cvr_gap (cvr_coverage)`, [2](#)
`cvr_geos_version`, [4](#)
`cvr_invalid_edges (cvr_coverage)`, [2](#)
`cvr_is_valid (cvr_coverage)`, [2](#)
`cvr_n_coord (cvr_coverage)`, [2](#)
`cvr_simplify (cvr_coverage)`, [2](#)
`cvr_simplify_each (cvr_coverage)`, [2](#)
`cvr_union (cvr_coverage)`, [2](#)