OBANSoft

Integrated software for Bayesian statistics and high performance computing with R

useR!

The R User Conference 2011

University of Warwick

Manuel Quesada, Domingo Giménez, Asunción Martínez

Coventry (UK), 16 of July of 2011

Content

1. Introduction and motivation
2. Preliminary analysis of the problem
3. Application design
4. Performance and parallelization
5. Conclusions and future directions

What is the motivation of the project?

To fill the gap with respect to applications to Bayesian analysis of data with minimal prior information…

…eventually high performance computing applied to problems of Bayesian statistics.

- As a starting point we have developed the first version of the desktop application **OBANSoft** with:
  - A modular design to facilitate:
    - Future extension with new functionality.
    - Non dependence on the statistical model.
  - Try to include aspects of **technology integration, parallelism and transparency to the user (self-optimization)**.
    - The integration of different languages, tools and parallel libraries (OpenMP, MPI, CUDA…) **would** be done transparently to the end user, who only uses the graphics application that remains invariable.

Research Groups

- **UMU: Parallel Computing Group.**

Experience in the development and optimization of parallel code. Including self-optimization techniques and the application of parallel computing in various scientific fields.

- **UMH: Bayesian Statistic Group.**

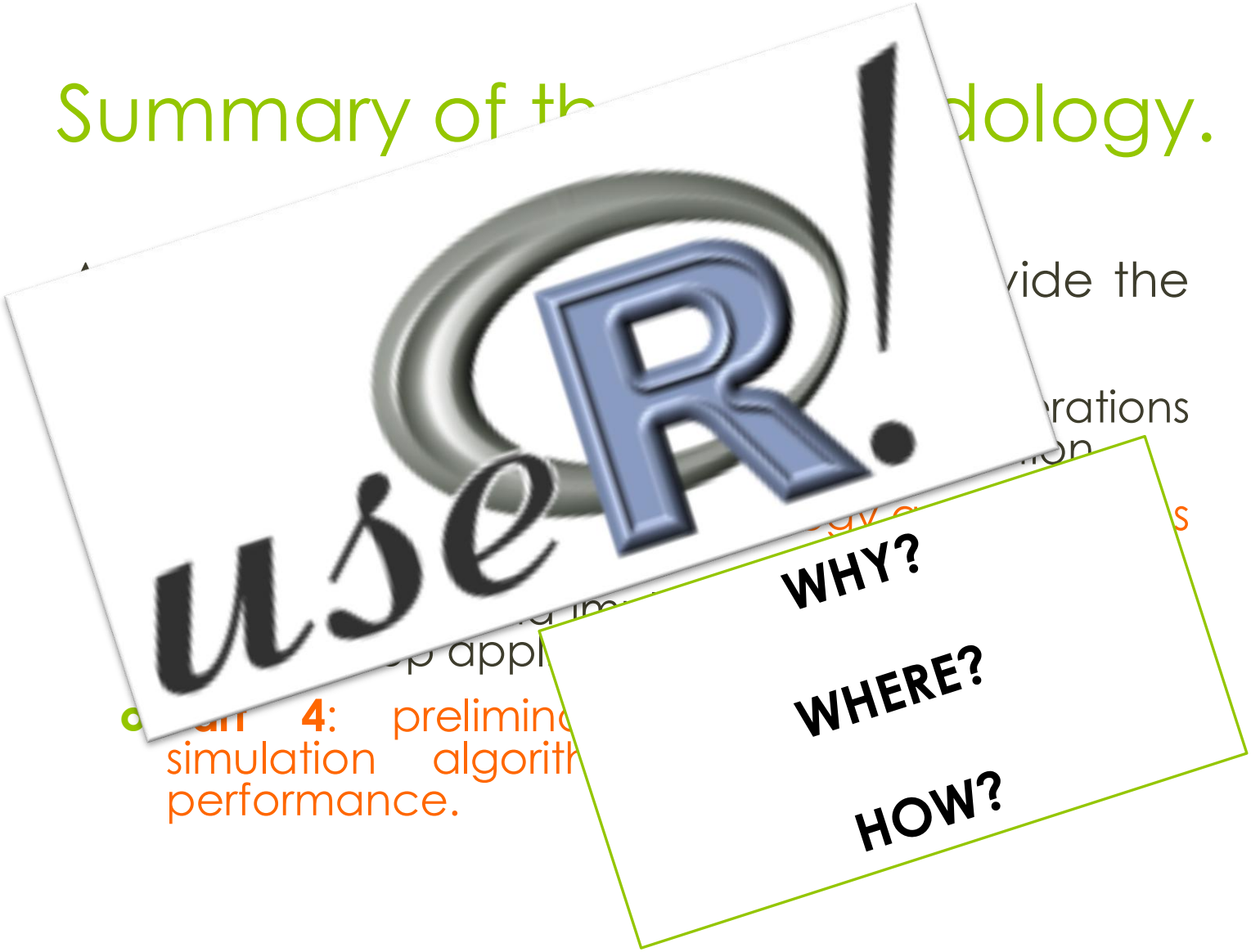
Experience in the development of simulation codes applicable to the resolution of Bayesian analysis in various fields.

Summary of the methodology.

Addressing various areas leads us to divide the methodology in 4 parts:

- **Part 1**: development of a Bayesian operations catalog to be supported by the application.
- **Part 2**: decision of the technology and resources to be used.
- **Part 3**: design and implementation of the library and desktop application.
- **Part 4**: preliminary parallelization of the simulation algorithms, and study of the performance.

Summary of th dology.



vide the

rations

p appl

4: prelimin
simulation algorith
performance.

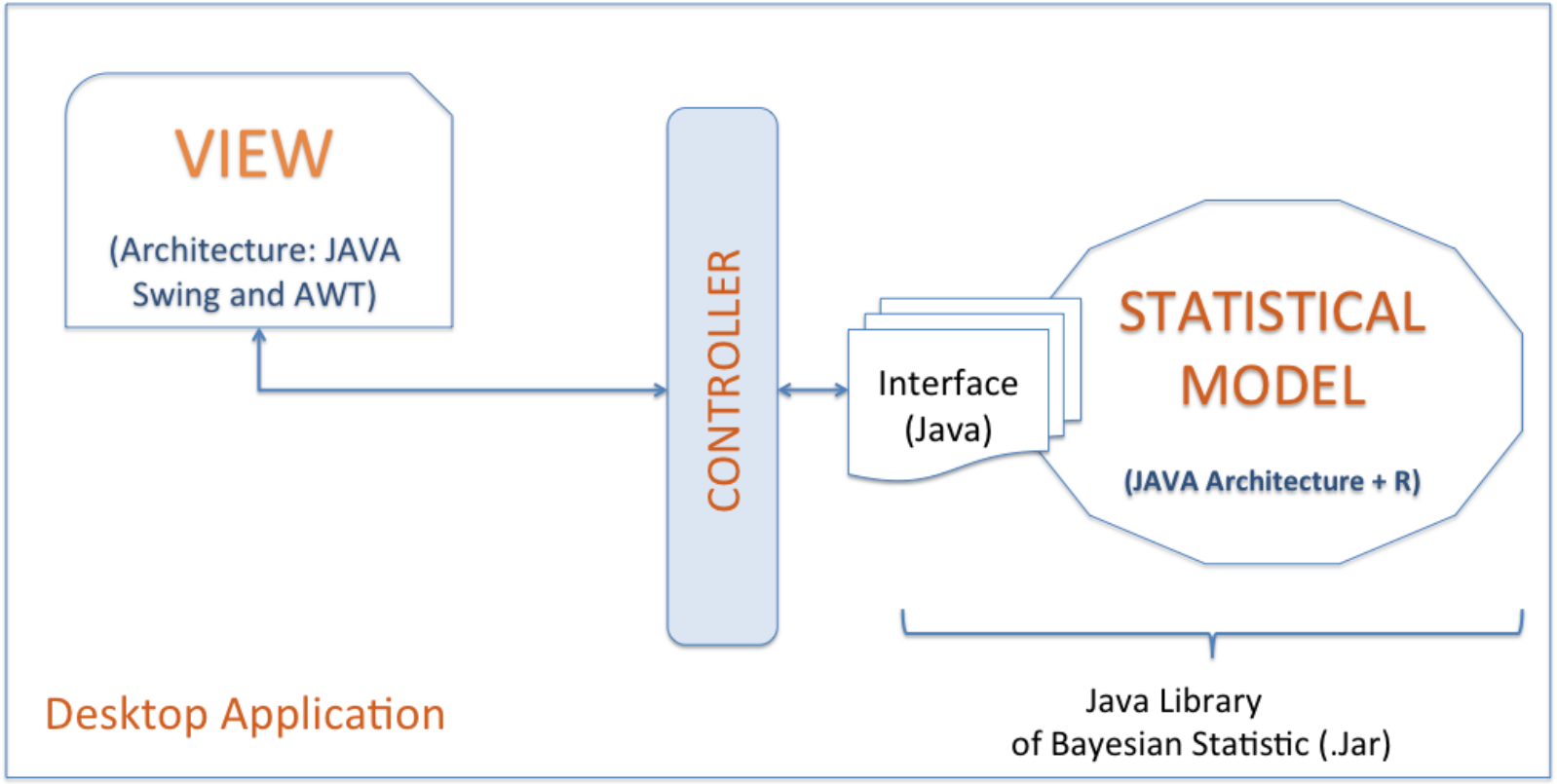
Artifacts, tools and technology

- After a preliminary **analysis** of the **alternatives** available to perform **Bayesian analysis**…

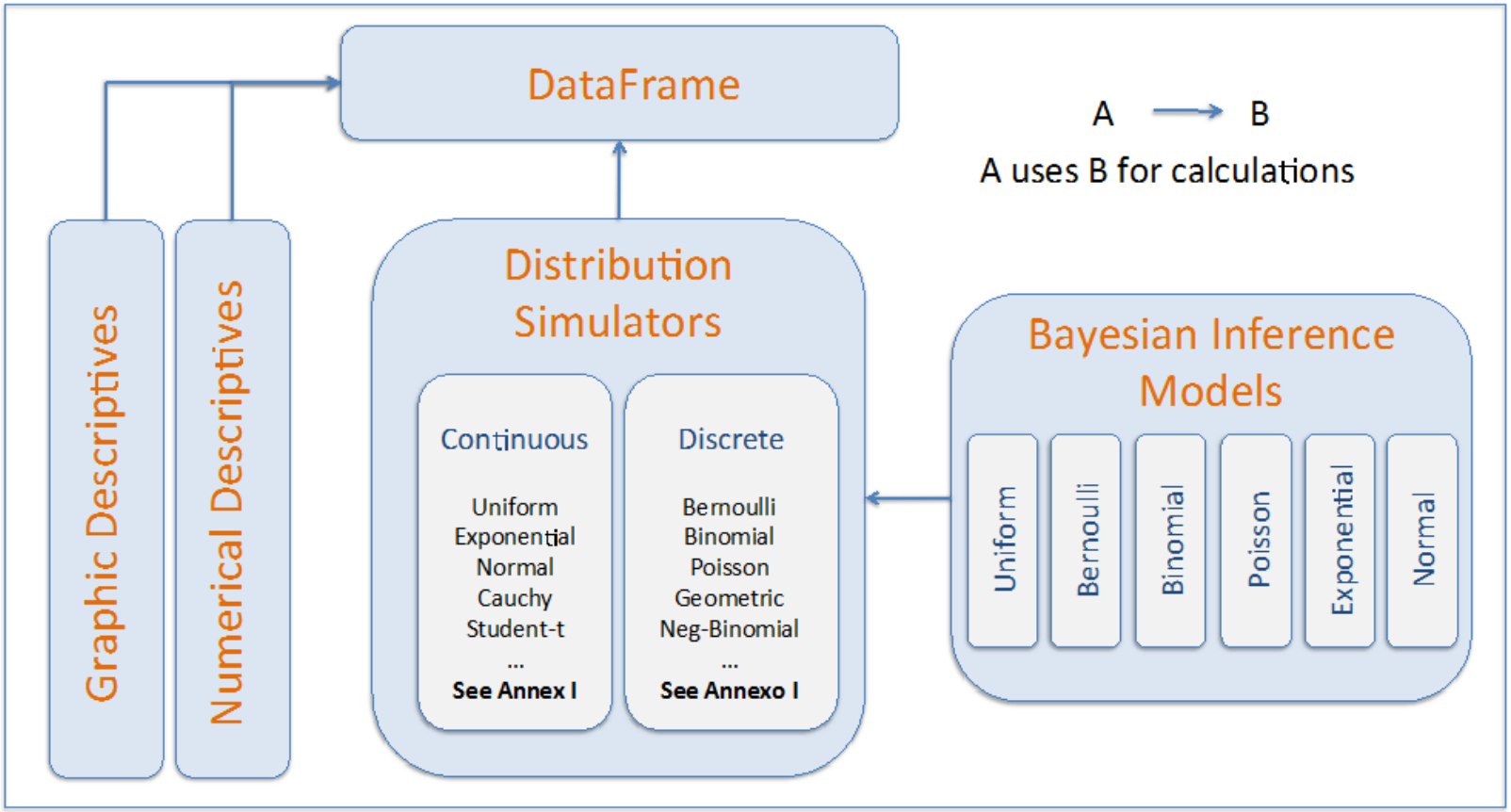
Software Element	Technologies	Libraries
Statistical Library	Java (JSE) + **R**	**JRI**
Desktop Application	Java Swing	Swing
Parallelization	**Parallel R**	**Snow Fall**

- …the above options were selected (free and reusable software platforms).

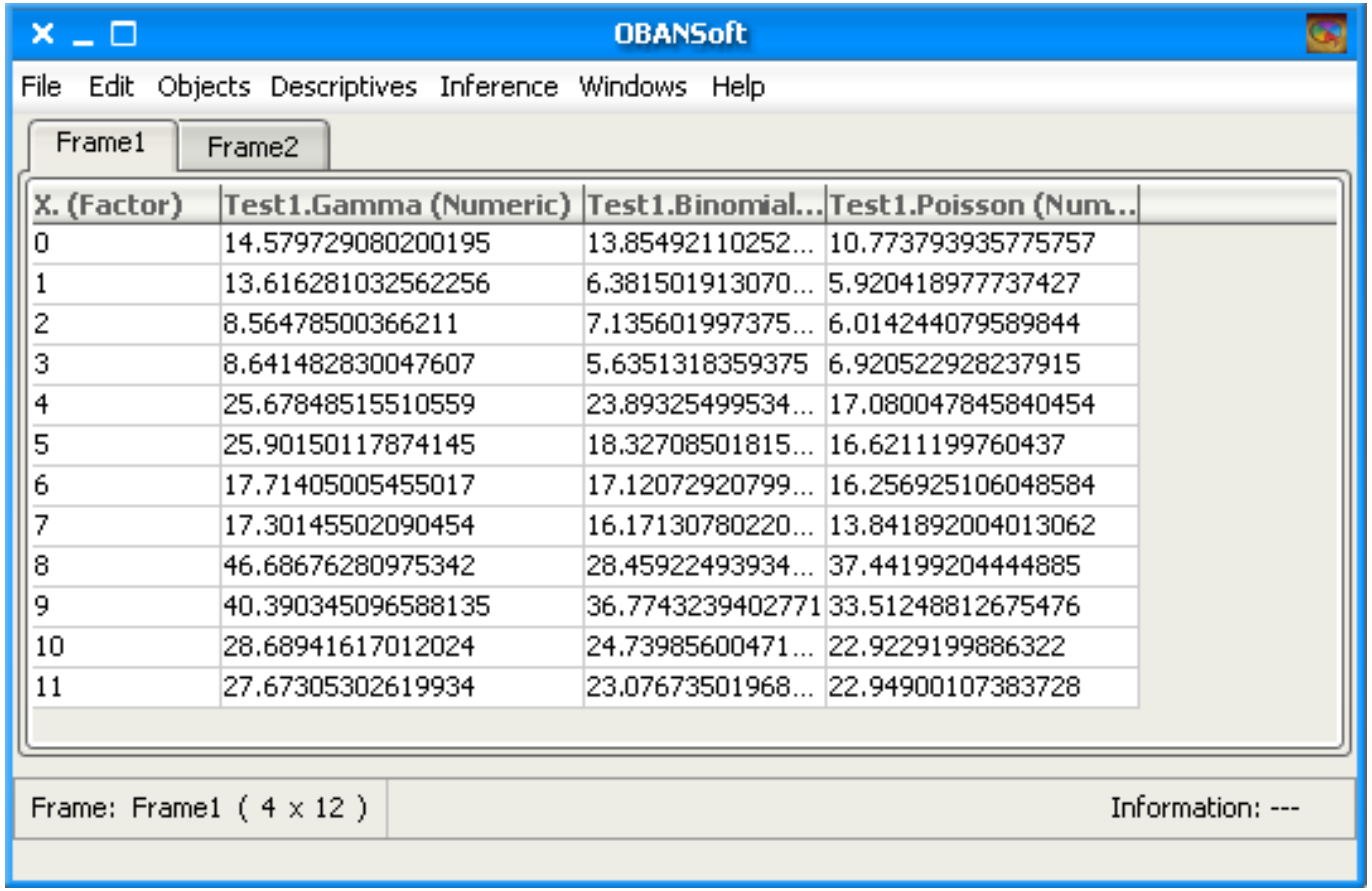
The model
Model-View-Controller



Object Model



View objects



Controller Objects

The Main Controller manages all events that require the participation of the "MainForm": **Main Controller**

Modular organization

FileController

EditController

DescriptiveController

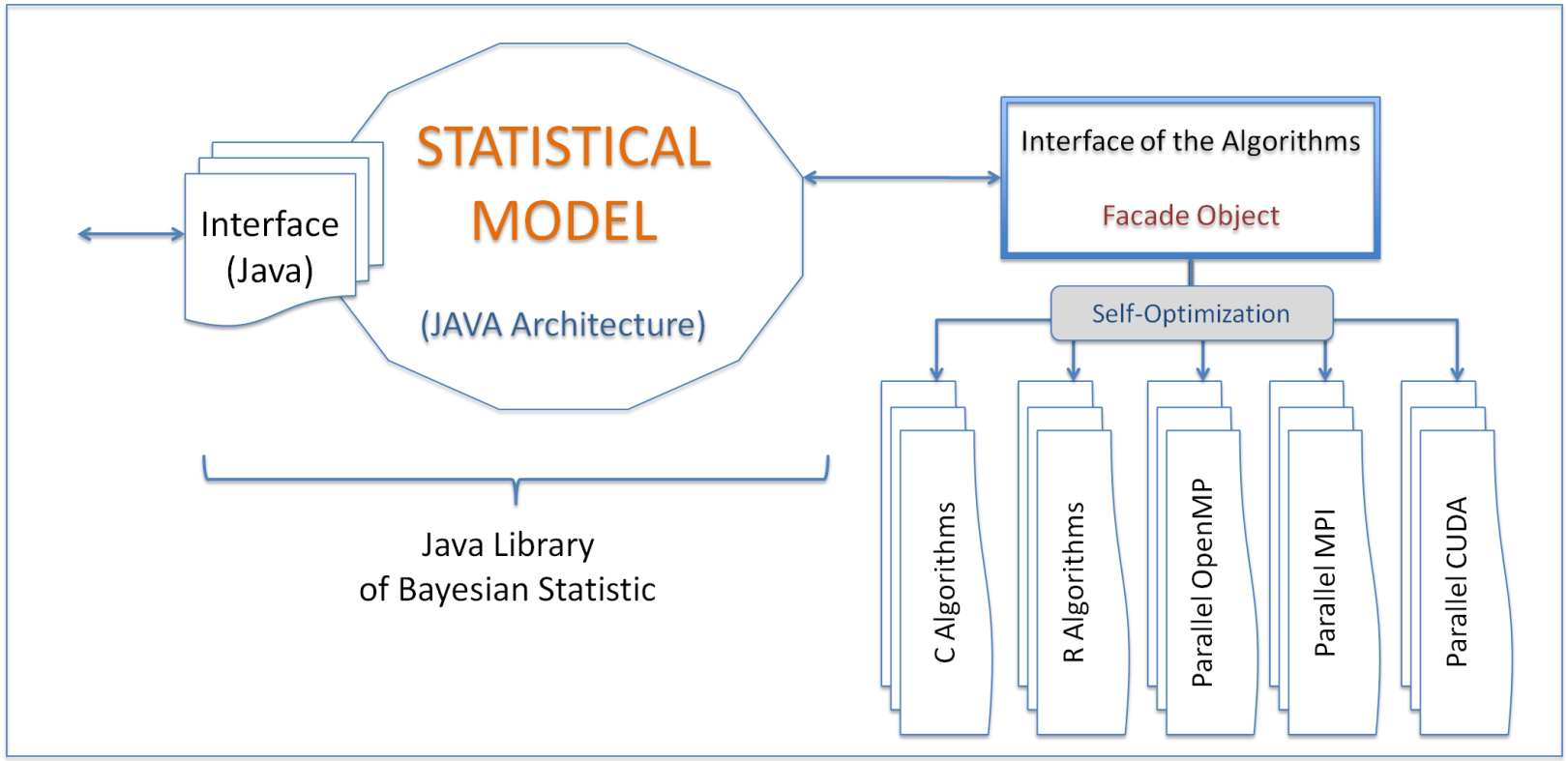
InferenceController

Other Objects

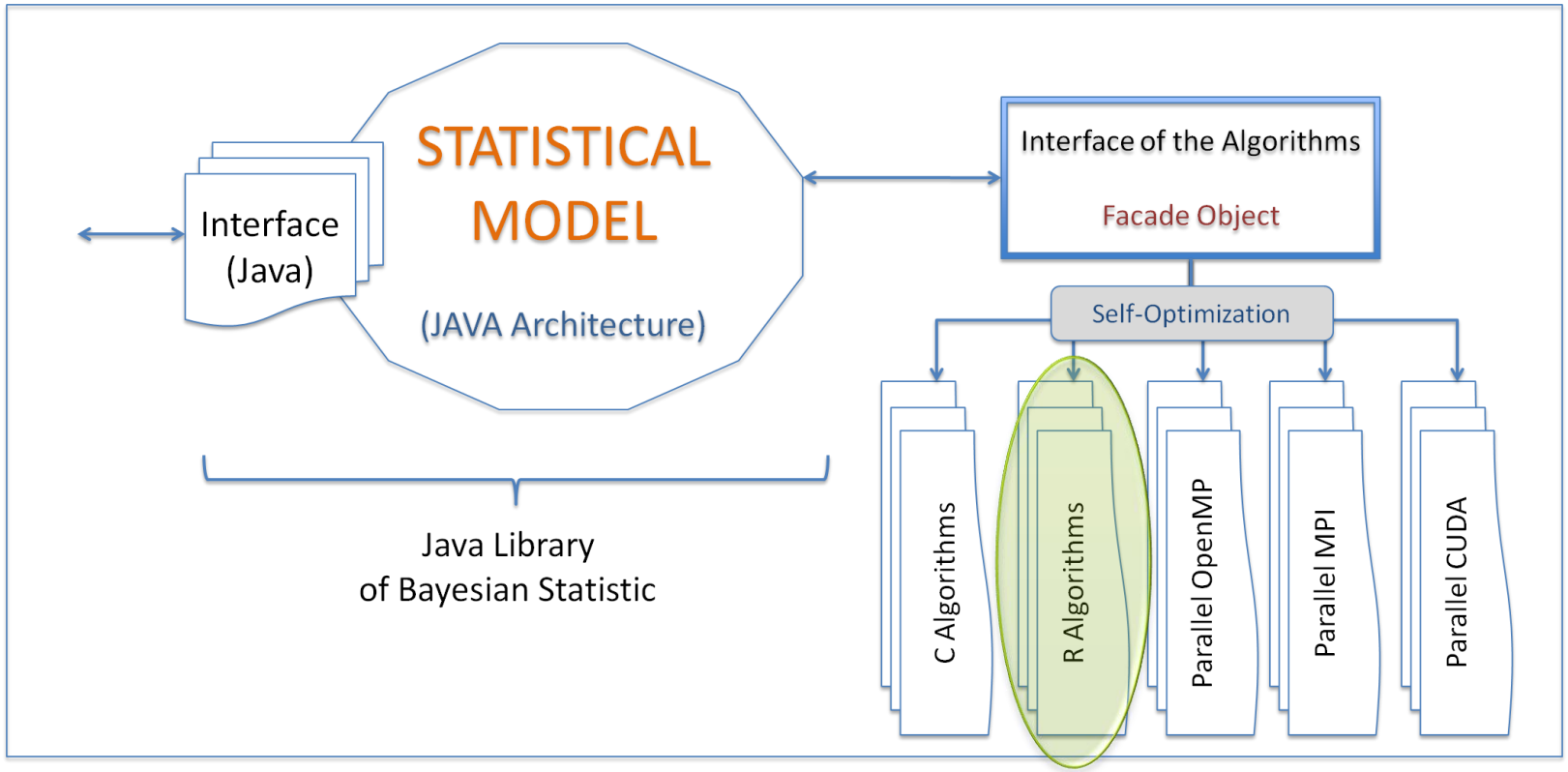
DataFramesController

…

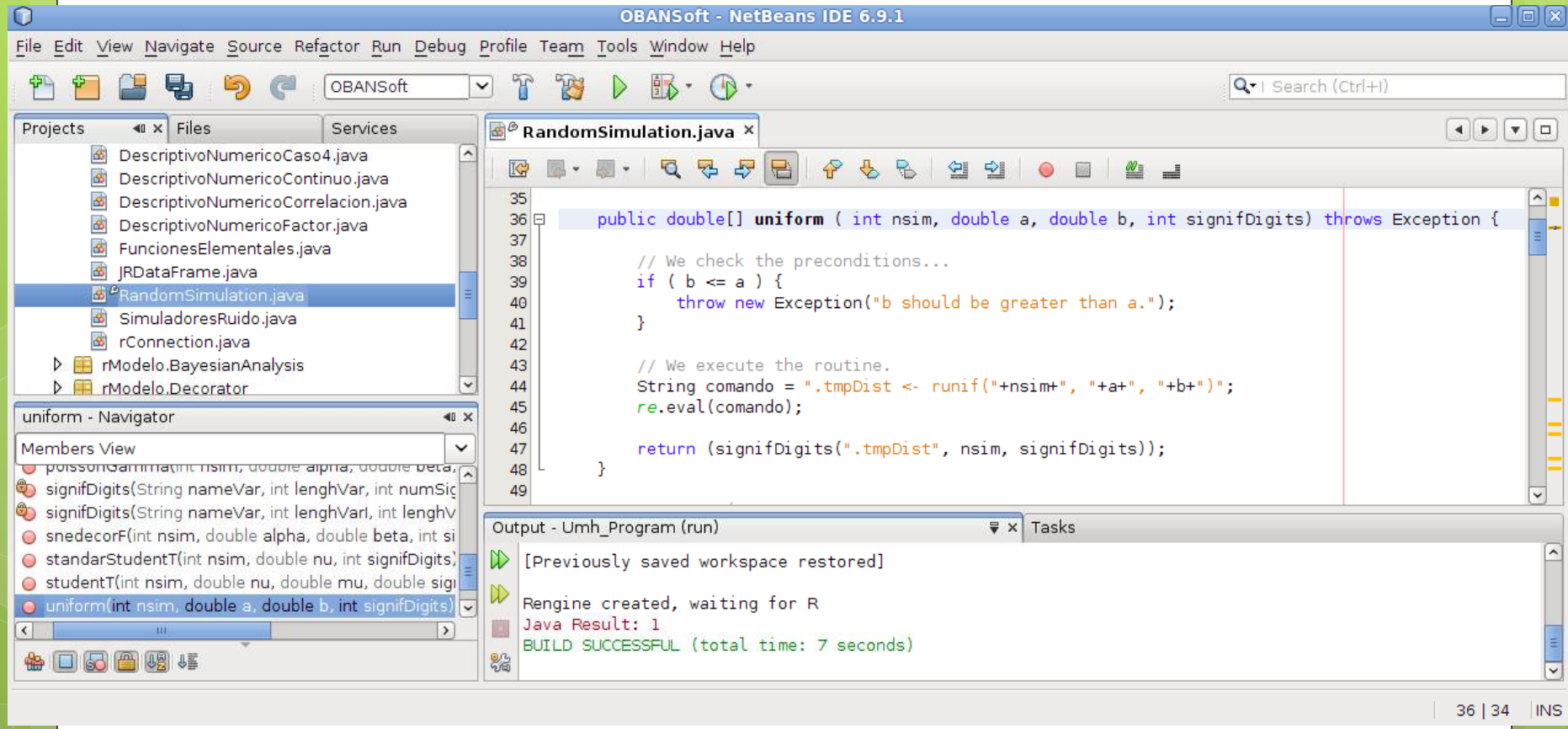
Bayesian algorithms. Integration of technologies.



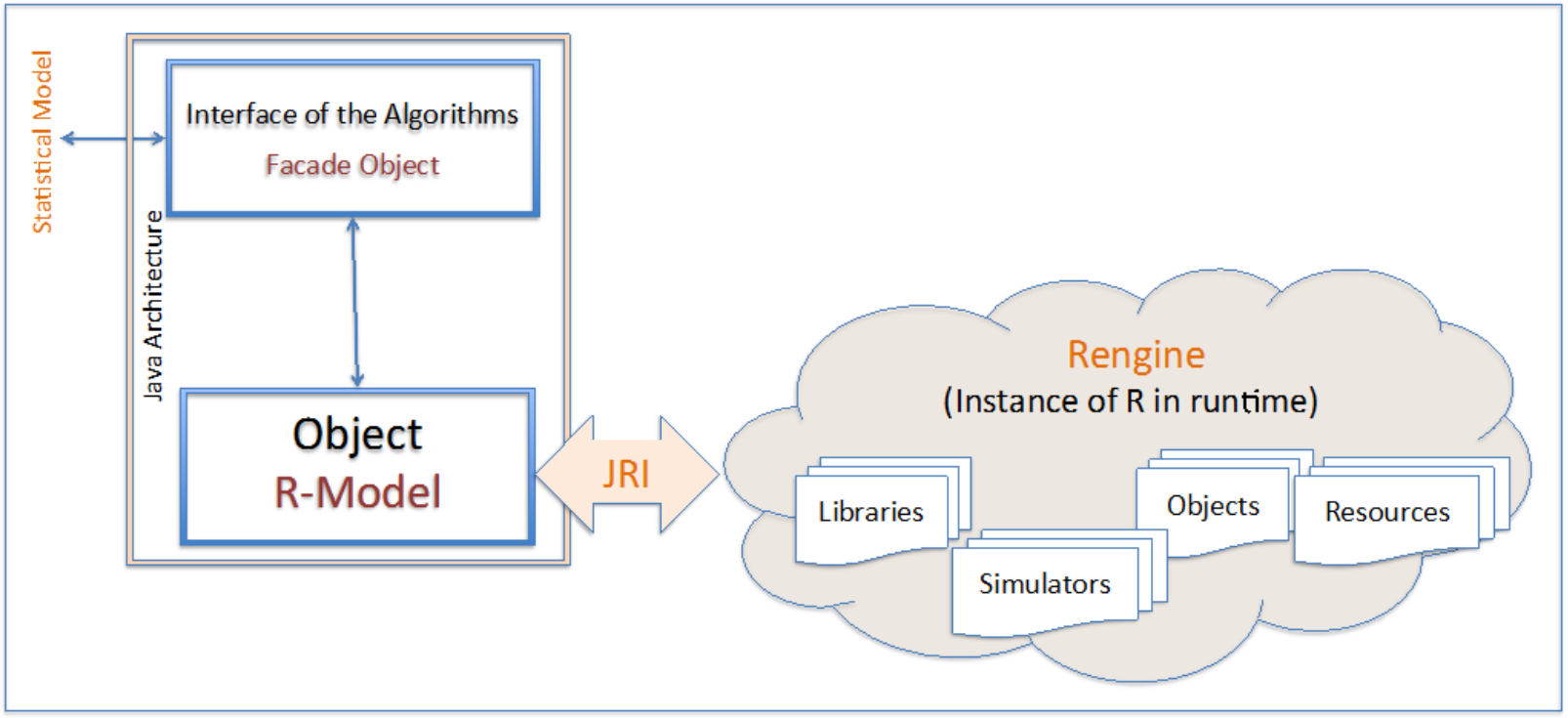
Bayesian algorithms. Integration of technologies.



Bayesian algorithms. Integration of technologies.



The R-Model and its integration with R.

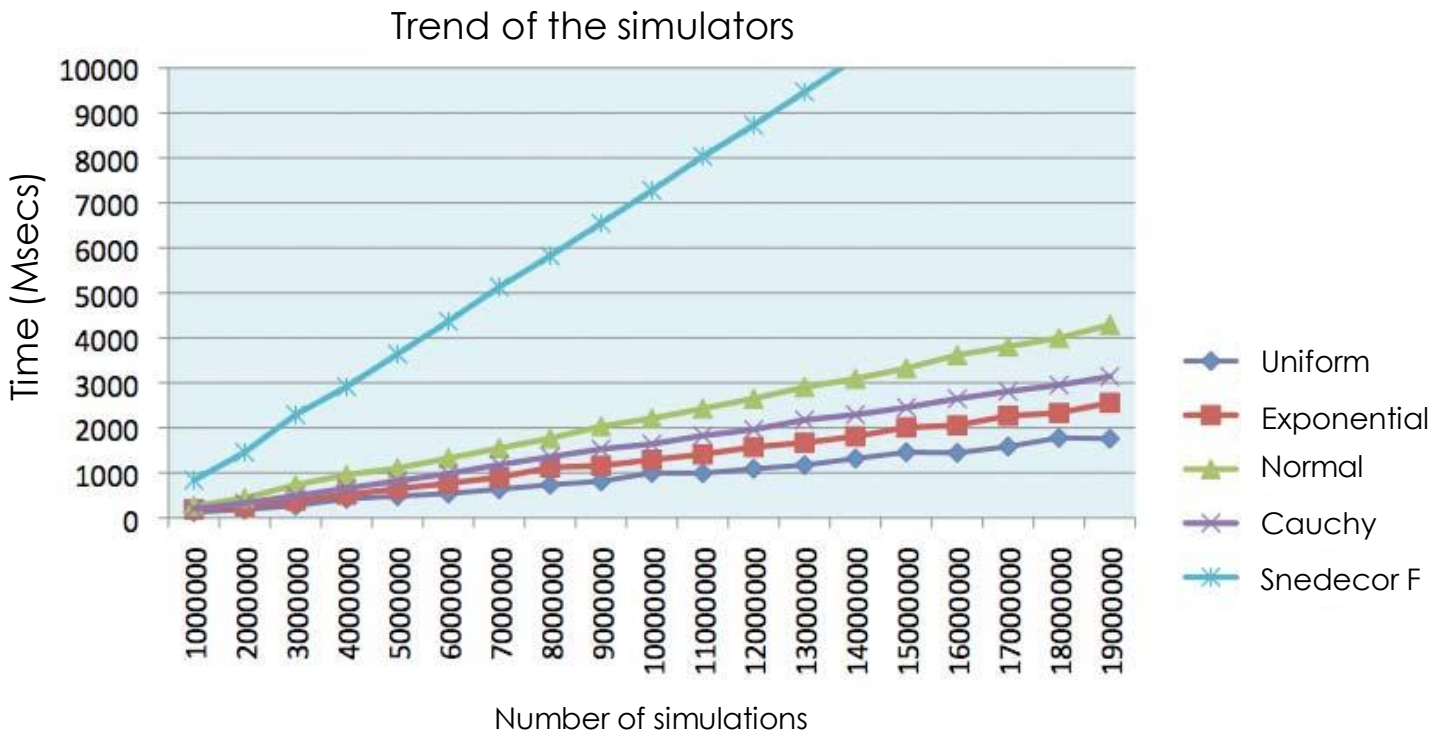


What algorithms to optimize and parallelize

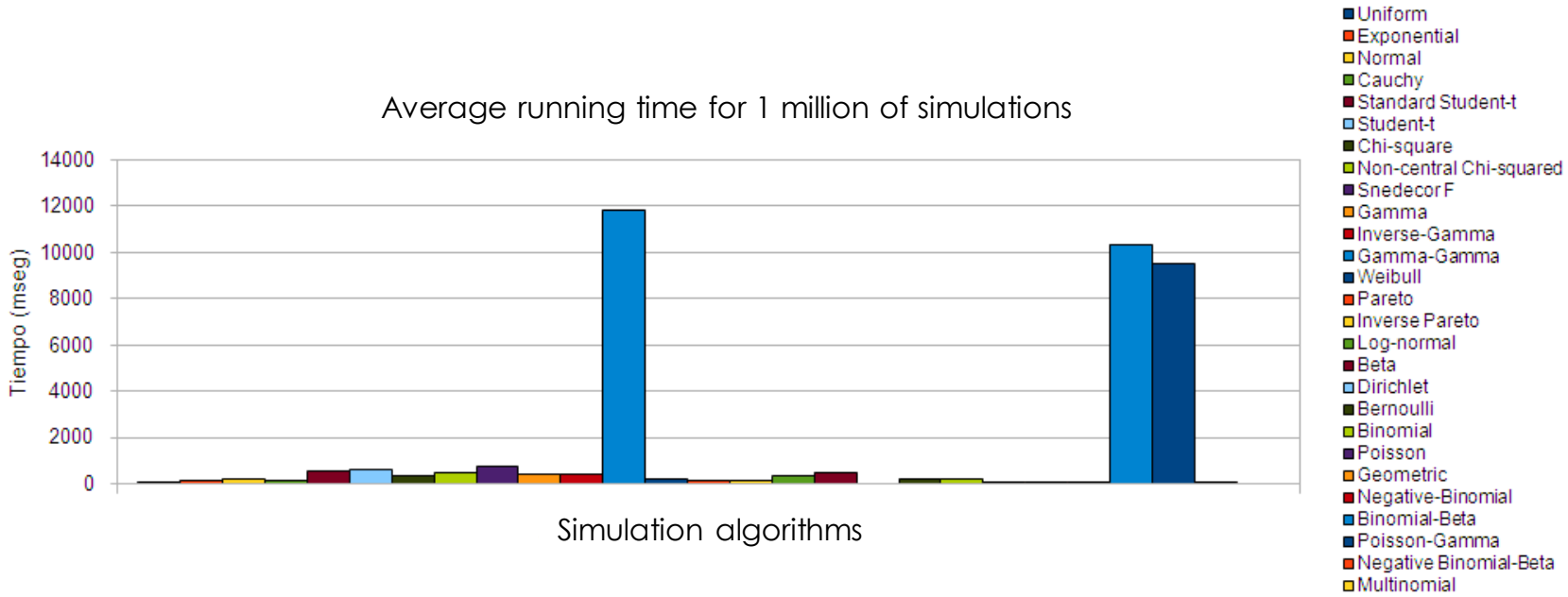
- Among all programming algorithms, we focus on **simulation algorithms.**
  - They require more runtime.
  - Critical point in the resolution of a Bayesian analysis.
  - All analyses are based on the simulation. They are used for Bayesian inference models.

However… there are 27, Who starts…?

Experiment 1: Trend growth



Experiment 2: Comparison of simulators



There were two types of simulators: **simple simulators** and **compound simulators**.

Composite Structure Simulator

- One invocation of a **simple function** of size X.
- X invocations of another simple function (**function chain**) with parameters extracted from the above function.

```
1  rgamma.gamma = function(nsim, alpha, beta, nu) {
2   theta=rgamma(nsim, alpha, beta);
3   x=vector(length=nsim);
4   for(i in 1:nsim) {
5           theta[i]=rgamma(1, nu, theta[i]);
6   };
7
8   return(theta);
9  }
```

Code 1: simulation algorithms of the composite function Gamma-Gamma

- The experiments indicated that the **function chain consumes 90% of the total execution time.**

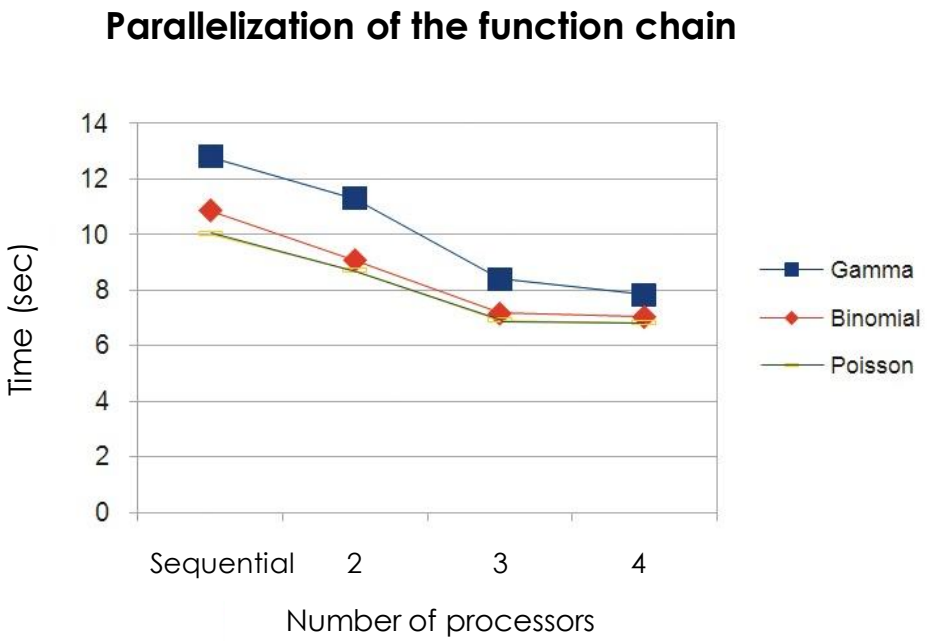
Chain function in parallel with R parallel code (library).

Parallelization for shared memory (**SnowFall**)

```
1   # Calcularíamos con la funcion simple los parametros
2   # alpha y beta que usaremos en la funcion rgamma.
3   ... ... ...
4   library(snowfall)
5
6   # 1. Inicializamos snowfall
7   sfInit(parallel=TRUE, cpus=1, type="SOCK")
8
9   # 2. Cargaríamos bancos de datos que queremos que
10  # sean leidos por todos los procesadores
11  # require(mvna)
12  # data(sir.adm)
13
14  # 3. Definimos el wrapper, el cual va a ser paralelizado.
15  wrapper <- function(idx) { return(rgamma(1,mu,theta)); }
16
17  # 4. Exportaríamos los datos y paquetes que queremos
18  # que sean leidos por todos los procesadores
19  # sfExport("sir.adm")
20  # sfLibrary(cmprsk)
21
22  # 5. Inicializamos el generador paralelo de numeros aleatorios
23  sfClusterSetupRNG()
24
25  # 6. Distribuimos los calculos
26  result <- sfLapply(1:tamSimulaciones, wrapper);
27
28  # 7. Detenemos snowfall
29  sfStop()
30  ... ... ...
31  ## Devolvemos el resultado de la simulacion
```

Code 2: Parallel algorithm chain simulator function (Gamma-Gamma)

Experiment 3: Results of the parallelization



The reduction in the execution time is far from the theoretical limit…
(Efficiency only 50%)

What is the reason…?

Current work….

- We are studying a Bayesian Analysis algorithm: **study** of parallelism (Snowfall, multithreaded BLAS, OPENMP…)
- We analyze the **simulation codes programmed in C** to compare with the corresponding R versions.

  IMSL Libraries for linux.
- **Parallelize** these algorithms programmed in C and compare **SnowFall** against **OpenMP**.

Future work….

With the tool we cover that gap in the applications of Bayesian statistics, and it serves as a basis for integrating future developments hiding parallelism.

- **Integrate** other models that involve the simulation algorithms based on **Markov chains**.
- **Expand** OBANSoft modules with new functionality.
- Adapt the **statistical model** in a website to exploit as **Cloud Computing**.

References

- Katagiri, T., K. Kise, H. Honda, and T. Yuba (2004). Effect of auto-tuning with user's knowledge for numerical software. In Proceedings of the 1st conference on Computing frontiers, pp. 12–25. ACM.

- Quesada, M. (2010, Julio). Obansoft: aplicación para el análisis bayesiano objetivo y subjetivo. estudio de su optimización y paralelización. Master's thesis, Universidad de Murcia.

- SnowFall (2011). Url http://cran.r-project.org/web/packages/snowfall/.

- Yang, R. and J. O. Berger (1996). A catalog on noninformative priors. Discussion Paper, 97-42, ISDS, Duke University, Durham, NC.

Thank you for your attention.

Any questions…?