

rgeos

spatial geometry predicates and topology operations in R

Colin Rundel (Duke University)
Roger Bivand (Norwegian School of Economics)
Edzer Pebesma (University of Münster)

June 14, 2012

What is GEOS?

- Implements all the OpenGIS Consortium's Simple Feature Access for SQL specification
- C++ port of the Java Topology Suite
- Available under the LGPL
- Geometry engine behind open source project like PostGIS, SpatiaLite, QGIS, ...

What is rgeos?

- Brings the functionality of GEOS to R
- Developed in collaboration with Roger Bivand and Edzer Pebesma
- Official GSoC project in Summer 2010
- Implements the majority of the v1.6.2 C API (GEOS 3.2.2 and above)
- Written to integrate with R spatial packages (sp, maptools, etc.)
- Available on CRAN

OGC Simple Feature Access for SQL

Common standard for the representation of Geospatial data

- Specifies common 2d geometric data types:
 - Point / MultiPoint
 - LineString / MultiLineString
 - Polygon / MultiPolygon
 - LinearRing
 - GeometryCollection
- Also specifies attributes, methods, and assertions for these geometries
- Common exchange formats: Well-known text, Well-known binary
- Standard - <http://www.opengeospatial.org/standards/sfs>

sp classes are very similar but differ in several important ways from SFS data types

- No native support for GeometryCollections or LinearRings (SpatialRings and SpatialCollections added in rgeos)
- Translation / implementation ambiguities
 - Should a SpatialPolygons object be a single MultiPolygon or a collection of polygons to iterate over?
- Differences in Polygon implementation

Topology Operations

Boolean

- gDifference
- gIntersection
- gSymdifference
- gUnion

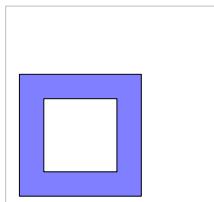
Constructive

- gBoundary
- gBuffer
- gCentroid
- gConvexHull
- gEnvelope
- gLineMerge
- gPointOnSurface
- gPolygonize
- gSimplify
- gUnionCascaded / gUnionUnary

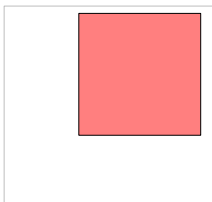
Metric

- gArea
- gDistance
- gLength

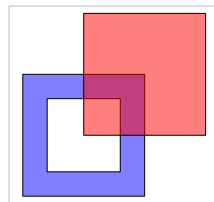
Topology - Boolean



X

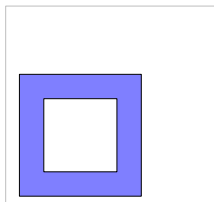


Y

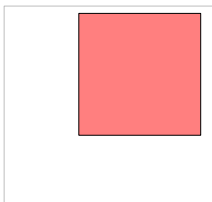


X and Y

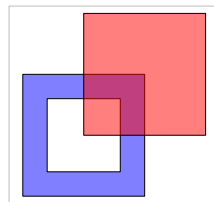
Topology - Boolean



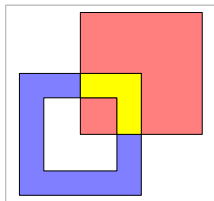
X



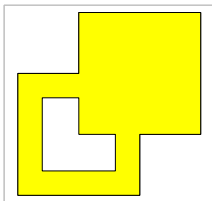
Y



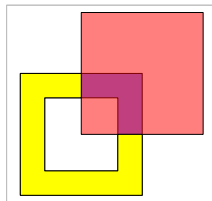
X and Y



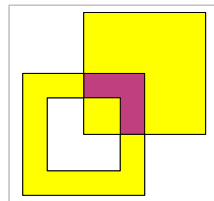
gIntersection



gUnion



gDifference



gSymdifference

The original impetus for `rgeos` was the need to replace Roger Peng's `gpclib` package which implemented these boolean operations for polygons.

- `gpclib` is a wrapper around Alan Murta's General Polygon Clipper library
- The GPC library has a restrictive license (free only for non-commercial use)
- The GPC library is limited to polygon clipping (boolean operations)
- `gpclib` uses `gpc.poly` S4 classes

`rgeos` has functionality to transparently replace `gpclib` for purposes of backwards compatibility

Example

```
library(maptools)
nc = readShapePoly(
  system.file("shapes/sids.shp", package="maptools")[1],
  proj4string=CRS("+proj=longlat +datum=NAD27")
)
pts = coordinates(nc)
ID = cut(pts[,1], quantile(pts[,1]), include.lowest=TRUE)
nc_regions = gUnionCascaded(nc, ID)
```



Example

```
library(maptools)
nc = readShapePoly(
  system.file("shapes/sids.shp", package="maptools")[1],
  proj4string=CRS("+proj=longlat +datum=NAD27")
)
pts = coordinates(nc)
ID = cut(pts[,1], quantile(pts[,1]), include.lowest=TRUE)
nc_regions = gUnionCascaded(nc, ID)
```



Spatial Predicates

Unary

- `gIsEmpty`
- `gIsRing`
- `gIsSimple`
- `gIsValid`

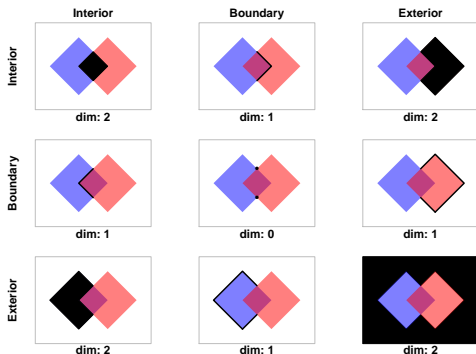
Binary

- `gContains`
- `gContainsProperly`
- `gCovers`
- `gCoveredBy`
- `gCrosses`
- `gDisjoint`
- `gEquals`
- `gEqualsExact`
- `gOverlaps`
- `gRelate`
- `gTouches`
- `gWithin`
- `gWithinDistance`

Package Documentation and JTS Technical Specifications cover the specifics of these functions (<http://bit.ly/L1uifo>)

Binary predicates are based on the Dimensionally Extended 9-Intersection Model

- Reports the dimensionality of the intersection of the interiors, boundaries, and exteriors
- Possible values: 0, 1, 2, F, T, *



⇒ “212101212”

Binary predicates are based on the Dimensionally Extended 9-Intersection Model

- Reports the dimensionality of the intersection of the interiors, boundaries, and exteriors
- Possible values: 0, 1, 2, F, T, *

DE-9IM Pattern	Function	f(x,y)
FF*FF****	gDisjoint	false
FT*****	gTouches	false
T*T***T**	gCrosses	true
TF*F*****	gWithin	false
T*T***T**	gOverlaps	true

For a more details - <http://bit.ly/Kxm6iz> or <http://bit.ly/L1uifo>

Spatial Predicates and Prepared Geometries

Common usage pattern with spatial predicates will involve a single geometry being compared to a series of test geometries.

- Initial Geometry is reused for each subsequent predicate
- Many of the underlying calculations / data structures can be cached
- In some cases evaluation steps can be skipped
- Potentially huge performance gains for minimal overhead (on by default when possible)
- This is an area of recent development activity for GEOS:
 - CAPI v1.6.2 (GEOS v3.2.2) supports: Contains, ContainsProperly, Covers, Intersects
 - CAPI v1.7.0 (GEOS v3.3+) added support for: CoveredBy, Crosses, Disjoint, Overlaps, Touches, Within

Prepared Geometry Performance

Example

```
library(maptools)

data(wrld_simpl)
US = wrld_simpl[wrld_simpl@data$NAME == "United States",]

gt = GridTopology(c(-180,-90),c(0.5,0.5),c(720,360))
grid = SpatialGrid(gt)
sp = as(grid, "SpatialPoints")
proj4string(sp) = proj4string(US)

system.time(gIntersects(US,sp,byid=TRUE,prepared=TRUE))
system.time(gIntersects(US,sp,byid=TRUE,prepared=FALSE))
```


Prepared Geometry Performance

Example

```
library(maptools)

data(wrld_simpl)
US = wrld_simpl[wrld_simpl@data$NAME == "United States",]

gt = GridTopology(c(-180,-90),c(0.5,0.5),c(720,360))
grid = SpatialGrid(gt)
sp = as(grid, "SpatialPoints")
proj4string(sp) = proj4string(US)

system.time(gIntersects(US,sp,byid=TRUE,prepared=TRUE))
system.time(gIntersects(US,sp,byid=TRUE,prepared=FALSE))
```

prepared=TRUE

user	system	elapsed
0.220	0.016	0.285

prepared=FALSE

user	system	elapsed
244.851	0.004	244.895

Future Work

- Complete update to GEOS C API 1.7.0 while preserving backwards compatibility
- Refine usability of STRtree spatial indexes
- Add support for WKB
- Clean up documentation / Package vignette
- Better handling of Spatial*DataFrames

Questions, Comments, Bugs?

email : rundel@gmail.com

mailing list : R-sig-Geo

r-forge : <http://r-forge.r-project.org/projects/rgeos>

github : <http://github.com/rundel/rgeos>

presentation : <http://github.com/rundel/UseR2012>