Large-Scale Spatial Data Science with ExaGeoStat

Sameh Abdulah, Marc G. Genton, Yan Song

In collaboration w/Ying Sun, Hatem Ltaief, David E. Keyes

Spatio-Temporal Statistics and Data Science (stsds.kaust.edu.sa)
Statistics Program (stat.kaust.edu.sa)
King Abdullah University of Science and Technology

February 26, 27, 28, 2024

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

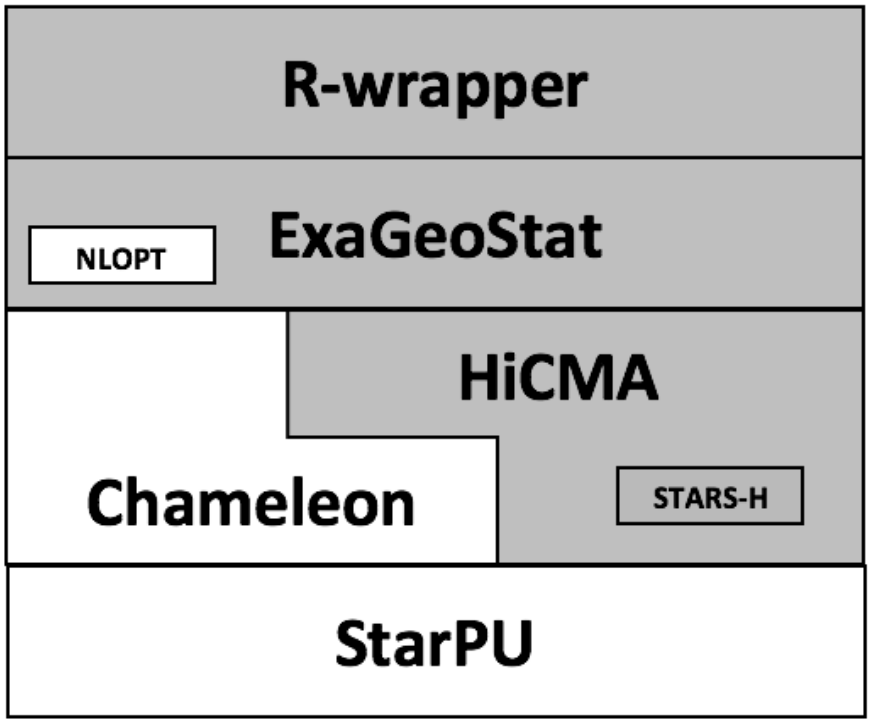
STATISTICS @KAUST

Part III

ExaGeoStatCPP Software in R

ExaGeoStatR

- ExaGeoStatR is a package for large-scale Geostatistics in R that supports parallel computation of the Gaussian maximum likelihood function on shared memory, GPU, and distributed memory systems



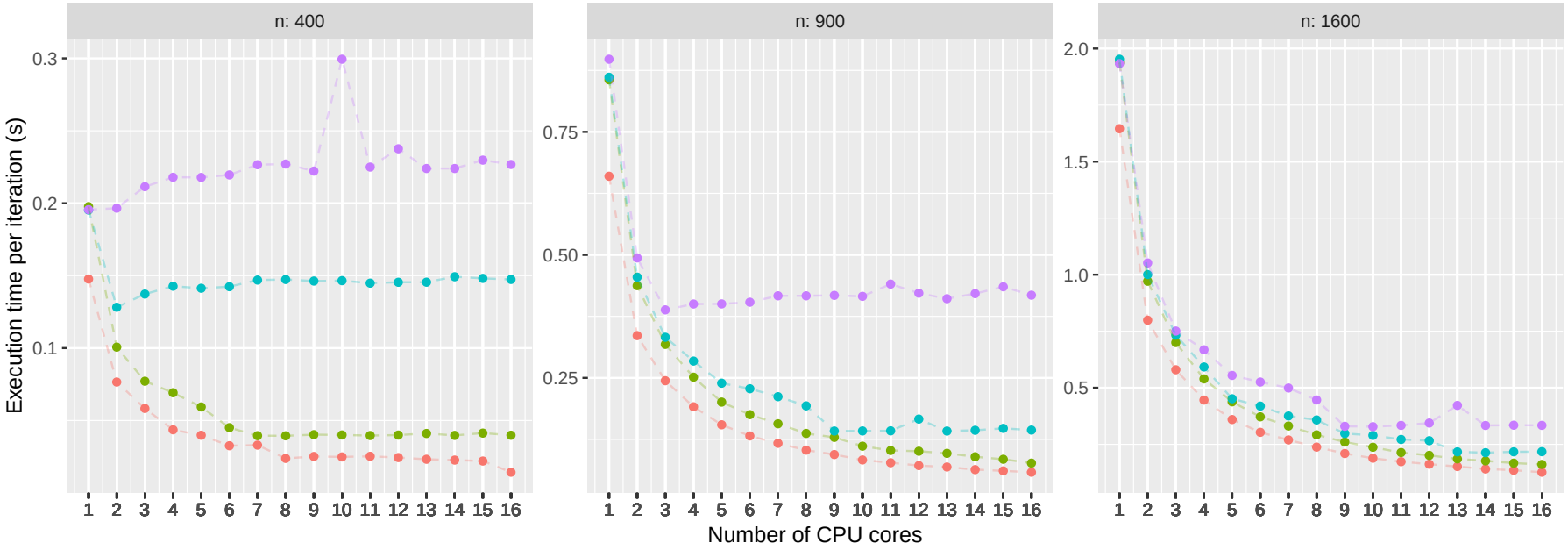
Existing Gaussian Likelihood Calculations in R packages

Package	geoR	fields	ExaGeoStatR
Function name	`likfit`	`MLESpatialProcess`	`exact_mle`
Mean	estimated	estimated	fixed as zero
Variance	estimated	estimated	estimated
Spatial Range	estimated	estimated	estimated
Smoothness	estimated	fixed	estimated
Default optimization method	`Nelder-Mead`	`BFGS`[1]	`BOBYQA`[2]

[1]. BFGS: Broyden-Fletcher-Goldfarb-Shanno. [2]. BOBYQA: bound optimization by quadratic approximation

ExaGeoStatR on Shared-memory System

- 16-core Intel Sandy Bridge Xeon E5-2650 Chip

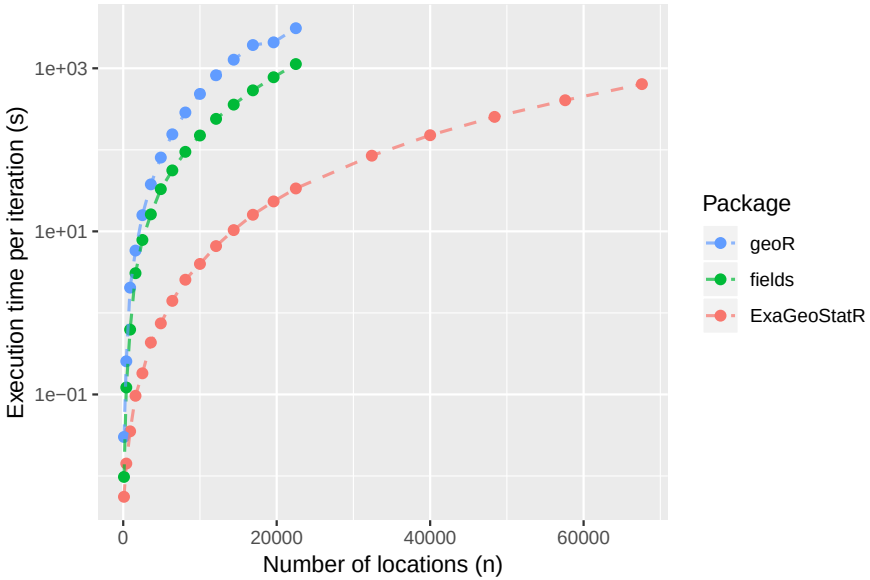


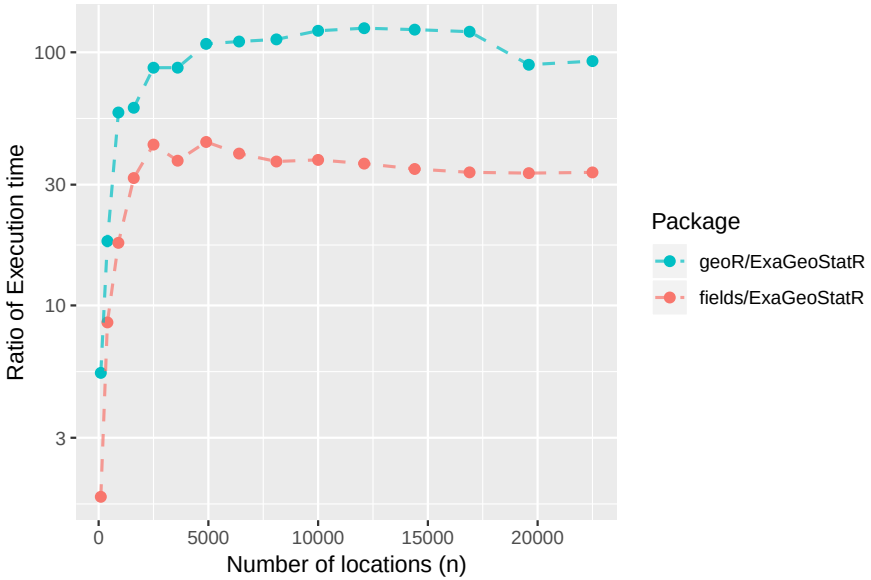
geoR - fields - ExaGeoStatR Comparison (Time)

- Average on 100 samples

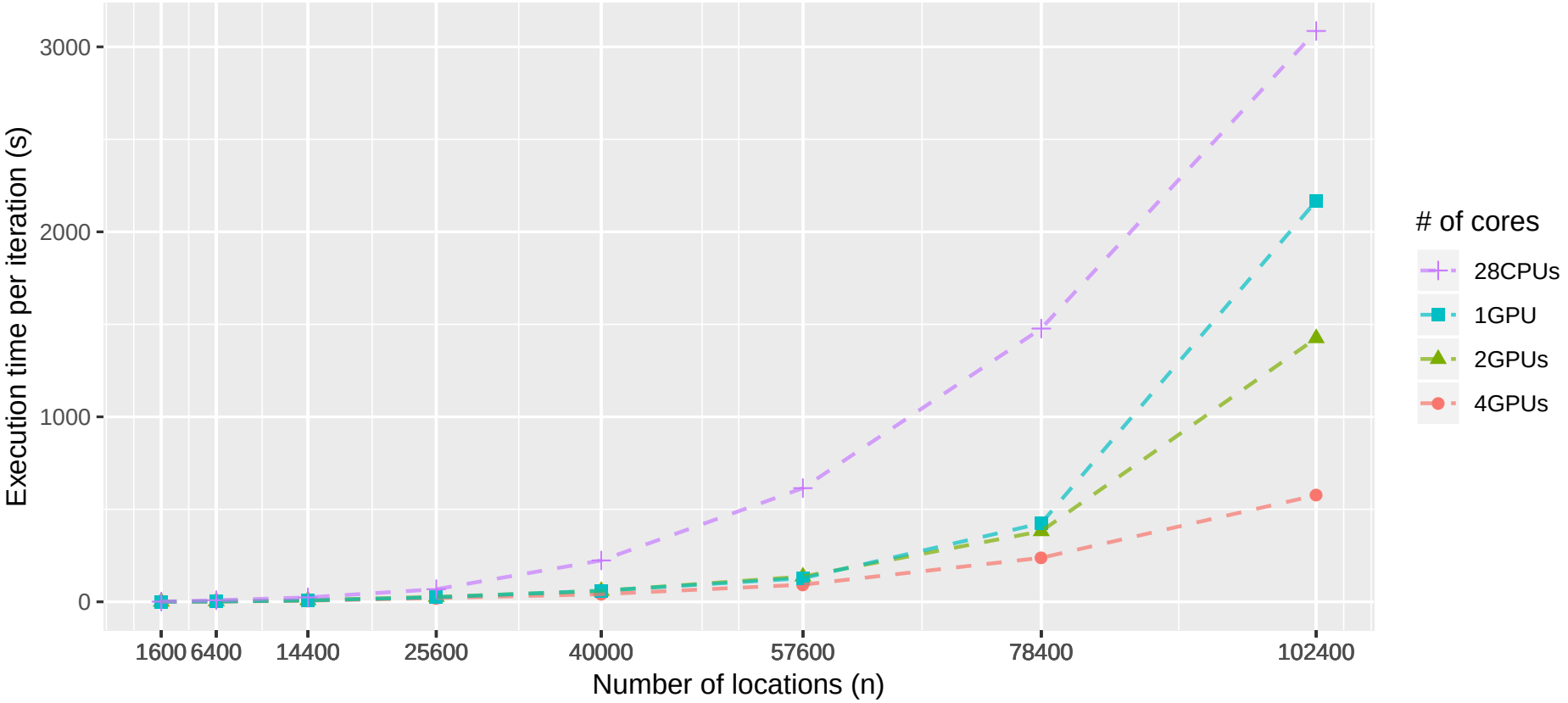
The average execution time per iteration (seconds)									
Package	**geoR**			**fields**			**ExaGeoStatR**		
$\nu=$ \ $\beta=$	0.03	0.1	0.3	0.03	0.1	0.3	0.03	0.1	0.3
0.5	1.39	1.49	1.47	0.75	0.97	0.99	0.10	0.12	0.12
1	1.35	1.49	1.56	0.66	0.90	0.90	0.09	0.13	0.13
2	1.34	1.56	1.57	0.67	0.91	0.93	0.09	0.13	0.13
The average number of iterations to reach the tolerance									
$\nu=$ \ $\beta=$	0.03	0.1	0.3	0.03	0.1	0.3	0.03	0.1	0.3
0.5	160	157	135	73	72	70	231	204	237
1	193	33	23	75	75	80	318	320	275
2	216	25	20	100	70	85	427	436	332

geoR - fields - ExaGeoStatR Comparison (Speedup)

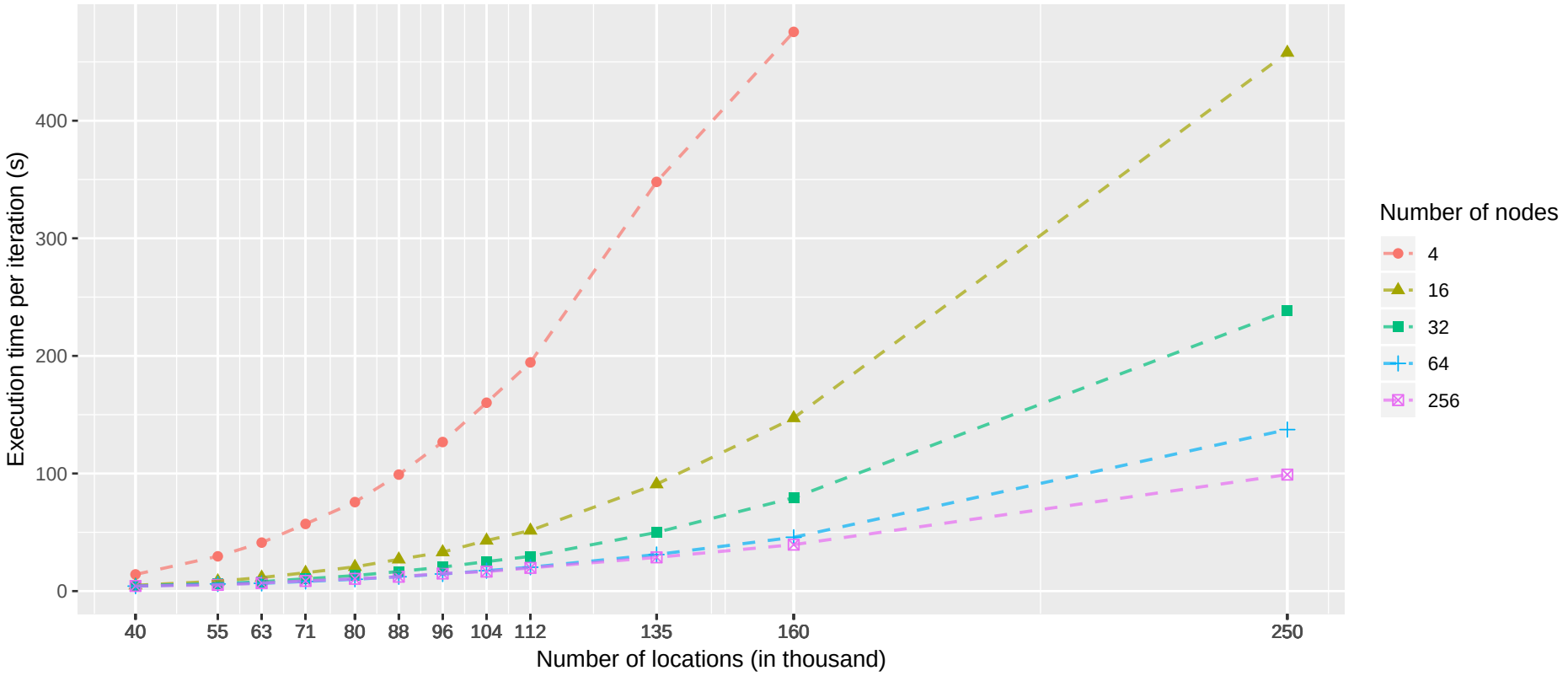




ExaGeoStatR Performance on Heterogeneous System (CPU/GPU)



ExaGeoStatR Performance on Distributed-Memory System (Shaheen-II)



Example (1)

```
### Example 1: Generate and model 2D spatial data using matern covariance without the nugget
# True parameter Theta=c(1,0.032579,1,0)
library("ExaGeoStatCPP")
ncores <- 39
ngpus <- 0
dts <- 360
lts <- 0
computation <- "exact"
hardware <- new(Hardware, computation, ncores, ngpus)



problem_size <- 10e3

config <- configurations_init(n=problem_size, kernel="univariate_matern_stationary", computation=computation, tile_size=c(dts,lts),
                              iTheta=c(1,0.032579,1), lb_ub=list(c(0.01,0.01,0.01),c(5,5,5)), mle_itr=1000, tol=9, prediction=c(0,0,0,0,0))

data_source <- new(Data, problem_size, "2D")

exageostat_data <- simulate_data(hardware=hardware, config=config, data=data_source)



model_data(hardware=hardware, config=config, data=exageostat_data)
```

Example (2)

```
### Example 2: Generate and model 3D spatial data using matern covariance without the nugget
library("ExaGeoStatCPP")
ncores <- 39
ngpus <- 0
dts <- 360
lts <- 0
computation <- "exact"
hardware <- new(Hardware, computation, ncores, ngpus)



problem_size <- 10e3

config <- configurations_init(n=problem_size, kernel="univariate_matern_stationary", computation=computation, tile_size=c(dts,lts),
                              iTheta=c(1,0.032579,1), lb_ub=list(c(0.01,0.01,0.01),c(5,5,5)), mle_itr=1000, tol=9, prediction=c(0,0,0,0,0))

data_source <- new(Data, problem_size, "3D")

exageostat_data <- simulate_data(hardware=hardware, config=config, data=data_source)



model_data(hardware=hardware, config=config, data=exageostat_data)
```

Example (3)

```
### Example 3: Generate and model 2D spatial data using matern covariance with the nugget
library("ExaGeoStatCPP")
ncores <- 39
ngpus <- 0
dts <- 360
lts <- 0
computation <- "exact"
hardware <- new(Hardware, computation, ncores, ngpus)



problem_size <- 10e3

config <- configurations_init(n=problem_size, kernel="UnivariateMaternNuggetsStationary", computation=computation, tile_size=c(dts,lts),
                              iTheta=c(1,0.032579,1, 0.2), lb_ub=list(c(0.01,0.01,0.01,0.01),c(5,5,5,1)), mle_itr=1000, tol=9, prediction=c(0,0,0,0,0))

data_source <- new(Data, problem_size, "2D")

exageostat_data <- simulate_data(hardware=hardware, config=config, data=data_source)



model_data(hardware=hardware, config=config, data=exageostat_data)
```

Example (4)

```
### Example 4: Generate 2D spatial data using matern covariance with the nugget but model it without nugget
library("ExaGeoStatCPP")
ncores <- 39
ngpus <- 0
dts <- 360
lts <- 0
computation <- "exact"
hardware <- new(Hardware, computation, ncores, ngpus)



problem_size <- 10e3

config <- configurations_init(n=problem_size, kernel="UnivariateMaternNuggetsStationary", computation=computation, tile_size=c(dts,lts),
                              iTheta=c(1,0.032579,1, 0.2), lb_ub=list(c(0.01,0.01,0.01,0.01),c(5,5,5,1)), mle_itr=1000, tol=9, prediction=c(0,0,0,0,0))

data_source <- new(Data, problem_size, "2D")

exageostat_data <- simulate_data(hardware=hardware, config=config, data=data_source)



config <- configurations_init(n=problem_size, kernel="univariate_matern_stationary", computation=computation, tile_size=c(dts,lts),
                             iTheta=c(1,0.032579,1), lb_ub=list(c(0.01,0.01,0.01),c(5,5,5)), mle_itr=1000, tol=9, prediction=c(0,0,0,0,0))

model_data(hardware=hardware, config=config, data=exageostat_data)
```

Example (5)

```
#!/bin/bash
#SBATCH --job-name=job_name
#SBATCH --output=output_file.txt
#SBATCH --partition=XXXX
#SBATCH --nodes=4
#SBATCH --ntasks=4
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=31
#SBATCH --time 00:30:00

srun Rscript Rtest.r
```

Example (6)

```
### Example 6: Play with ncores to show the advantage of increasing the number of cores in performance
library("ExaGeoStatCPP")
# ncores <- 39
ncores <- 19
# ncores <- 9
ngpus <- 0
dts <- 360
lts <- 0
computation <- "exact"
hardware <- new(Hardware, computation, ncores, ngpus)


problem_size <- 30e3

config <- configurations_init(n=problem_size, kernel="univariate_matern_stationary", computation=computation, tile_size=c(dts,lts),
                              iTheta=c(1,0.032579,1), lb_ub=list(c(0.01,0.01,0.01),c(5,5,5)), mle_itr=5, tol=9, prediction=c(0,0,0,0,0))

data_source <- new(Data, problem_size, "2D")

exageostat_data <- simulate_data(hardware=hardware, config=config, data=data_source)


model_data(hardware=hardware, config=config, data=exageostat_data)
```

Example (7)

```
### Example 7: Play with matrix size to show the scalability
library("ExaGeoStatCPP")
ncores <- 39
ngpus <- 0
dts <- 360
lts <- 0
computation <- "exact"
hardware <- new(Hardware, computation, ncores, ngpus)


# problem_size <- 30e3
problem_size <- 10e3

config <- configurations_init(n=problem_size, kernel="univariate_matern_stationary", computation=computation, tile_size=c(dts,lts),
                              iTheta=c(1,0.032579,1), lb_ub=list(c(0.01,0.01,0.01),c(5,5,5)), mle_itr=5, tol=9, prediction=c(0,0,0,0,0))

data_source <- new(Data, problem_size, "2D")

exageostat_data <- simulate_data(hardware=hardware, config=config, data=data_source)


model_data(hardware=hardware, config=config, data=exageostat_data)
~
```

Example (8)

```
### Example 8: Use GPU
library("ExaGeoStatCPP")
ncores <- 39
ngpus <-2
dts <- 960
lts <- 0
computation <- "exact"
hardware <- new(Hardware, computation, ncores, ngpus)


problem_size <- 50e3

config <- configurations_init(n=problem_size,cores_gpus=c(ncores, ngpus), kernel="univariate_matern_stationary", computation=computation, tile_size=c(dts,lts),
                              iTheta=c(1,0.032579,1), lb_ub=list(c(0.01,0.01,0.01),c(5,5,5)), mle_itr=5, tol=9, prediction=c(0,0,0,0,0))

data_source <- new(Data, problem_size, "2D")

exageostat_data <- simulate_data(hardware=hardware, config=config, data=data_source)


model_data(hardware=hardware, config=config, data=exageostat_data)
~
```

Thank You!

Questions?