

# Package 'GridR'

Paulina Lewińska

# Ogólnie informacje o bibliotece Package 'GridR'

- Bibliotek grid została stworzona aby wykorzystywać funkcje programu R ( oraz wprowadzone do programu R) do tworzenia odwzorowań typu grid oraz siatek punktów. Dodatkowo pozwala ona użytkownikom na wymianę funkcji oraz ich współczynników między sobą.
- Biblioteka jest głównie modułem obliczeniowym oraz może służyć do wymiany oraz ochrony danych. Posiada kilka podstawowych opcji które zostaną dalej omówione

# Omówienie podstawowych opcji

- ▮ **grid.share** – funkcja służy do wymiany danych, funkcji oraz współczynników między użytkownikami
- ▮ **grid.apply** – wykorzystuje wybraną funkcję do obróbki wskazanego pliku, grupy punktów
- ▮ **grid.check** – sprawdza czy wybrana do tworzenia grida funkcja korzysta ze wszystkich zadanych punktów. W razie wykrycia nieprawidłowości zwraca informacje o nie wykorzystanych punktach
- ▮ **grid.compileScript** – kompilatorach skryptu gridowania. Kompiluje – sprawdza napisany skrypt wykonawczy gridu. Zachowuje i sprawdza wszystkie wprowadzone linijki kodu w sesji. W razie wykrycia błędu pokazuje informacje o wykryciu go.
- ▮ **grid.consistency** – funkcja działa jak nieco uproszczony kompilator. Szuka błędów w zmiennych wykorzystanych do gridowania, sprawdza czy istnieją odwołania do nie wykorzystanych plików czy istnieją odwołania do nie istniejących plików. Kasuje stare pliki tmp.

# Omówienie podstawowych opcji

- ▣ **grid.disableSharing** – uniemożliwia dzielenie się przez użytkowników funkcjami
- ▣ **grid.exit** – usuwa wszelkie utworzone wcześniej pliki grid, usuwa historie ( wykonuje swoje funkcje przy każdym naciśnięciu klawisza ENTER)
- ▣ **grid.FTPDownload** – pobiera plik z url oraz zapamiętuje ścieżkę do niego
- ▣ **grid.init** – inicjalizuje pakiet GridR. Jeśli pliki nie są skonfigurowane konieczne jest dodanie dodatkowych parametrów. ( np. localTmpDir – ścieżka do miejsca przechowywania tymczasowych danych, remoteRPath – ścieżka do R jeśli jest konieczna)
- ▣ **grid.printJobs** – drukuje, wysyła do druku ( konwersji) wszystkie wykonywane aktualnie przez GridR zadania
- ▣ **grid.restartJob** – przywraca ostatnie zadanie

# Grid.check – sprawdza wykorzystanie plików, wskaźników i argumentów

```
grid.check(grid.input.Parameters.f,x=„”,varlist=c(10,100,1000),  
fName="C:\\Users\\Sintiara\\Desktop\\grid\\w1.txt", intern=FALSE)
```

```
[1] 10 100 1000
```

Argumenty

x do użytku wewnętrznego

varlist – wektor który zawiera argumenty konieczne do wykonania funkcji f

fName – funkcja wewnętrzna

intern=FALSE jeśli wszystkie argumenty są wykorzystywane albo wypisanie nie wykorzystanych

# grid.init – uruchomienie funkcji

```
a<-function(s){return(2*s)}  
#zdefiniowanie funkcji  
library("GridR")  
#uruchomienie biblioteki  
(service="local",debug=FALSE, localTmpDir="GridRTmp/")  
#inicjalizacja gridR  
grid.apply("x",a, 3, wait=TRUE)  
#uruchamia funkcje `a` z parametrem 3 i zapisuje wyniki do zmiennej X
```

# grid.apply

```
grid.apply(a,"grid.input.C:\\Users\\Sintiara\\Desktop\\grid\\gridd.txt",3,wait=FALSE)
```

Pierwszy argument funkcja odwołania

Drugi plik input

3 – współczynnik funkcji

Wait=FALSE – działanie w tle

# grid.apply

- Arguments
- `grid.input.Parameters.y` – miejsce danych
- `...` – parametry funkcji
- `wait if wait=TRUE` – działania w tle
- `Plots`- ewentualne wytworzenie obrazu działań
- `check if check=TRUE` sprawdza czy funkcja potrzebuje dodatkowych zmiennych
- `Batch` – dodatkowe parametry dla zmiennych wynikowych



# Przykład na przeprowadzenie działania funkcji na danym hoscie i zapisanie zmiennej do x

```
> #example to execute function a on a single remote host via ssh and save the result to x
> library("GridR")
> a<-function(s){Sys.sleep(s); return(s)}
> grid.init(service="local", localTmpDir="GridRTmp/")
> grid.apply("x",a, 3, wait=TRUE, check=TRUE)
Grid job finished, result written to variable x
> x
[1] 3
```

```
> #example for batch mode
> library("GridR")
> a<-function(s, p, q){return(s+p+q)}
> grid.init(service="local", localTmpDir="GridRTmp/")
> grid.apply("y",a, c(0,1,2),1, c(100,200,300), wait=TRUE, check=FALSE, batch=c(1,3))
Grid job finished, result written to variable y
> #Here all combinations of (0,1,2), 1 and (100,200,300) are executed as parameters of function y
> y
[[1]]
[1] 101

[[2]]
[1] 201

[[3]]
[1] 301

[[4]]
[1] 102

[[5]]
[1] 202

[[6]]
[1] 302

[[7]]
[1] 103

[[8]]
[1] 203

[[9]]
[1] 303
```

# grid.compileScript

**grid.compileScript** –  
kompilatorach skryptu  
gridowania. Kompiluje  
sprawdza napisany  
skrypt

wykonawczy gridu.  
Zachowuje i sprawdza  
wszystkie wprowadzone  
linijki kodu w sesji. W  
razie wykrycia błędu  
pokazuje informacje o  
wykryciu go.

```
> library("GridR")
> grid.init(service="local")
localTmpDir is not specified. Check config file and parameters.
[1] FALSE
> a=1
> 1
[1] 1
> grid.compileScript("test.R")
Script is written, but there is an Error in Line 11
```

# Opcje grid.lock, grid.unlock, oraz grid.unlockAll, grid.isLocked

- Grid.lock – blokuje zmienną do dalszych operacji grid
- Grid.unlock – odblokowuje zmienną wybraną
- Grid.unlockAll – odblokowuje wszystkie zmienne
- Grid.isLocked – informuje które zmienne są zablokowane a które nie

```
> library("GridR")
> grid.init(service="local")
> x=1
> y=2
> grid.lock("x")
> ## Not run: x

> library("GridR")
> grid.init(service="variablesharing", sharedDir="GridRTmp/share")
> x=2
Error in function (...) : object x has write lock from grid function
> grid.share("x")
Error in function (...) : object x has write lock from grid function
|

> library("GridR")
> grid.init(service="local")
localTmpDir is not specified. Check config file and parameters.
[1] FALSE
> x=1
Error in function (...) : object x has write lock from grid function
> grid.lock("x")
object x has lock> grid.unlock("x")

> library("GridR")
> grid.init(service="variablesharing")
> x=1
> library("GridR")
> grid.init(service="variablesharing")
> x=1
> y=2
> grid.lock("x")
> grid.isLocked("x")
[1] TRUE
> grid.isLocked("y")
[1] FALSE

> library("GridR")
> grid.init(service="local")
> x=1
> y=1
> grid.lock("x")
object x has lock> grid.lock("y")
> grid.unlockAll()
>
```

