

Tutorial: The OpenMOLE platform for model exploration and validation

Juste Raimbault^{1,2,3,4,*}, Paul Chapron¹, Maxime Colomb^{5,1}, Julien Perret¹, Romain Reuillon^{3,4}, Mathieu Leclaire^{3,4}
* juste.raimbault@ign.fr

¹ LASTIG, IGN-ENSG

²CASA, UCL

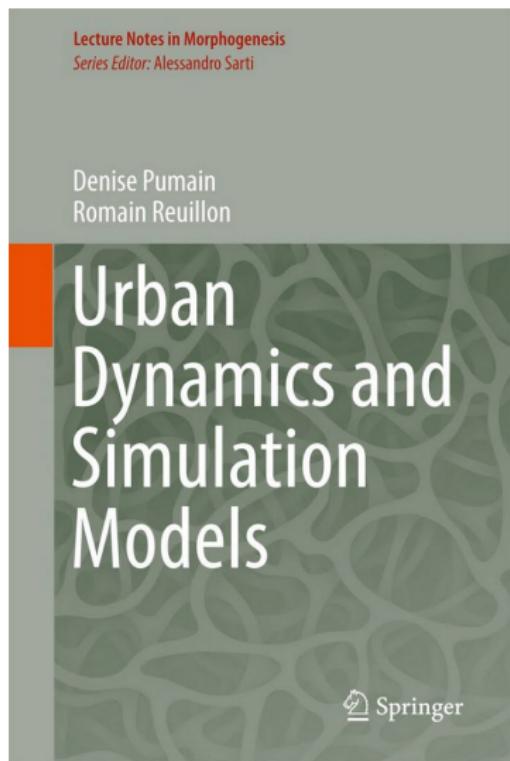
³UPS CNRS 3611 ISC-PIF

⁴UMR CNRS 8504 Géographie-cités

⁵ASCII, INRIA

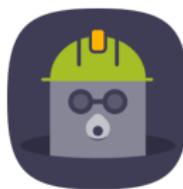
Journée de la Recherche UGE-IGN-ENSG 2024

28/03/2024



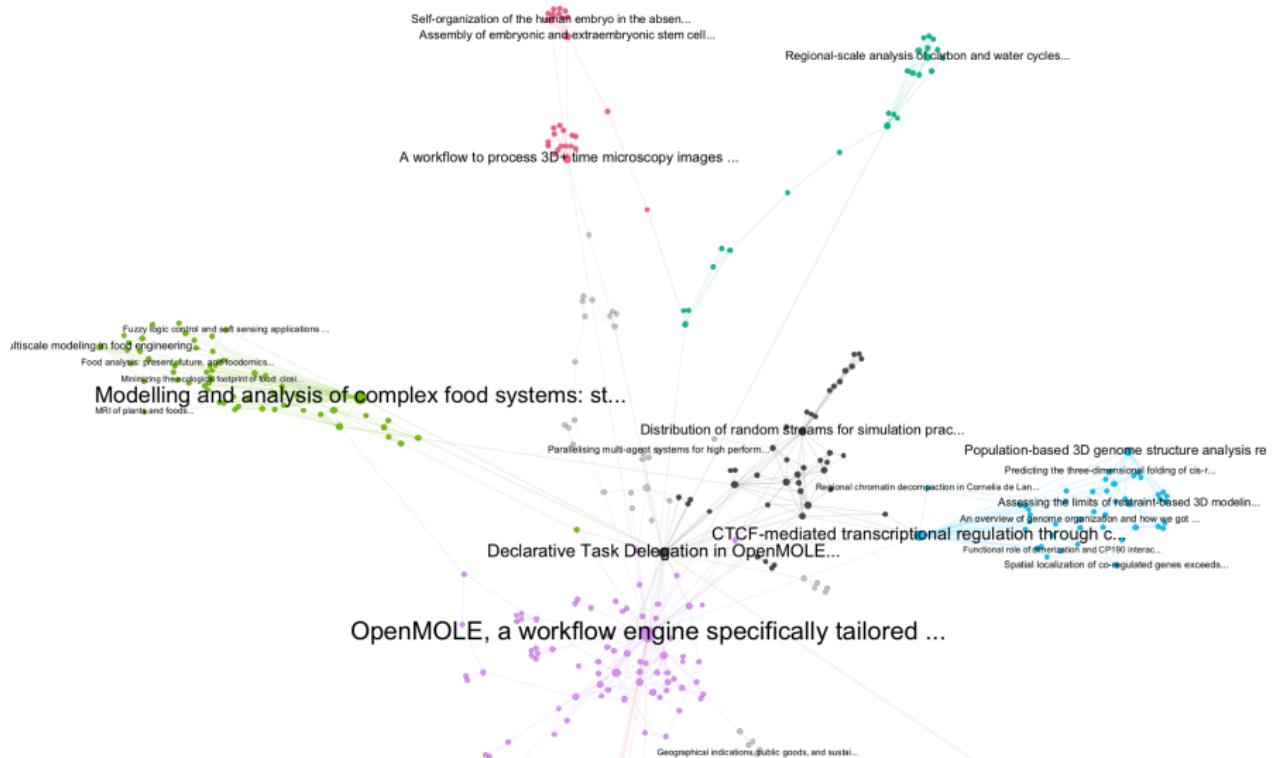
Evolutionary urban theory [Pumain, 2018]

- Stylised facts on main systems of cities worldwide
- Simulation models with an explicative function
- Tools and model exploration methods: OpenMOLE mainly developed by R. Reuillon and M. Leclaire since 2008 at l'ISC-PIF



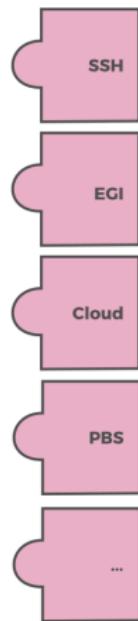
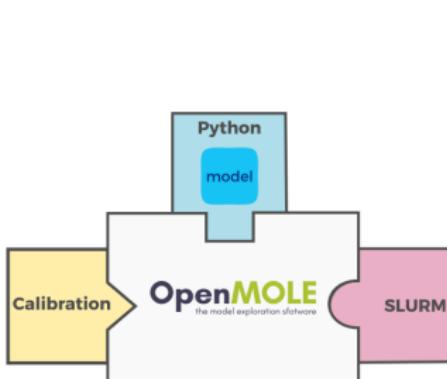
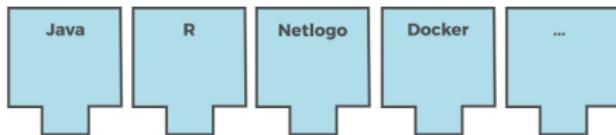
OpenMOLE

Scientific environment [Raimbault, 2018]



OpenMOLE principles

(i) State-of-the-art exploration and validation methods; (ii) Scaling with High Performance Computing; (iii) Model embedding.



Web application interface

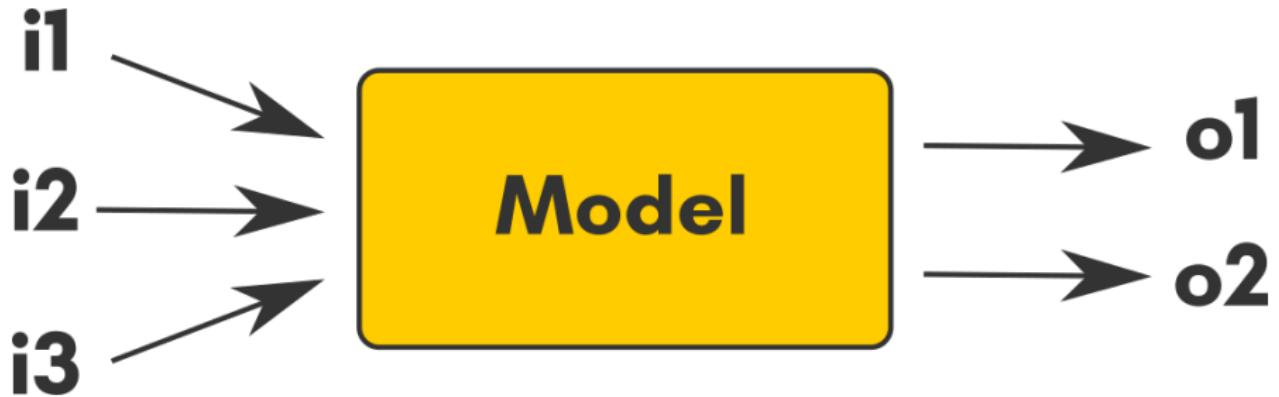
The screenshot shows the OpenMOLE web application interface. On the left, there is a sidebar with a list of projects and tutorials, each preceded by a blue plus sign icon. The projects listed are:

- + 00_Hello World in Python
- + 01_Hello World in R
- + 02_Hello World in Scilab
- + 03_Hello World in Java
- + 04_Hello World in NetLogo
- + 05_Hello World in Julia
- + 06_Hello World in GAMA
- + 07_Hello World in a Container
- + 08_Hello with OpenMOLE plugin
- + 09_Model Exploration Tutorial
- + 10_Native Application Tutorial
- + 11_Workflow Tutorial
- + 12_Morris Sensitivity Analysis
- + 13_Saltelli Sensitivity Analysis
- + 14_ABC
- + 15_Calibration of Ants
- + 16_Optimise Ackley function in Python
- + 17_Pi Computation
- + 18_Random Forest
- + 19_SimpopLocal
- + 20_Metamimetic Networks
- + 21_Segmentation with FSL
- + 22_NSGA2 Test Functions
- + 23_Generate Visualisation

The main area contains a code editor titled "Container.oms". The code is written in Scala and defines a task for containerization. It includes a "ContainerTask" block with a command to update the system and install dependencies, followed by a loop that sets up inputs and outputs for each iteration. The code also includes a "parse" method to handle results and a "DirectSampling" block for evaluation.

```
1 val i = Val[Int]
2 val result = Val[String]
3 val resultant = Val[Int]
4
5 val container =
6   ContainerTask(
7     "debian:stable-slim",
8     "echo ${${i}+2}",
9     install = Seq("apt update", "apt install -y bash"),
10    start = result
11  ) set (
12    inputs ++ i
13  )
14
15 val parse =
16   scalatask("""val resultInt = result.split("\n").last.toInt""") set (
17    inputs ++ result
18    outputs += resultant
19  )
20
21 DirectSampling{
22   sampling = i in (0 to 10),
23   evaluation = container --> parse
24 } hook (workDirectory / "result.csv")
```

Simulation models



Language agnosticity

C
R
C++
Java
Scala
Scilab
Octave
Python
Netlogo
...



Example: NetLogo model

```
val model =  
  NetLogo6Task(  
    workDirectory / "Fire.nlogo",  
    List("setup", "while [any? turtles] [go]")) set (  
      inputs += seed,  
      outputs += (seed, density),  
      inputs += density mapped "density",  
      outputs += burned mapped "burned-trees"  
  )
```

DSL based on scala for scripts

R code

```
val i = Val[Int]

val rTask =
  RTask("""
    source("function.R")
    function(i) {
      set(
        resources += workDirectory / "function.R",
        inputs += i
    )
  }
```

Similar syntax for the PythonTask

Docker container

```
val i = Val[Int]
val result = Val[String]
val resultInt = Val[Int]

val container =
  ContainerTask("debian:stable-slim", "echo $(( ${i}*2 ))",
    install = Seq("apt update", "apt install -y bash"),
    stdOut = result
  ) set (
    inputs += i
  )

val parse =
  ScalaTask("""val resultInt = result.split("\n").last.toInt""") set (
    inputs += result,
    outputs += resultInt
  )

DirectSampling(
  sampling = i in (0 to 10),
  evaluation = container -- parse
) hook (workDirectory / "result.csv")
```

Methods

- Parameter estimation
- Sensitivity analysis
- Robustness analysis
- Optimisation

Designed to be scalable, to take stochasticity into account, to be usable on any model and computing environment.

Methods

```
DirectSampling(  
    evaluation = myModel,  
    sampling =  
        LHS(  
            500,  
            diffusion in (10.0, 100.0),  
            evaporation in (10.0, 100.0)  
        )  
)
```

Example of method syntax: explicit sampling

Computing environment: up-scaling

Local prototypes, transparent scaling.

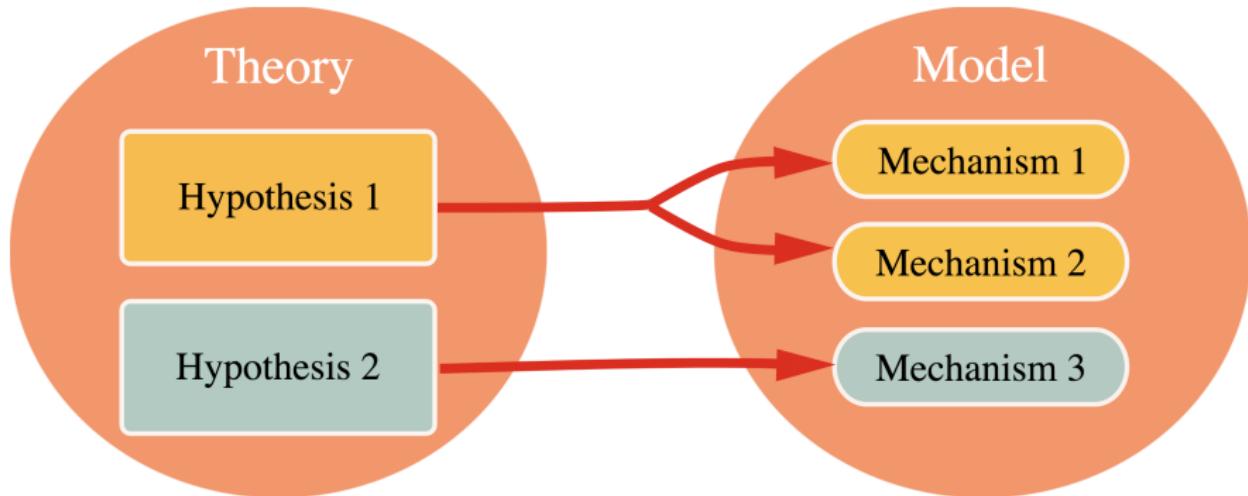
```
val cluster = SLURMEnvironment("login", "cluster.domain.org")

DirectSampling(
    evaluation = myModel on cluster,
    sampling =
        LHS(
            500,
            diffusion in (10.0, 100.0),
            evaporation in (10.0, 100.0)
        )
)
```

Computing environment

- Multi-thread
- Delegation through SSH
- PBS
- SLURM
- Condor
- SGE
- OAR
- EGI Grid

Theories and model evalution



Construct and evaluate a theory implying causal mechanisms.

Evaluation: How to ensure

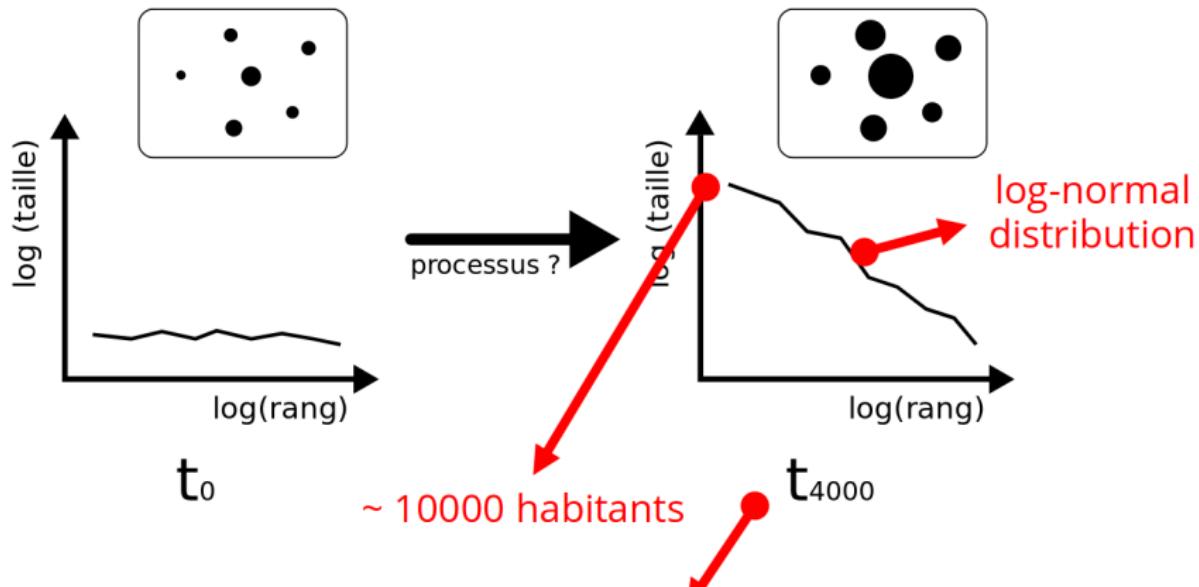
- ① the sufficiency of the mechanisms ?
- ② the necessity of the mechanisms ?
- ③ the uniqueness of the mechanisms ?

Sufficiency [Schmitt et al., 2015]

Classical approach: parameter space sampling (ex. Sobol) → large dataset produced and parameter space remains unexplored.

Inverse approach: from outputs to parameters

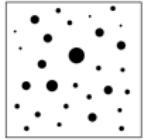
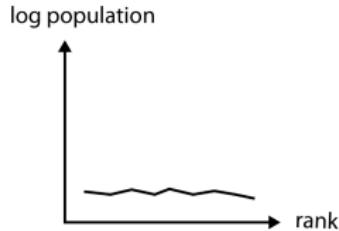
Formalising the expectations:



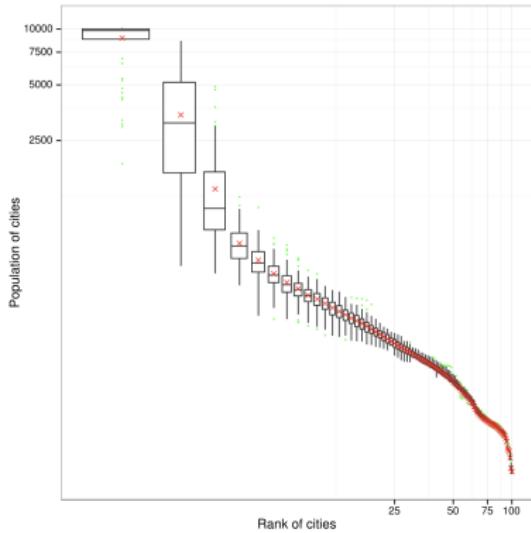
Calibration results

No compromise between the 3 objectives.

Searched pattern



Produced pattern

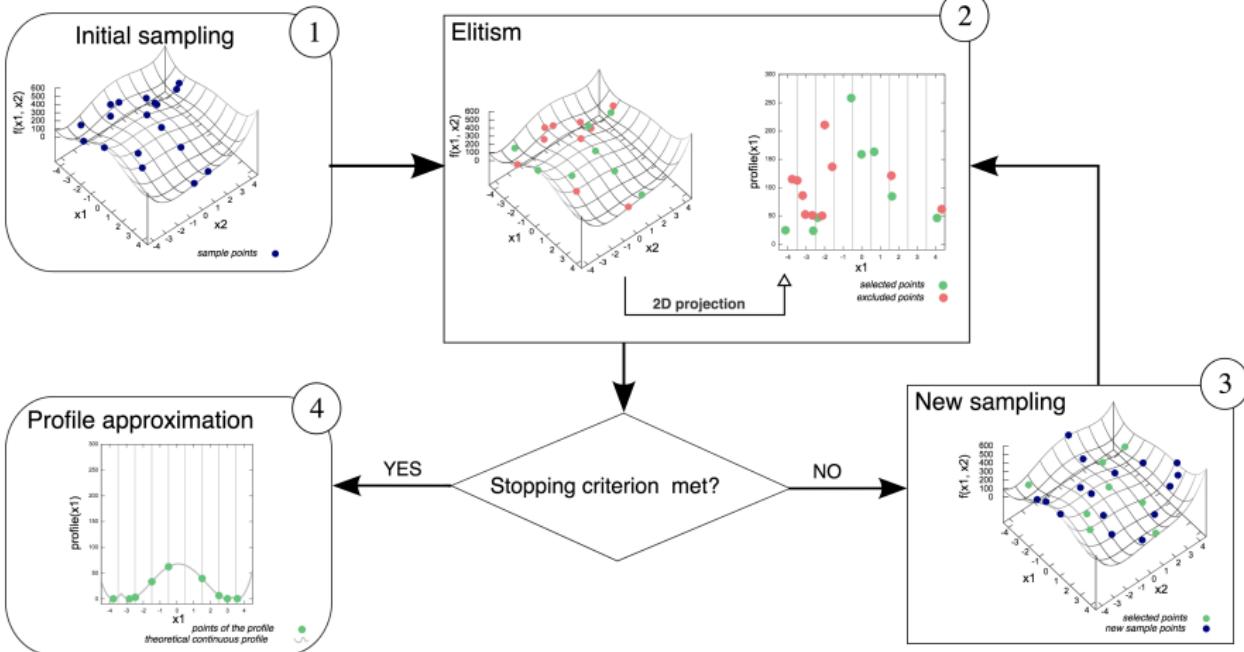


The method is tractable (even for ABMs): Handles stochasticity: 100x gain; Support for distributed computing: 1000x gain.

A new algorithm

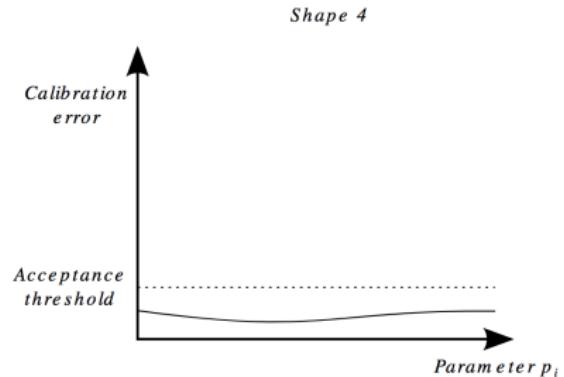
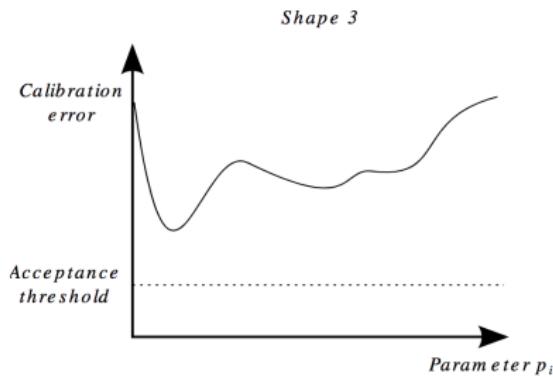
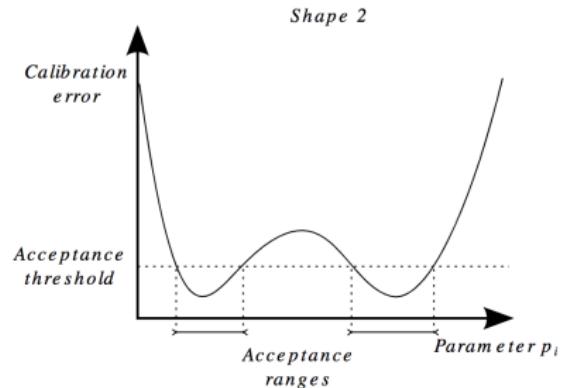
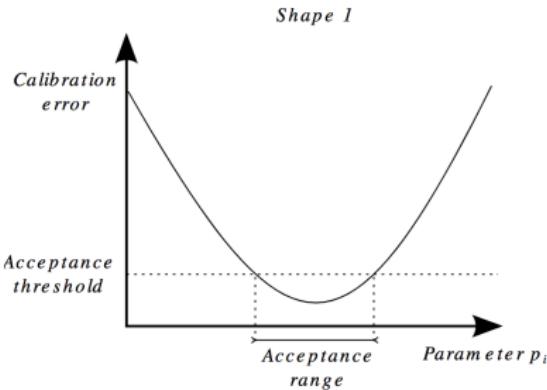
- ① To detect if a parameter is useful: it impacts the capacity of the model to produce plausible outcomes.
- ② To better constrain the parameter ranges.
- ③ As an indirect way to detect if some of the mechanisms are expandable

Profile algorithm

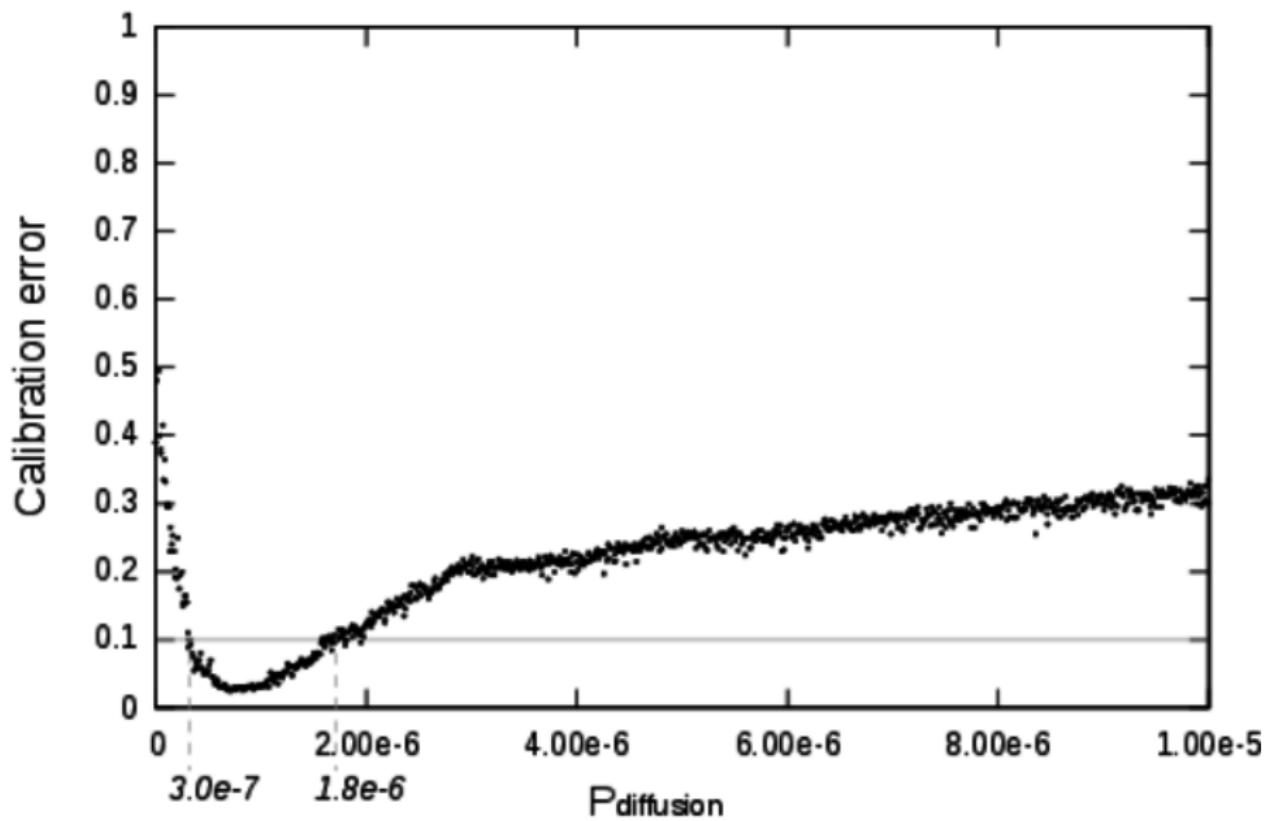


Compute the best of calibration for hundreds of values along the definition domain of a parameter.

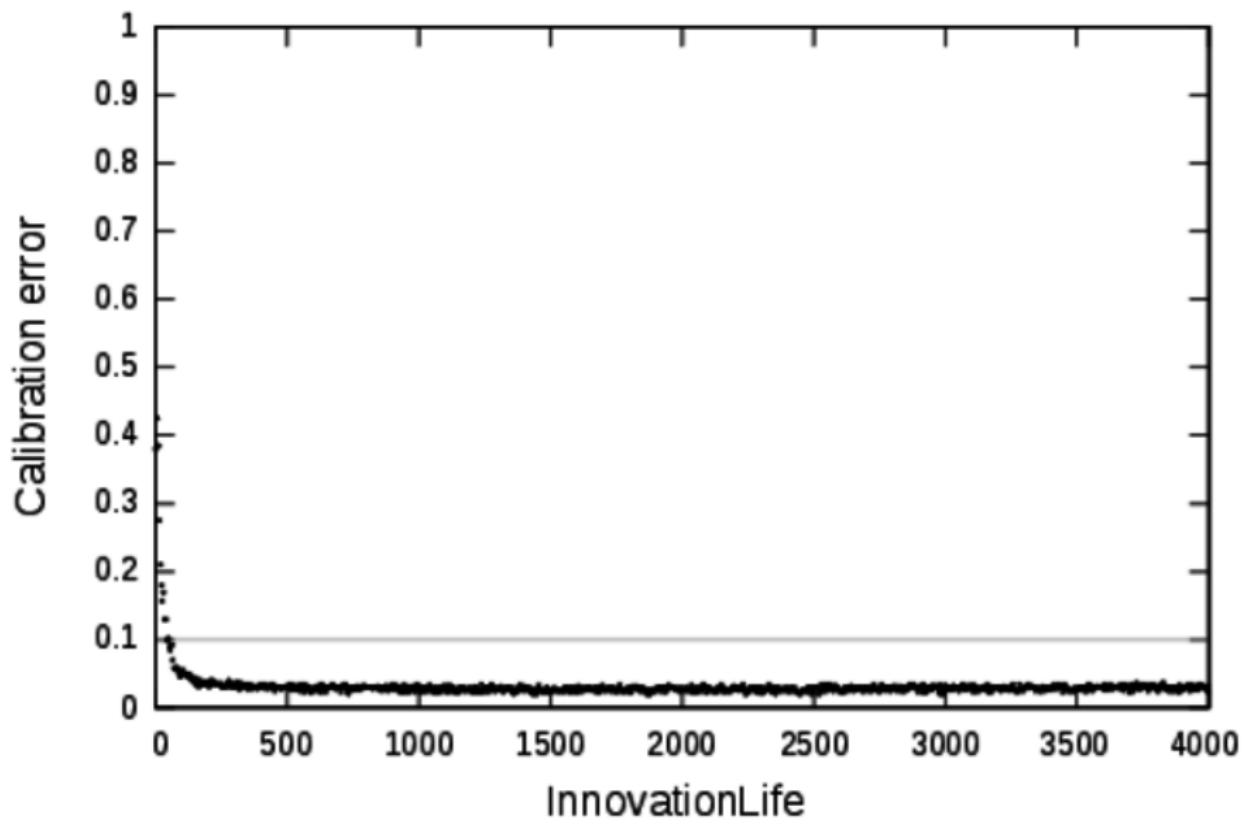
Profile algorithm



Results

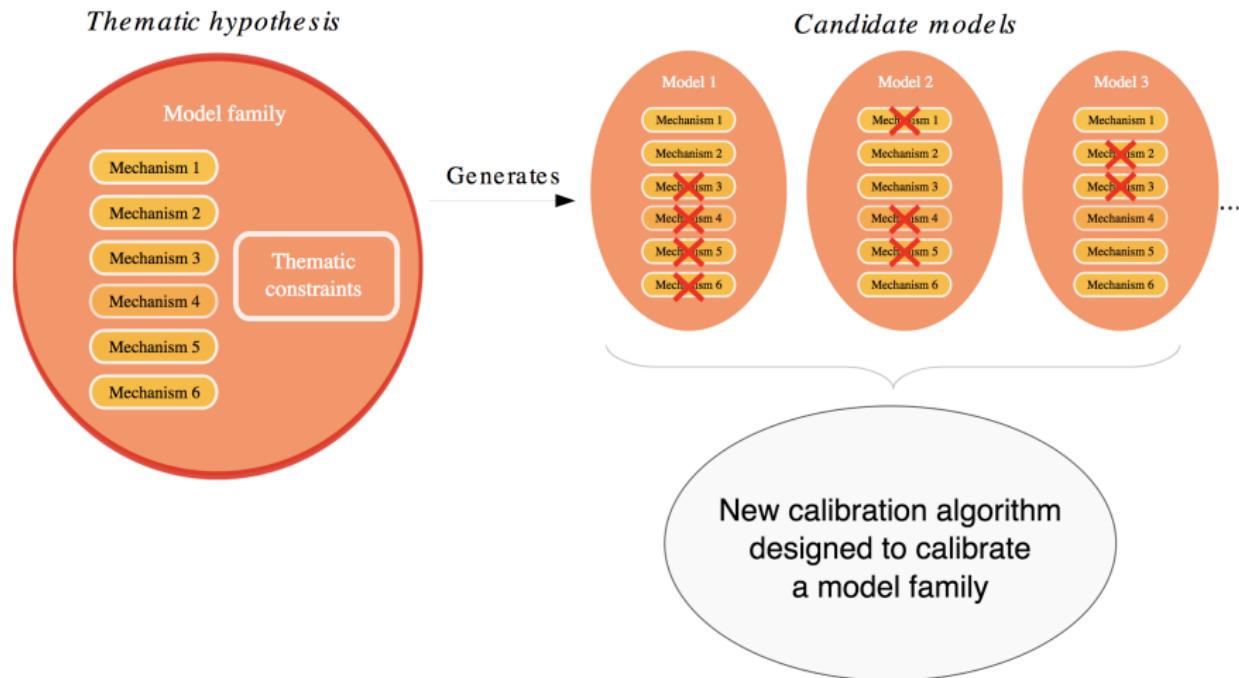


Results

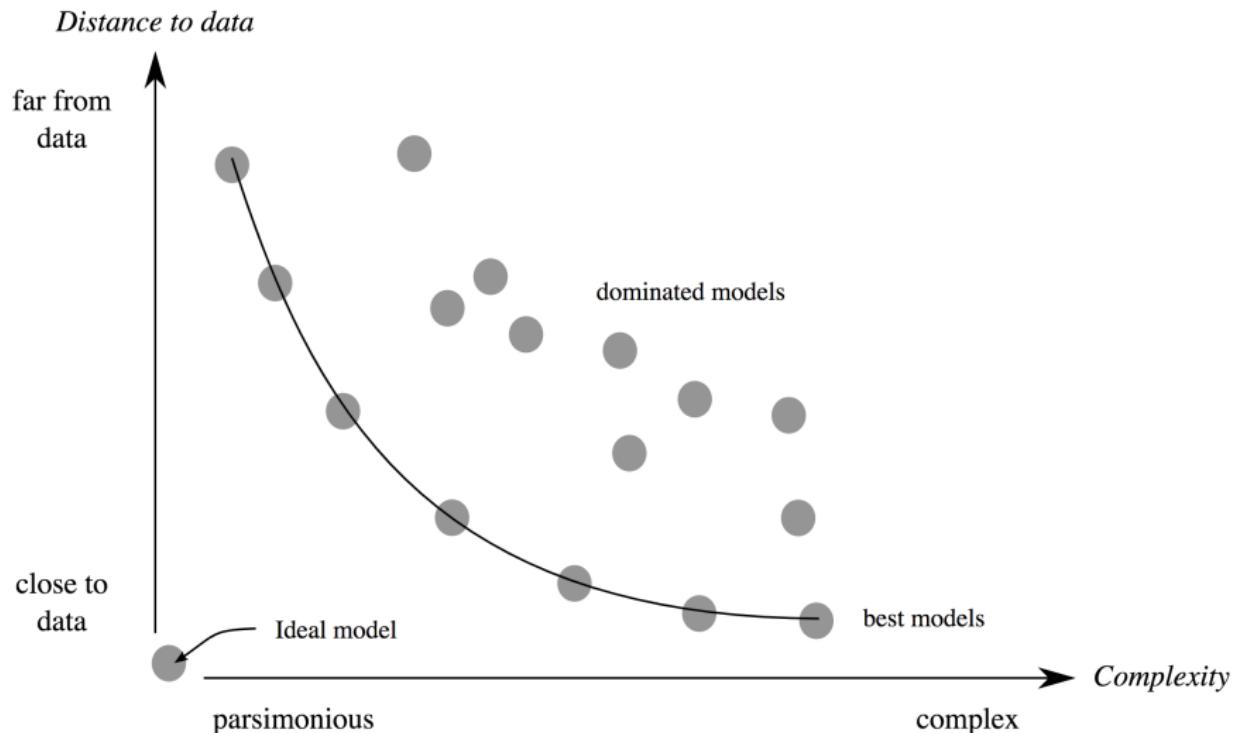


Unicity [Cottineau, 2014]

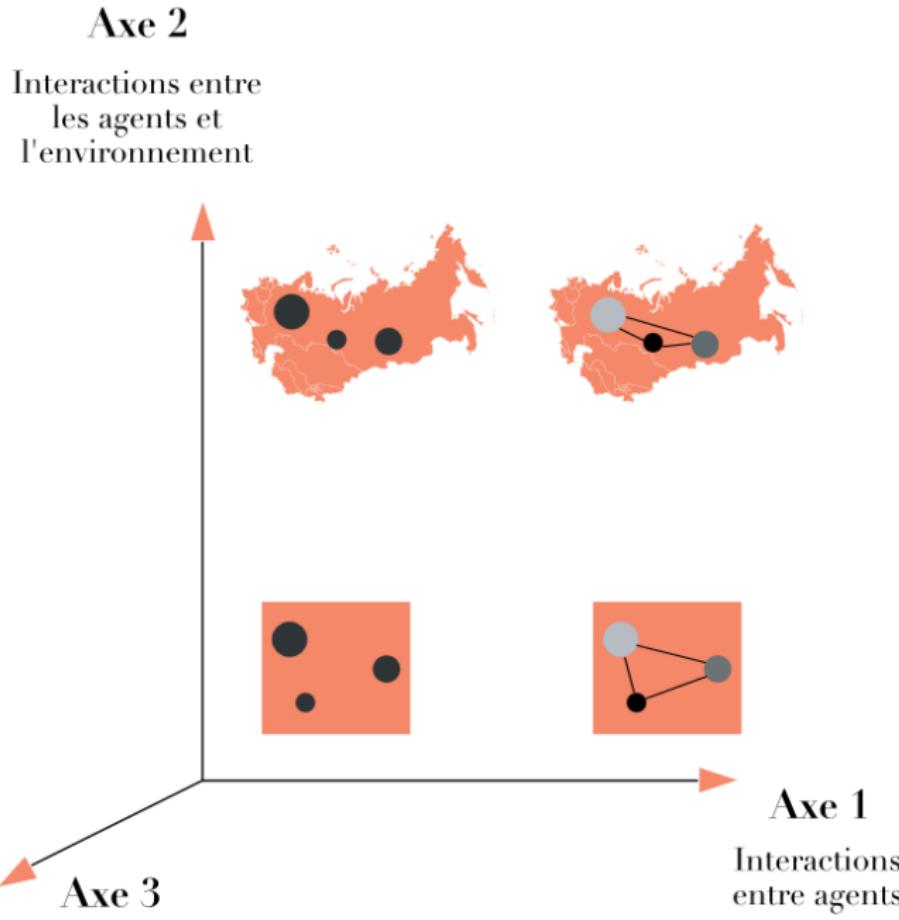
Automate the confrontation of alternative hypothesis / mechanisms.



Objective



Multi-modeling (64 models)

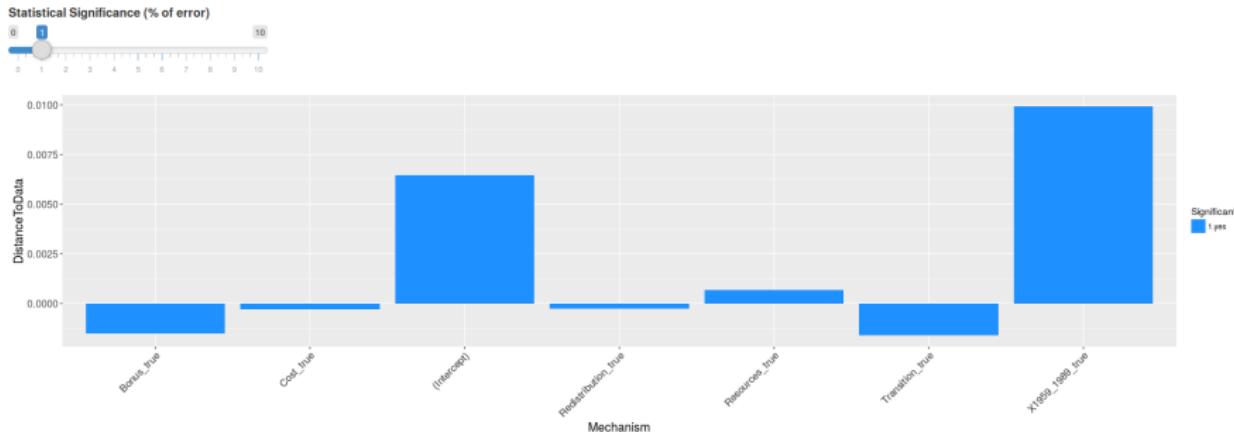


Calibration of the model family

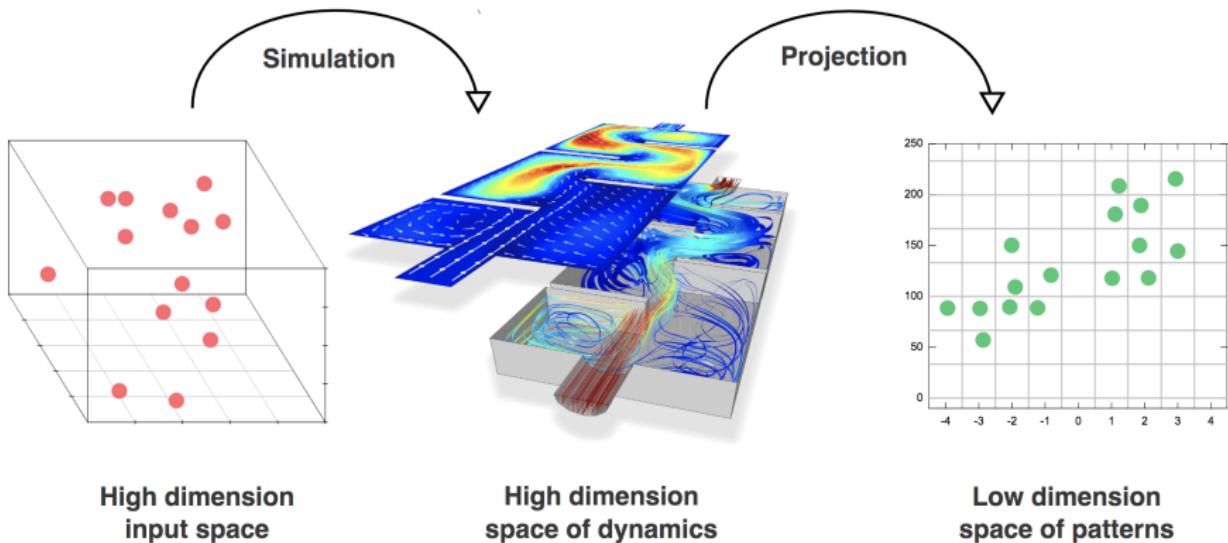
Compute the best set of parameters for all 64 models, using a niched NSGA2 algorithm.

Contribution of mechanisms to the quality of simulation (closeness to data)

Models with different combination of mechanisms have been calibrated intensively against empirical data, using generic algorithms for more than 100000 generations. This plot shows the results of a regression explaining one measure of the quality of models (a small difference between simulated and empirical urban trajectories) by their mechanisms composition (the fact that any of the supplementary mechanisms is activated or not). Each bar represents the value of the estimated coefficient for each activated mechanism, in comparison with the same model structure without this mechanism, everything else being equal.

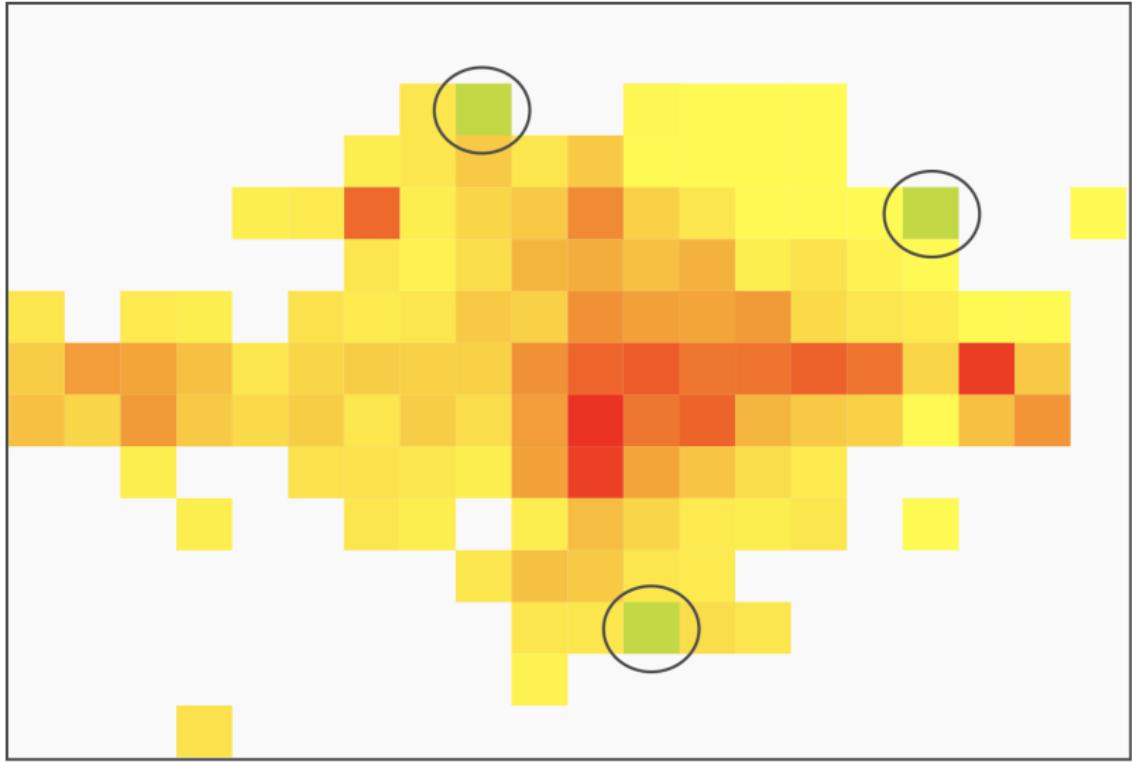


Novelty search [Chérel et al., 2015]

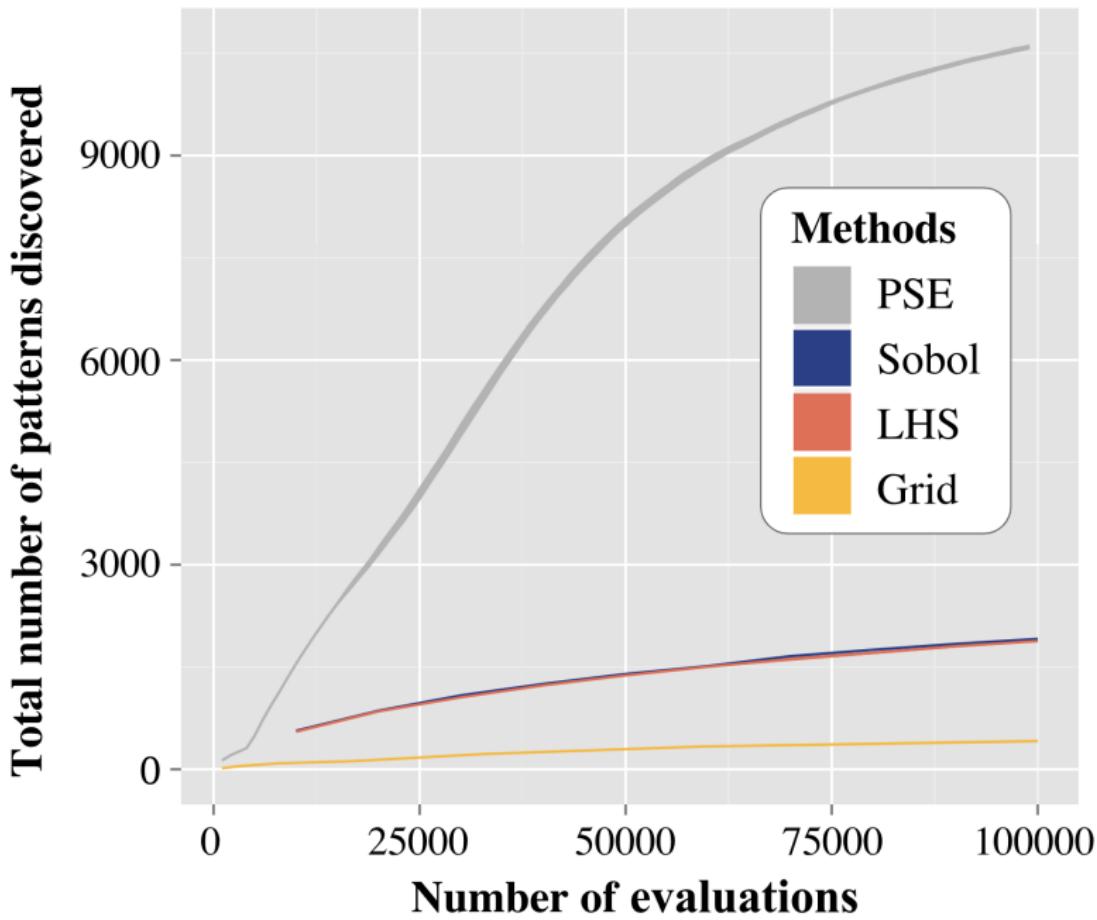


Novelty search

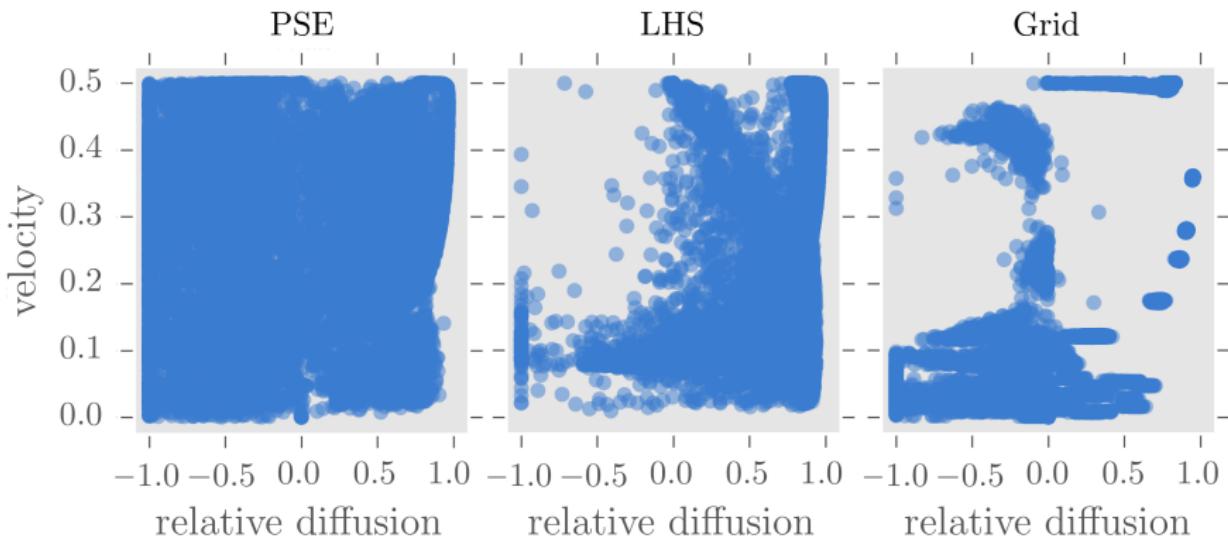
The inputs producing rare patterns have high fitness values.



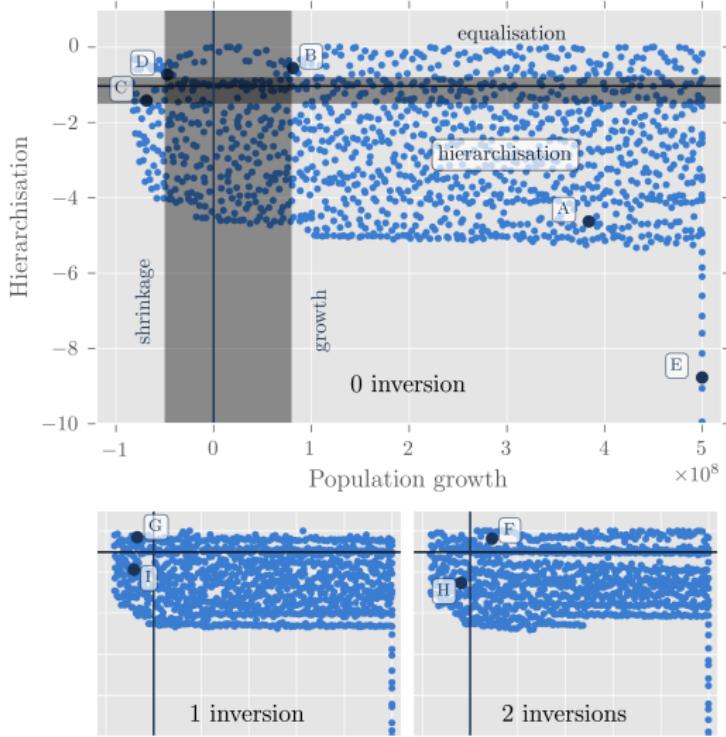
Results



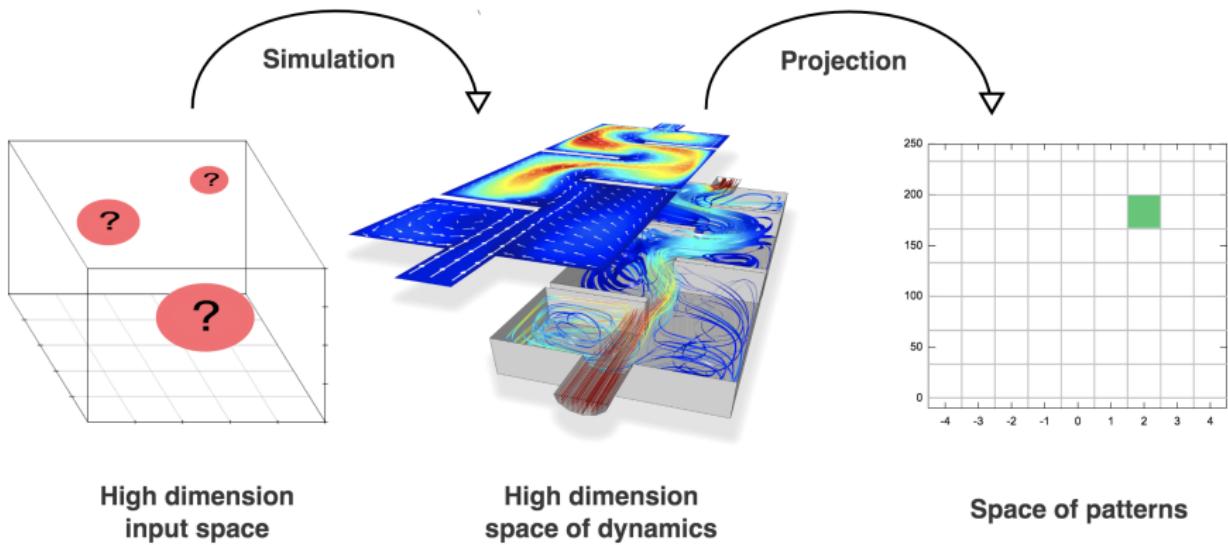
Results



Results

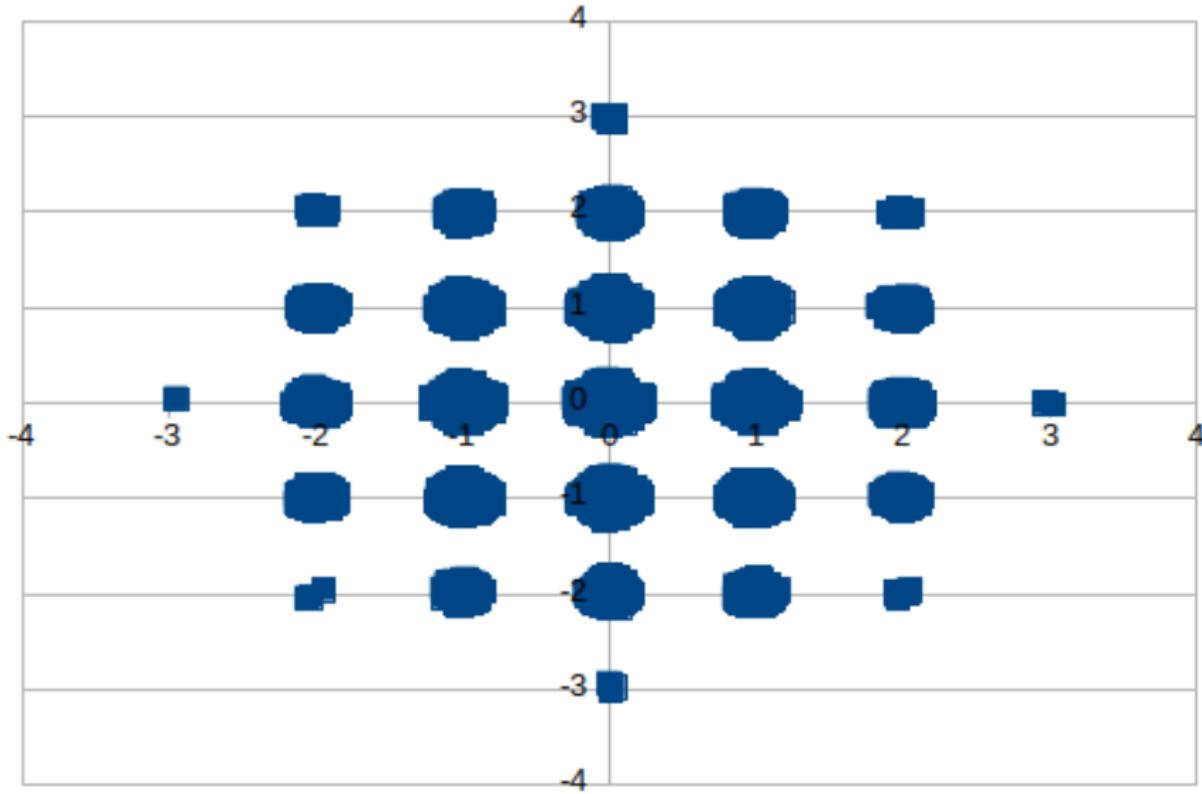


Inverse problem

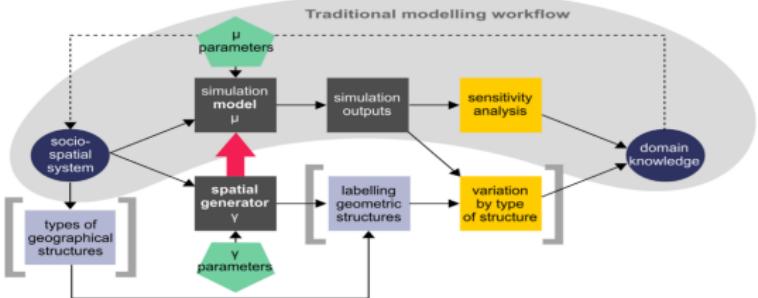


Results : minimising a Rastrigin function

$\Delta \text{ pattern} < \varepsilon$

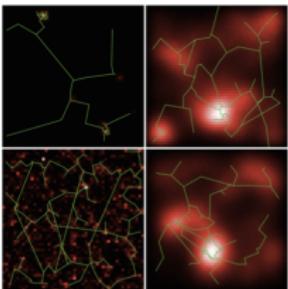


Spatial sensitivity analysis

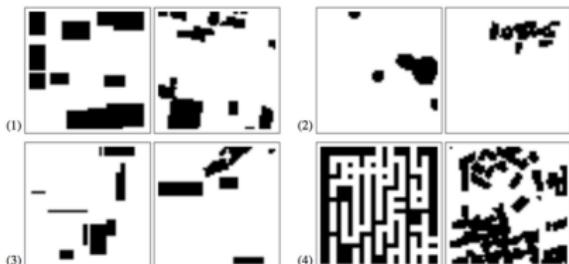


Raimbault, J., Cottineau, C., Le Texier, M., Le Nechet, F., Reuillon, R. (2019). Space Matters: Extending Sensitivity Analysis to Initial Spatial Conditions in Geosimulation Models. *Journal of Artificial Societies and Social Simulation*, 22(4).

Raimbault, J., Perret, J., & Reuillon, R. (2020). A scala library for spatial sensitivity analysis. *GISRUK 2020 Proceedings*, 32.



Raimbault, J. (2019). Second-order control of complex systems with correlated synthetic data. *Complex Adaptive Systems Modeling*, 7(1), 1-19.



Raimbault, J., Perret, J. (2019). Generating urban morphologies at large scales. In *Artificial Life Conference Proceedings* (pp. 179-186).

Model coupling

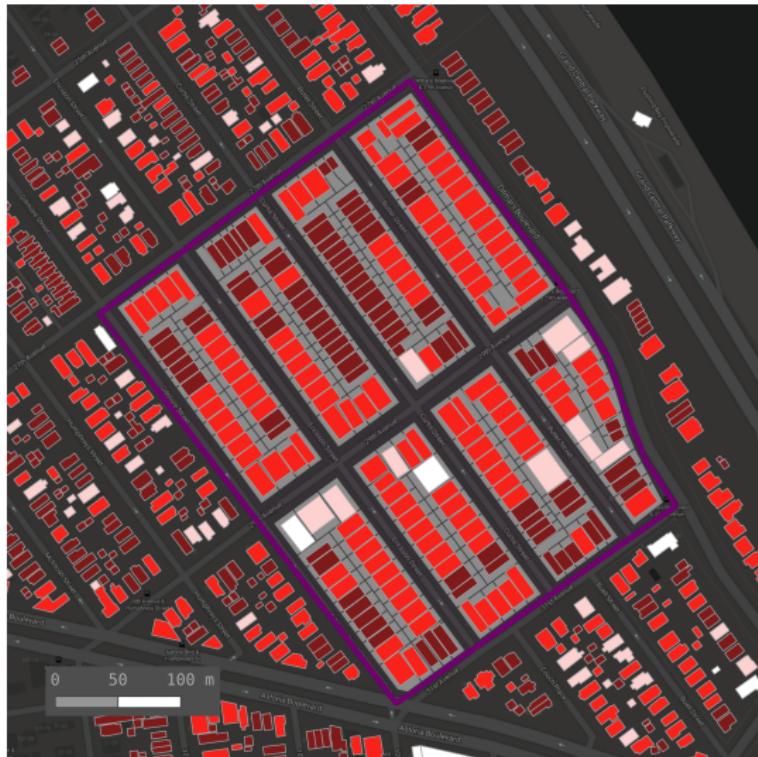
Transport model built using modular and open models

Modèles intégrés :

- MATSim (MATSim Community) for transport
[W Axhausen et al., 2016]
- SPENSER (University of Leeds) for synthetic population
[Spooner et al., 2021]
- QUANT (CASA, University College London) for spatial interactions
[Batty and Milton, 2021]
- spatialdata library (OpenMOLE community) for spatial data
[Raimbault et al., 2020]

Raimbault, J., & Batty, M. (2021). Estimating public transport congestion in UK urban areas with open transport models. GISRUK 2021 Proceedings.

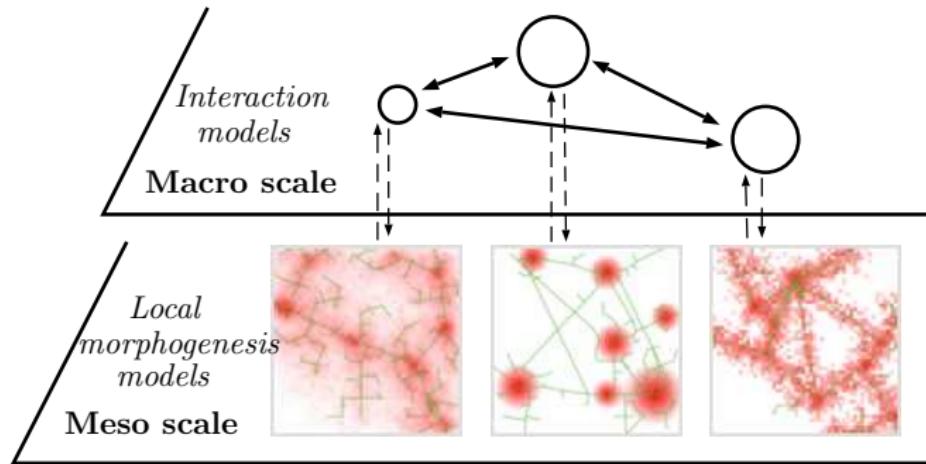
Model coupling: urbanism and heat island effect



SURE project (ASTIG, ISC-PIF, EPIDAPO collaboration)

→ coupling the SimPLU3D model [Brasebin et al., 2017] with an UHI model.

Model coupling: towards multi-scale models



Processes specific to scales, coupling requires a dedicated ontology.

Raimbault, J. (2021). Strong coupling between scales in a multi-scalar model of urban dynamics. arXiv preprint arXiv:2101.12725.

Raimbault, J. (2021). A multiscale model of urban morphogenesis. arXiv preprint arXiv:2103.17241.

Raimbault, J. and Pumain, D. (2023). Innovation dynamics in multi-scalar systems of cities. ALIFE 2023.

Synthesis: OpenMOLE benchmark

Type	Critères	Apache Taverna	Spark	Jupyter	R	Dakota	OpenTURNS	PEST++	OpenMOLE
Mo	Appel d'executable	Green	Green	Green	Green	Green	Green	Green	Green
Mo	Execution de containers	Red	Red	Red	Red	Red	Red	Red	Green
Mo	Transmission transparente de données structurées	Green	Red	Green	Red	Red	Red	Red	Green
Me	Méthodes d'exploration	Red	Red	Red	Green	Green	Green	Green	Green
Me	Échantillonage adaptatif	Red	Red	Red	Green	Green	Green	Green	Green
Me	Optimisation globale	Red	Red	Red	Green	Green	Green	Green	Green
Me	Recherche de diversité	Red	Red	Red	Red	Red	Red	Red	Green
E	Calcul distribué	Green	Green	Red	Green	Green	Red	Green	Green
E	Zero-deploiement	Red	Red	Red	Red	Red	Red	Red	Green
C	Communauté exploration de modèles	Red	Red	Red	Yellow	Green	Green	Green	Green
I	Logiciel installable	Green	Green	Green	Green	Green	Green	Green	Green
I	Service en ligne	Green	Green	Green	Green	Red	Red	Red	Green
I	Langage généraliste	Red	Green	Green	Green	Red	Red	Red	Green
I	Système de workflow	Green	Red	Red	Red	Red	Green	Red	Green
I	Programation Graphique	Green	Red	Red	Red	Red	Red	Red	Red

Use OpenMOLE

The screenshot shows the homepage of the OpenMOLE website. At the top right, there are links for Documentation, Demo, Download, and Community. Below this, there's a large image of a mole wearing a hard hat, with the text "OpenMOLE" overlaid. To the left of the mole, there's a block of text: "Free and open source model exploration software." Below this are two buttons: "Learn more" and "Get Started". To the right of the mole, there's a diagram illustrating the adaptive design of experiments. It shows a series of vertical lines with open circles at various points, representing data points. Three specific points on these lines are highlighted with solid black dots. Callouts explain: "to gain knowledge" points to one dot, "Adaptive design of experiments" points to another, and "on your model dynamics" points to the third. At the bottom right of the page are standard navigation icons.

Documentation Demo Download Community

OpenMOLE

Free and open source model exploration software.

Learn more

Get Started

to gain knowledge

Adaptive design of experiments

on your model dynamics.

- Java executable without installation at <https://openmole.org>, code source to compile at <https://github.com/openmole/openmole>
- Setup an online instance

Help and contribution

The screenshot shows the homepage of the chat.openmole.org website. On the left, there is a sidebar with a dark background containing a list of channels and discussions. The channels listed include Documentation, Chapitre 3, Chapitre 4, Chapitre 2, Presentation ExModelo UR REVERSAAL - L., zombie island, Glasses' team, error netlogo, #dev, #general, #codingcamp, #help, #exmodelo-school, #my-openmole, #shitchat, #explorationMap, #exmodelo-2021, #egi, and #exmodelo-2020. The discussions listed under Documentation include Documentation, Chapitre 3, Chapitre 4, Chapitre 2, and Presentation ExModelo UR REVERSAAL - L.

The main content area has a light gray background. At the top, it says "Welcome to chat.openmole.org". Below that, there's a section titled "Some ideas to get you started" with three buttons: "Create channels", "Join rooms", and "Mobile apps".

The "Create channels" section has a sub-section titled "Desktop apps" with buttons for "Windows", "Linux", and "Mac". The "Join rooms" section has a sub-section titled "Documentation" with a button for "See documentation". The "Mobile apps" section has links for "Google Play" and "App Store".

At the bottom of the main content area, there's a section titled "Welcome to the OpenMOLE community Rocket.Chat!". It provides instructions for asking help on the #help channel, contacting the dev team on the #dev channel, and noting the English language preference. It also mentions the ticket system at ask.openmole.org.

Very reactive chat: <https://chat.openmole.org>

Contribution

The screenshot shows the GitHub repository page for 'openmole-market'. At the top, there are navigation links for Code, Issues, Pull requests, Discussions, Actions, Projects, Wiki, Security, Insights, and Settings. Below the header, there's a search bar with the placeholder 'Type ⌘ to search'. The main content area displays a list of pull requests. One pull request by 'romainreuillon' titled 'Update task name' is highlighted. The list includes other contributions from users like 'R-hello', 'abc', 'ackley', 'ants-extended', 'ants', and 'fire'. On the right side, there's an 'About' section with details about the repository, including its purpose ('OpenMOLE marketplace: complete workflow based on real-world solutions'), documentation link ('next.openmole.org/Documentation...'), and various statistics: Readme, Activity, 9 stars, 9 watching, 7 forks, and a 'Report repository' button.

The screenshot shows the OpenMOLE web interface. At the top, there's a dark header with the 'OpenMOLE' logo and a 'New project' button. Below the header, there are four tabs: 'Empty project', 'From your model', 'From examples' (which is selected), and 'From URL'. The main content area displays a 'script market' with three items: 'Hello World in Python' (Python), 'Hello World in R' (R), and 'Hello World in Scilab' (Scilab). Each item has a small preview box and a language label.

Add thematic example to the script market

<https://github.com/openmole/openmole-market>

eX Modelo Workshop

Model Evaluation & Exploration with OpenMOLE

November 13th and 14th, 2023 Paris - France

Apply

Learning two days workshop (ISC-PIF, November)
<https://workshop.exmodelo.org/>: applications open

R&D and scientific consulting

NOTRE ENTREPRISE

RÉFÉRENCES SCIENTIFIQUES

10+
ans de recherche

50+
communications internationales

100+
supports modélisateur

8
supports de thèse

COOPÉRATION AVEC LA RECHERCHE PUBLIQUE

2
consultants en recherche CNRS

Participation au développement de la plateforme libre d'exploration de modèles

OpenMOLE
the model exploration software

UNE ENTREPRISE ÉTHIQUE

Trempline est une SCIC

- 1 personne = 1 voix
- pas de valorisation du capital
- 2/3 des bénéfices annuels minimum réinvestis

Open-source

Notre expertise est à vendre, pas notre logiciel

<https://trempline.io/>

Summary of OpenMOLE positioning

A qualitative shift in knowledge that can be extracted from a simulation model with model exploration methods.

Application in different disciplines: Geography

[Schmitt et al., 2015][Chérel et al., 2015], Ecological modeling
[Lavallée et al., 2018], epidemiology [Arduin, 2018], etc.

Main characteristics:

- Complementary role of the three axis: methods, computing environments, model embedding.
- Iterative and integrated construction of models and theories.
- Model coupling and reproducibility made easy through the scripting language [Passerat-Palmbach et al., 2017].

References I

-  Arduin, H. (2018).
Modélisation mathématique des interactions entre pathogènes chez l'hôte humain: Application aux virus de la grippe et au pneumocoque.
PhD thesis, Université Paris-Saclay.
-  Batty, M. and Milton, R. (2021).
A new framework for very large-scale urban modelling.
Urban Studies, 58(15):3071–3094.
-  Brasebin, M., Chapron, P., Chérel, G., Leclaire, M., Lokhat, I., Perret, J., and Reuillon, R. (2017).
Apports des méthodes d'exploration et de distribution appliquées à la simulation des droits à bâtir.
In *Spatial Analysis and GEOMatics 2017*.

References II

-  Chérel, G., Cottineau, C., and Reuillon, R. (2015).
 Beyond corroboration: Strengthening model validation by looking for unexpected patterns.
PLoS ONE, 10(9):e0138212.
-  Cottineau, C. (2014).
L'évolution des villes dans l'espace post-soviétique. Observation et modélisations.
PhD thesis, Université Paris 1 Panthéon-Sorbonne.
-  Lavallée, F., Alvarez, I., Dommangeat, F., Martin, S., Reineking, B., and Smadi, C. (2018).
A dynamical model for the growth of a stand of japanese knotweed including mowing as a management technique.
In *Conference on Complex Systems 2018*.

References III

-  Passerat-Palmbach, J., Reuillon, R., Leclaire, M., Makropoulos, A., Robinson, E. C., Parisot, S., and Rueckert, D. (2017). Reproducible large-scale neuroimaging studies with the openmole workflow management system. *Frontiers in neuroinformatics*, 11:21.
-  Pumain, D. (2018). An evolutionary theory of urban systems. In *International and Transnational Perspectives on Urban Systems*, pages 3–18. Springer.
-  Raimbault, J. (2018). Extracting knowledge from simulation models: trends and perspectives from the viewpoint of quantitative geography. In *Conference on Complex Systems 2018*.

References IV



Rimbault, J. (2019).

Second-order control of complex systems with correlated synthetic data.

Complex Adaptive Systems Modeling, 7(1):1–19.



Rimbault, J. (2021a).

A multiscale model of urban morphogenesis.

arXiv preprint arXiv:2103.17241.



Rimbault, J. (2021b).

Strong coupling between scales in a multi-scalar model of urban dynamics.

arXiv preprint arXiv:2101.12725.



Rimbault, J. and Batty, M. (2021).

Estimating public transport congestion in uk urban areas with open transport models.

GIS Research UK (GISRUK).

References V

-  Rimbault, J., Cottineau, C., Le Texier, M., Le Nechet, F., and Reuillon, R. (2019).
Space matters: Extending sensitivity analysis to initial spatial conditions in geosimulation models.
Journal of Artificial Societies and Social Simulation, 22(4).
-  Rimbault, J. and Perret, J. (2019).
Generating urban morphologies at large scales.
In *Artificial Life Conference Proceedings*, pages 179–186. MIT Press
One Rogers Street, Cambridge, MA 02142-1209, USA
journals-info
-  Rimbault, J., Perret, J., and Reuillon, R. (2020).
A scala library for spatial sensitivity analysis.
GISRUK.

References VI

-  Reuillon, R., Schmitt, C., De Aldama, R., and Mouret, J.-B. (2015).
A new method to evaluate simulation models: The calibration profile (cp) algorithm.
Journal of Artificial Societies and Social Simulation, 18(1):12.
-  Schmitt, C., Rey-Coyrehourcq, S., Reuillon, R., and Pumain, D. (2015).
Half a billion simulations: Evolutionary algorithms and distributed computing for calibrating the simpoplocal geographical model.
Environment and Planning B: Planning and Design, 42(2):300–315.
-  Spooner, F., Abrams, J. F., Morrissey, K., Shaddick, G., Batty, M., Milton, R., Dennett, A., Lomax, N., Malleson, N., Nelissen, N., et al. (2021).
A dynamic microsimulation model for epidemics.
Social Science & Medicine, 291:114461.

References VII

-  W Axhausen, K., Horni, A., and Nagel, K. (2016).
The multi-agent transport simulation MATSim.
Ubiquity Press.