

GridDER: Grid Detection and Evaluation in R

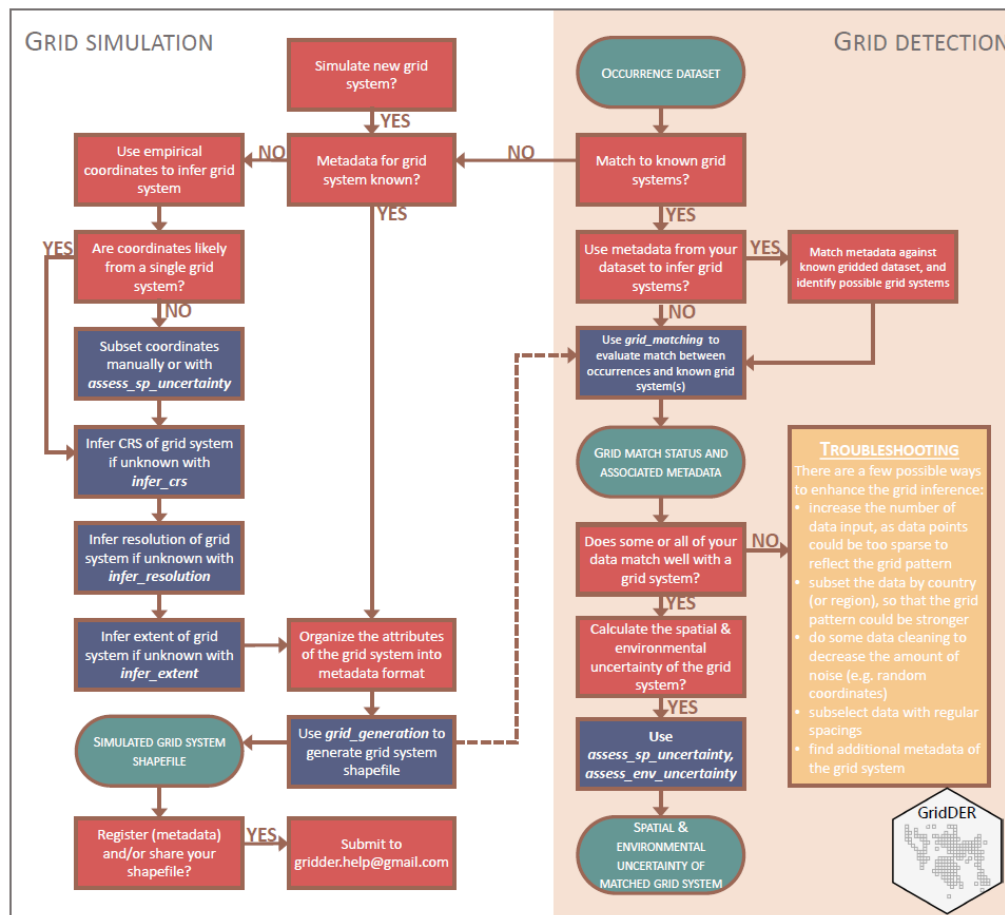
Xiao Feng , Tainá Rocha , Hanna Thammavong , Rima Tulaiha , Xin Chen , Yingying Xie , Daniel Park

Abstract Observações e coleções de organismos formam a base da nossa compreensão sobre a biodiversidade da Terra e são um recurso indispensável para inúmeros estudos. A informação geográfica é fundamental, servindo de elo entre os organismos e os ambientes em que residem. No entanto, a informação geográfica associada a esses registros é muitas vezes imprecisa, limitando a sua eficácia para pesquisas. Nessa linha, foram desenvolvidas várias soluções para identificar coordenadas errôneas e georreferenciados a centroides de países, estados ou municípios. Outra fonte de imprecisões proeminente, mas menos discutida e documentada, surge devido ao uso grades em muitas regiões do mundo. Por isso, desenvolvemos o pacote R GridDER, para identificar registros de biodiversidade provenientes de sistemas de grades. Nossa ferramenta também estima o grau de heterogeneidade ambiental associado a essas grades, permitindo que os usuários tomem decisões informadas sobre como usar esses dados de ocorrência em estudos de mudanças globais. Portanto, o GridDER pode servir como uma ferramenta não apenas para triagem de dados georreferenciados em grade, mas para quantificar as incertezas geográficas e ambientais associadas a esses registros. Nesta edição do LantInR, as principais funções do pacote serão mostrados através de exemplos práticos.

Palabras clave: Pacote R - Biodiversidade - Dados de ocorrência

```
##  
## Found the following errors:  
## * Author 1: missing email, country, and affiliation  
## * Author 3: missing email, country, and affiliation  
## * Author 4: missing email, country, and affiliation  
## * Author 5: missing email, country, and affiliation  
## * Author 6: missing email, country, and affiliation  
## * Author 7: missing email, country, and affiliation
```

Fluxo de trabalho do pacote



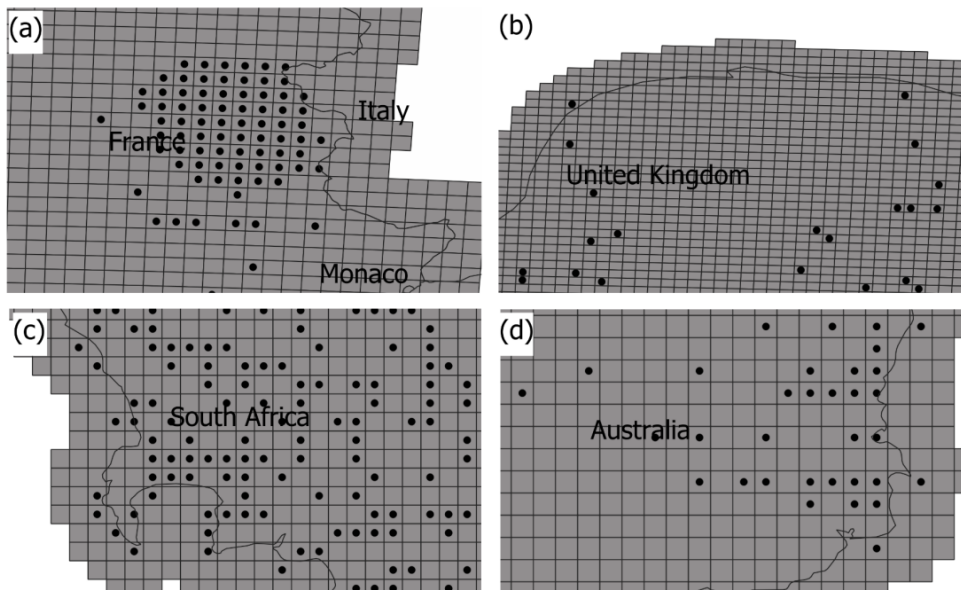
Instalação

```
# remotes::install_github("BiogeographyLab/gridder")
```

Exemplos práticos

Exemplos de sistemas de grade

Exemplos de quatro sistemas de rede usados na França (a), Reino Unido (b), África do Sul (c) e Austrália (d). Os quatro sistemas de grade têm diferentes soluções espaciais – 10km para (a), 1km para (b), 5 minutos de arco para (c) e 6 minutos de arco para (d). Os pontos pretos representam o biológico coleções atribuídas ao centroide dos sistemas de grade correspondentes. Todos os mapas construídos usando WGS84.



Veja alguns dos dados do pacote

```
library(gridder)
```

```
## Warning: replacing previous import 'data.table::last' by 'dplyr::last' when
## loading 'gridder'

## Warning: replacing previous import 'data.table::first' by 'dplyr::first' when
## loading 'gridder'

## Warning: replacing previous import 'data.table::between' by 'dplyr::between'
## when loading 'gridder'

## Warning: replacing previous import 'dplyr::count' by 'kit::count' when loading
## 'gridder'

## Warning: replacing previous import 'dplyr::union' by 'raster::union' when
## loading 'gridder'

## Warning: replacing previous import 'data.table::shift' by 'raster::shift' when
## loading 'gridder'

## Warning: replacing previous import 'dplyr::select' by 'raster::select' when
## loading 'gridder'

## Warning: replacing previous import 'kit::getData' by 'raster::getData' when
## loading 'gridder'

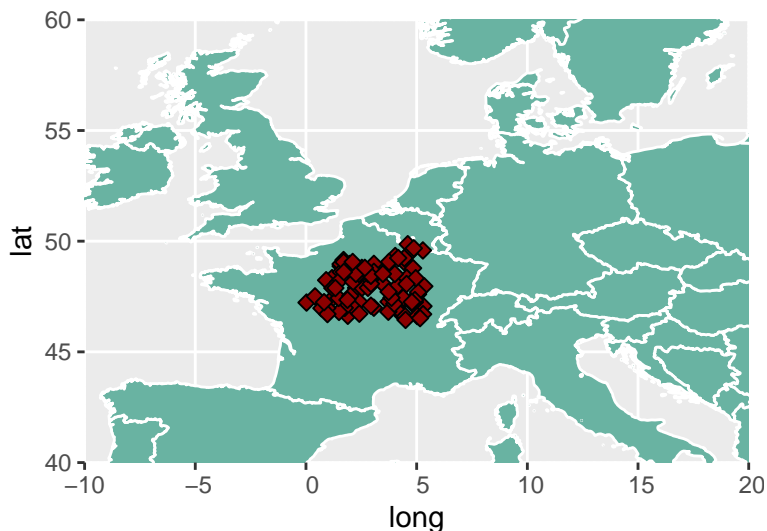
## Warning: replacing previous import 'dplyr::intersect' by 'raster::intersect'
## when loading 'gridder'

## Registered S3 method overwritten by 'geojsonlint':
##   method      from
##   print.location dplyr
```

```
library(ggplot2)
library(broom)

data("occs_unique") # dataframe de ocorrências disponibilizada no pacote
data("ne_10m_admin_0_countries") # shapefile países disponibilizado no pacote

spdf_fortified = broom::tidy(ne_10m_admin_0_countries, region = "ADMIN")
ggplot2::ggplot() +
  geom_polygon(data = spdf_fortified,
              aes( x = long, y = lat, group = group),
              fill="#69b3a2", color="white") +
  geom_point(data = occs_unique,
            aes(x = decimalLongitude, y = decimalLatitude),
            size = 2,
            shape = 23, fill = "darkred") +
  coord_sf(xlim = c(-10, 20), ylim = c(40, 60), expand = FALSE)
```



Inferir o CRS (sistema de referência de coordenadas)

```
# Pode demorar bastante a depender do número de núcleos disponíveis na máquina (argumento cup_num)

# results_crs = infer_crs(occ_path = occs_unique, cup_num = 2)

# Vamos ler o resultado que foi salvo em .rds

results_crs = readRDS("results_crs.rds")
print(results_crs$selected[1:10, c("code", "note")])
```

```
##      code      note
## 1    2154    truth
```

```
## 1336 2154 RGF93 / Lambert-93
## 6350 5698 RGF93 / Lambert-93 + NGF-IGN69 height
## 6351 5699 RGF93 / Lambert-93 + NGF-IGN78 height
## 4355 3947 RGF93 / CC47
## 4354 3946 RGF93 / CC46
## 2944 30731 Nord Sahara 1959 / UTM zone 31N
## 3515 32431 WGS 72BE / UTM zone 31N
## 4353 3945 RGF93 / CC45
## 4352 3944 RGF93 / CC44
```

Inferir a resolução espacial

```
# Ler como objeto espacial
```

```
input_occ = gridder::load_occ(occs_unique)
```

```
## Loading required package: sp
```

```
##
```

```
## Attaching package: 'data.table'
```

```
## The following object is masked from 'package:raster':
```

```
##
```

```
## shift
```

```
input_occ_prj = sp::spTransform(input_occ,crs(paste0("+init=epsg:", "2154"))) # Já sabemos o CRS
```

```
result_res = gridder::infer_resolution(input_coord=input_occ_prj@coords,flag_unit="meter")
print(result_res$res_x)
```

```
## [1] 40000
```

Inferir a extensão espacial

```
result_ext = gridder::infer_extent(method="crs_extent",
                                   crs_grid=results_crs$selected$code[1],
                                   flag_adjust_by_res=TRUE,
                                   res_x=result_res$res_x,
                                   res_y=result_res$res_y)
```

```
print(result_ext)
```

```
## [1] -400000 6080000 1200000 7180000
```

Simulando um sistema de grade baseado em metadados

```
## [1] 40000
```

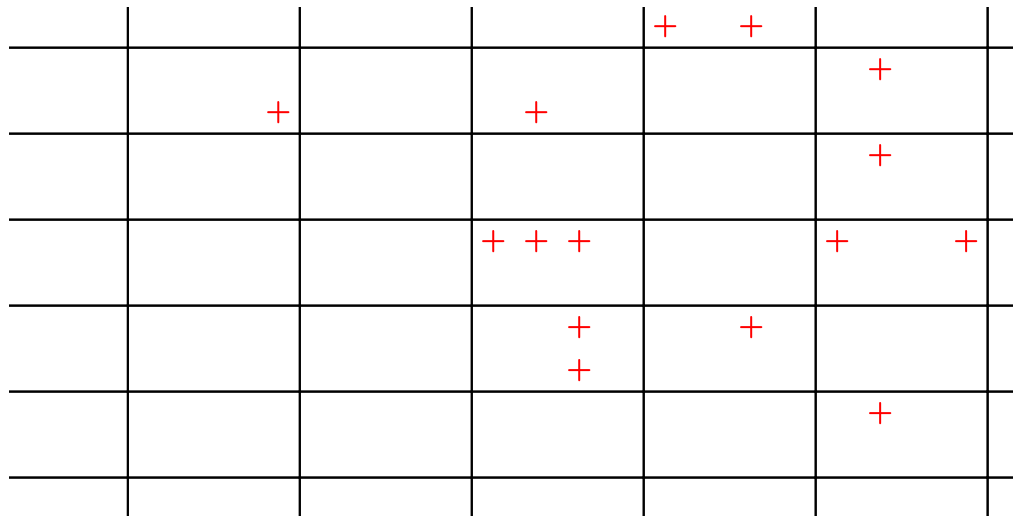
```
## [1] 20000
```

```
## [1] "m"
```

```
## [1] -400000 6280000 1200000 7180000
```

Checando o sistema de grade simulada com as ocorrências

```
plot(simulated_grid,
     xlim=c(extent(input_occ_prj)[1],extent(input_occ_prj)[1]+110000),
     ylim=c(extent(input_occ_prj)[3],extent(input_occ_prj)[3]+110000))
plot(input_occ_prj,add=T,col="red")
```



Calcular a variação ambiental do sistema de grade

```
# Grid Id 9, exemplo de um sistema de grade disponibilizado no pacote
```

```
data("grid_ID_9")
```

```
## Usamos algumas funções baseadas no pacote rgee. Por isso começamos pelo função ee_Initialize
```

```
rgee::ee_Initialize()
```

```
## -- rgee 1.1.4 ----- earthengine-api 0.1.321 --
## v user: not_defined
## v Initializing Google Earth Engine: v Initializing Google Earth Engine: DONE!
## v Earth Engine account: users/taína013
## -----
```

```
nasadem = rgee::ee$Image('NASA/NASADEM_HGT/001')$select('elevation') # Global elevation data in 30 meters. We do not plot
```

```
grid = grid_ID_9 |>
  rgee::sf_as_ee()
```

```
## Registered S3 method overwritten by 'geojsonsf':
##   method      from
##   print.geojson geojson

std_dev = gridder::assess_env_uncertainty(x= nasadem, y= grid)
```

```
## [1] "Extracting information [0/336]..."
## Number of features: Calculating ...Number of features: 336
```

```
std_dev
```

```
## Simple feature collection with 336 features and 12 fields
## Geometry type: POLYGON
## Dimension:      XY
## Bounding box:  xmin: 38.1667 ymin: 55.0833 xmax: 43 ymax: 56.8333
## Geodetic CRS:  WGS 84
## First 10 features:
##   Name alttdMd begin drwOrdr   dscripnt end extrude icon tesslft timestp
## 1   ?1      NA   NA      NA ??????? ?1   NA      0   NA      -1      NA
## 2   ?2      NA   NA      NA ??????? ?2   NA      0   NA      -1      NA
## 3   ?3      NA   NA      NA ??????? ?3   NA      0   NA      -1      NA
## 4   ?1      NA   NA      NA ??????? ?1   NA      0   NA      -1      NA
## 5   ?2      NA   NA      NA ??????? ?2   NA      0   NA      -1      NA
## 6   ?3      NA   NA      NA ??????? ?3   NA      0   NA      -1      NA
## 7   ?4      NA   NA      NA ??????? ?4   NA      0   NA      -1      NA
## 8   ?1      NA   NA      NA ??????? ?1   NA      0   NA      -1      NA
## 9   ?2      NA   NA      NA ??????? ?2   NA      0   NA      -1      NA
## 10  ?3      NA   NA      NA ??????? ?3   NA      0   NA      -1      NA
##   visblty elevation geometry
## 1      -1 12.579737 POLYGON ((39.5 56.75, 39.66...
## 2      -1  6.765410 POLYGON ((39.6667 56.75, 39...
## 3      -1  6.164659 POLYGON ((39.8333 56.7083, ...
## 4      -1 13.234414 POLYGON ((39.3333 56.6667, ...
## 5      -1  7.550874 POLYGON ((39.5 56.6667, 39...
## 6      -1 14.769370 POLYGON ((39.6667 56.6667, ...
## 7      -1  8.457091 POLYGON ((39.8333 56.625, 4...
## 8      -1 38.233946 POLYGON ((38.3333 56.5833, ...
## 9      -1 19.173704 POLYGON ((38.5 56.5833, 38...
## 10     -1 20.865249 POLYGON ((39.3333 56.5833, ...
```

Visualizando a variação ambiental no espaço geográfico

```
library(ggplot2)
ggplot(data = grid_ID_9) +
  geom_sf(aes(fill = std_dev$elevation))+
  scale_fill_viridis_c(option = "plasma", trans = "sqrt")
```

