Spatial modelling with the R–GRASS Interface

Rainer M. Krug

Centre of Excellence for Invasion Biology, Stellenbosch University

The R User Conference 2011 August 16-18 2011, University of Warwick, Coventry, UK
Outline

1. Case Study

2. Usage Scenarios
   - GRASS as spatial DB
   - R as script engine
   - both combined
Why do we worry about aliens?
Fire Module

Begin

Determine fuel model → Determine ros → Determine ignition risk

Determine ignition points

Calculate temporal and spatial fire spread

Identify cells to be burned

Burn cells

End

Different scenarios

1. GRASS as DB for spatial data
2. R as scripting language
3. Combination of both
GRASS as spatial DB

- Initialisation
- Read spatial data
- Do calculations
- Write spatial data
- Write non-spatial data
Requirements

- Connect to spatial data DB
- read spatial data
- write spatial data

Packages

- spgrass6
- rgdal
- SQLiteMap
- RSAGA
Examples I

Initialisation GRASS

```r
initGRASS(
    gisBase = parms(ASM)$grassPATH,
    home = tempdir(),
    SG = region,
    gisDbase = paste(getwd(), "/../", sep=""),
    location = "grass",
    mapset = "AlienSpreadSim",
    override = TRUE
)
```
```r
readRAST6 <- function(
  ...,
  ignore.stderr=!Debug,
  useGDAL=gdalGRASSenabled
) {
  oldWarn <- options()$warn
  options(warn=-1)
  result <- spgrass6::readRAST6(
    ...,
    ignore.stderr=ignore.stderr,
    useGDAL=useGDAL,
    plugin=FALSE
  )
  options(warn=oldWarn)
  return(result)
}
```
Improvements

- making connection to GRASS easier and more “transparent” for user
- session wide options for reading / writing spatial data spatial commands — use options?
- “Native” interface R ↔ GRASS — direct link
  - primary spatial back-end for storage
  - a “spatial dbi package” with different back-ends in other packages — mainly connect, read, write, delete and some query functions
  - switch between different connections to spatial sources
Improvements

- making connection to GRASS easier and more “transparent” for user
- session wide options for reading / writing spatial data spatial commands — use options?
- "Native” interface R ⇔ GRASS — direct link
- primary spatial back-end for storage
  - a “spatial dbi package” with different back-ends in other packages — mainly connect, read, write, delete and some query functions
  - switch between different connections to spatial sources
Improvements

- making connection to GRASS easier and more “transparent” for user
- session wide options for reading / writing spatial data spatial commands — use options?
- “Native” interface R $\leftrightarrow$ GRASS — direct link
- primary spatial back-end for storage
- a “spatial dbi package” with different back-ends in other packages — mainly connect, read, write, delete and some query functions
- switch between different connections to spatial sources
Improvements

- making connection to GRASS easier and more “transparent” for user
- session wide options for reading / writing spatial data spatial commands — use options?
- “Native” interface $R \leftrightarrow GRASS$ — direct link
- primary spatial back-end for storage
- a “spatial dbi package” with different back-ends in other packages — mainly connect, read, write, delete and some query functions
- switch between different connections to spatial sources
Mahongo
R as script engine

initialisation
request calculations
do calculations
done
Requirements

- Connect to GIS
- Execute commands in GIS

Packages

- spgrass6
- RSAGA
- SQLiteMap
Example

access GRASS functions

```r
execGRASS(
    cmd = "r.spread",
    flags = c("o"),
    parameters = list(
        max = "ros.2006.max",
        dir = "ros.2006.maxdir",
        base = "ros.2006.base",
        start = "ignition_2006_Points",
        output = "SpreadTime_2006"
    ),
    ignore.stderr = !Debug
)
```
Improvements

- implementation of r.mapcalc (not relevant any more for GRASS 7?)
- session wide options for executing GRASS commands — use options?
- use ... as an alternative way of passing parameter to GRASS function?
- primary GIS backend for analysis
Improvements

- Implementation of `r.mapcalc` (not relevant any more for GRASS 7?)
- Session wide options for executing GRASS commands — use options?
- Use ... as an alternative way of passing parameter to GRASS function?
- Primary GIS backend for analysis
both combined

R

initialisation

read spatial data

do calculations

write spatial data

write non-spatial data

request calculations

do calculations

done

read spatial data

do calculations

R

GRASS

do calculations

GRASS
Requirements

- Frequent data exchange $R \leftrightarrow$ GRASS
- Working with a MASK
- Easily portable to other computer (PCs, hpc, ...)

Improvements

- Increased speed in reading / writing
- Parallelization of routines (map arithmetic)
Requirements

- Frequent data exchange \( R \leftrightarrow \text{GRASS} \)
- Working with a MASK
- Easily portable to other computer (PCs, hpc, …)

Improvements

- Increased speed in reading / writing
- Parallelization of routines (map arithmetic)
Conclusions

- Good infrastructure for GIS work
- Room for improvements

We should not reinvent the wheel, but rather make it run smoother
Conclusions

- Good infrastructure for GIS work
- Room for improvements

We should not reinvent the wheel, but rather make it run smoother
Acknowledgements

Funding provided by

- Global Environmental Facility (GEF) through the C.A.P.E. Program and CapeNature
- Table Mountain Fund & WWF–SA

All developers — great work!
All participants on R and GRASS mailing lists (Roger Bivand)