On the "Local-to-Global" Issue in Self-Organisation

Chemical Reactions with Custom Kinetic Rates

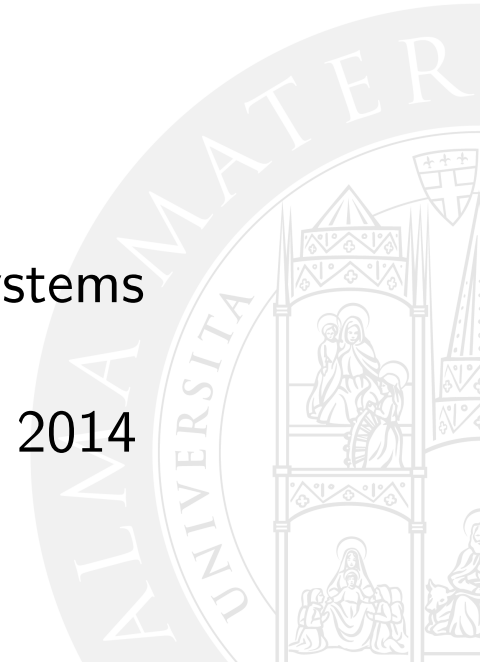
Stefano Mariani

s.mariani@unibo.it

DISI

Alma Mater Studiorum—Università di Bologna

Talk @ Fundamentals of Collective Adaptive Systems (SASO Workshop)
Imperial College, London, UK - 8th September 2014



Outline

Context

- One of the foremost issues in the engineering of *self-organising systems* is the so-called "local-to-global" issue

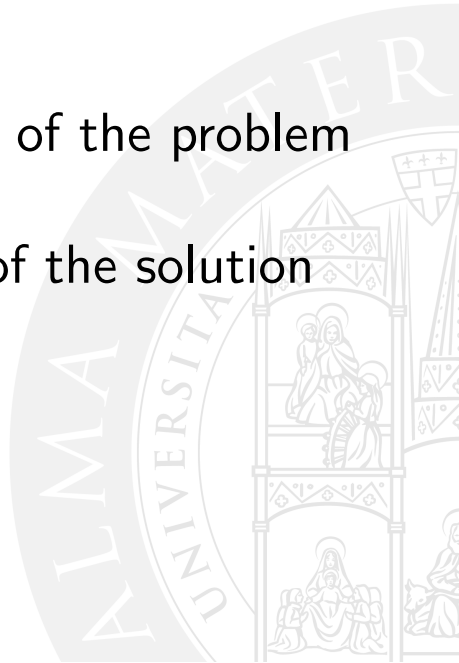
Local-to-global

How to "link" the *local* mechanisms, through which the components of the system interact, to the *emergent, global* behaviour, exhibited by the system as a whole [Beal and Bachrach, 2006]

- Existing approaches to alleviate such issue are mostly based on:
  - simulation [Gardelli et al., 2006]
  - parameter tuning [Gardelli et al., 2009]
  - (approximate) model checking [Casadei and Viroli, 2013]
  - "bio-inspired design patterns" [Fernandez-Marquez et al., 2012]

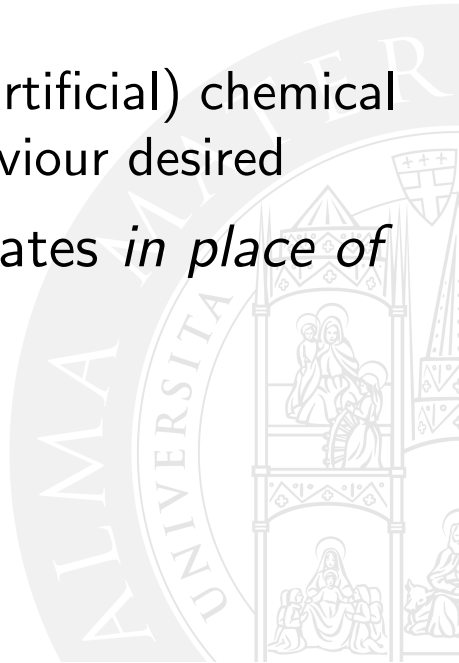
Motivation

- Nevertheless, these approaches may be not enough—*especially if used separately*:
  - simulation may not be able to accurately reproduce real world contingencies
  - parameter tuning may lead to sub-optimal settings
  - model checking may be impractical for the complexity of the problem at hand
  - design patterns give no guarantees about the quality of the solution



Goals

- For these reasons, we propose an integrated approach:
  1. rely on design patterns — design the local mechanisms by implementing *self-organisation primitives* as artificial chemical reactions
  2. go beyond the *law of mass action* [Cardelli, 2008] — engineer custom kinetic rates for such reactions
  3. "simulate-then-tune" — adjust the dynamics of the (artificial) chemical system obtained to achieve the emergent, global behaviour desired
- Also, we remark the benefits of using custom kinetic rates *in place of* the law of mass action [Mariani, 2013]



Outline

Self-organisation "primitives" I

- Bio-inspired design patterns are decomposable into "primitive" mechanisms [Fernandez-Marquez et al., 2012], that is, into self-organisation primitives
- A survey of state-of-art literature (see the paper for citations) led to the following "core" set of primitives—the local mechanisms:

decay	destroys information as time passes
feed	increases information "relevance" (e.g. quantity) according to some kind of feedback mechanism
activation/inhibition	changes information "status" (e.g. attributes, values, etc.) depending on external stimuli
aggregation	fuses information together (e.g. filtering, merging, composing, transforming, etc.)
diffusion	moves information within a topology (e.g. migration, replication, etc.)
repulsion/attraction	drifts apart / approaches information

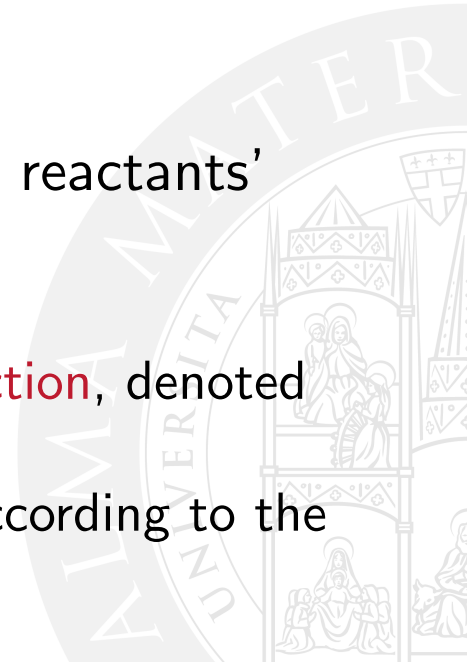
Self-organisation "primitives" II

- ! Diffusion and repulsion/attraction are left out from simulations because they involve spatial aspects which, although can be simulated with our chosen tool, BioPEPA[1], visualisation of their emergent behaviour is best done with tools like NetLogo[2]
- Doing so is part of our ongoing work...

[1] Home page at `http://homepages.inf.ed.ac.uk/jeh/Bio-PEPA/biopepa.html`
[2] Home page at `http://ccl.northwestern.edu/netlogo/index.shtml`

Chemical Reactions Simulation

- BioPEPA [Ciocchetta and Hillston, 2009] is a language and tool for the *simulation of biochemical processes*
- As regards our goals, its most appealing features are:
  - support to *custom kinetic laws* by designing functional rate expressions
  - support to *stoichiometry* ("how many" molecules of a given kind participate) and *role* played by the species (reactant, product, enzyme, etc.) in a given reaction
  - roots in CTMC semantics [Hermanns, 2002]
- Rate expressions are mathematical functions involving reactants' concentrations and supporting:
  - mathematical operators, e.g., `exp` and `log` functions
  - built-in common kinetic laws, e.g., the law of mass action, denoted with the keyword `fMA`
  - *time dependency* through variable `time`, increasing according to the current simulation time step



Outline

Experimental Setting

- We model self-organisation primitives as artificial chemical reactions
- We then encode them in the BioPEPA language, to simulate the emergent, global behaviour achievable
- While doing so, we engineer custom kinetic rates in different ways, comparing BioPEPA plots to investigate how a change in local mechanisms affect the global behaviour

Technical Details

- Gillespie simulation algorithm [Gillespie, 1977]
- x-axis plots the time steps of the simulation
- y-axis plots reactants concentration expressed as units of "molecules"

Outline

Usual fMA-based Rate I

Artificial Chemical Decay

$$data \xrightarrow{r_{decay}} \perp$$

BioPEPA Encoding

```
1 DECAY_CONSTANT = 0.5;
2 r_decay = [fMA(DECAY_CONSTANT)]; // kinetic rate
3 data = (r_decay, 1) <<;          // chemical reaction
```

- Species `data` participates as a *reactant* (`<<`)—thus being consumed
- It participates with *stoichiometry* 1—thus one unit of `data` is involved in `r_decay` chemical reaction
- Reaction rate (`r_decay`) follows the usual law of mass action (`fMA`) [Cardelli, 2008]

Usual fMA-based Rate II

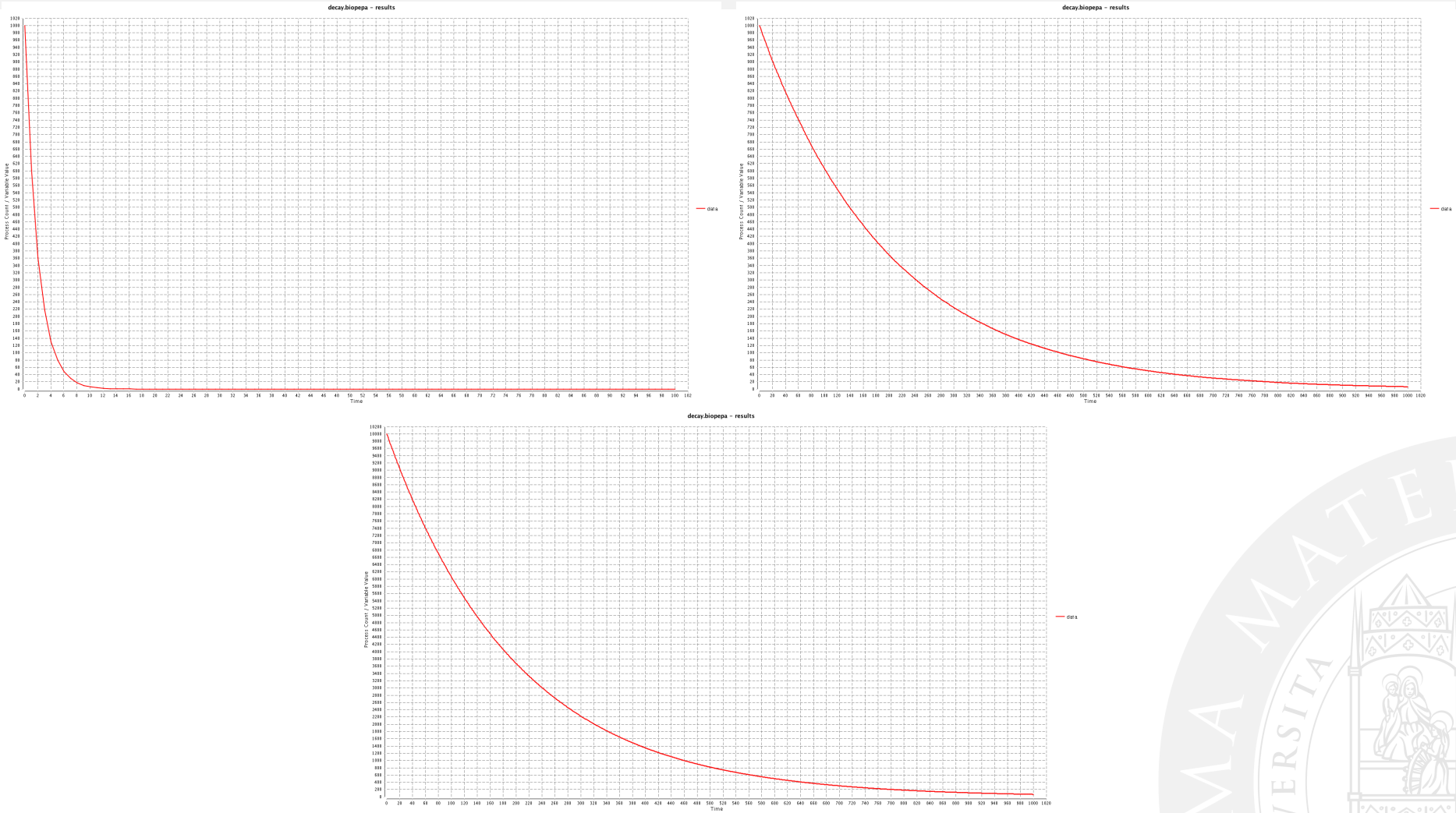


Figure 1 : Rate constant from 0.5 (left plot) to 0.005 (right plot), thus time from 100 steps to 1,000. Data quantity from 1,000 units (right/left plots) to 10,000 (bottom plot), but decay time remained the same (still 1,000).

Usual fMA-based Rate III

- "Fast-then-slow" decay—the emergent, global behaviour
- *Independent* of the quantity of data to decay—compare right plot to bottom plot
- Timing can be tuned by changing the rate constant—compare left plot to right plot

fMA Limitations

- What if such trend is not the best to suit the *application needs*?
- What if the self-organising system to be deployed should display a *different trend*, e.g. an opposite "slow-then-fast" decay? Possibly, also sensitive to the quantity of information to decay?

Flexibility

In an information management scenario, for instance, novel data can be produced/consumed anytime and *when* a given piece of information may become interesting is unknown. There, shifting to a "slow-then-fast" trend is better.

Custom Rate I

BioPEPA Encoding

```
1 DECAY_CONSTANT = 0.5;
2 r_decay = [fMA(DECAY_CONSTANT) + H(data) * time/data];
3 data = (r_decay, 1) <<;
```

- `time` is the BioPEPA variable tracking simulation time steps
- `H(·)` is the Heaviside step function[3]—useful to avoid meaningless negative rates

[3] http://en.wikipedia.org/wiki/Heaviside_step_function

Custom Rate II

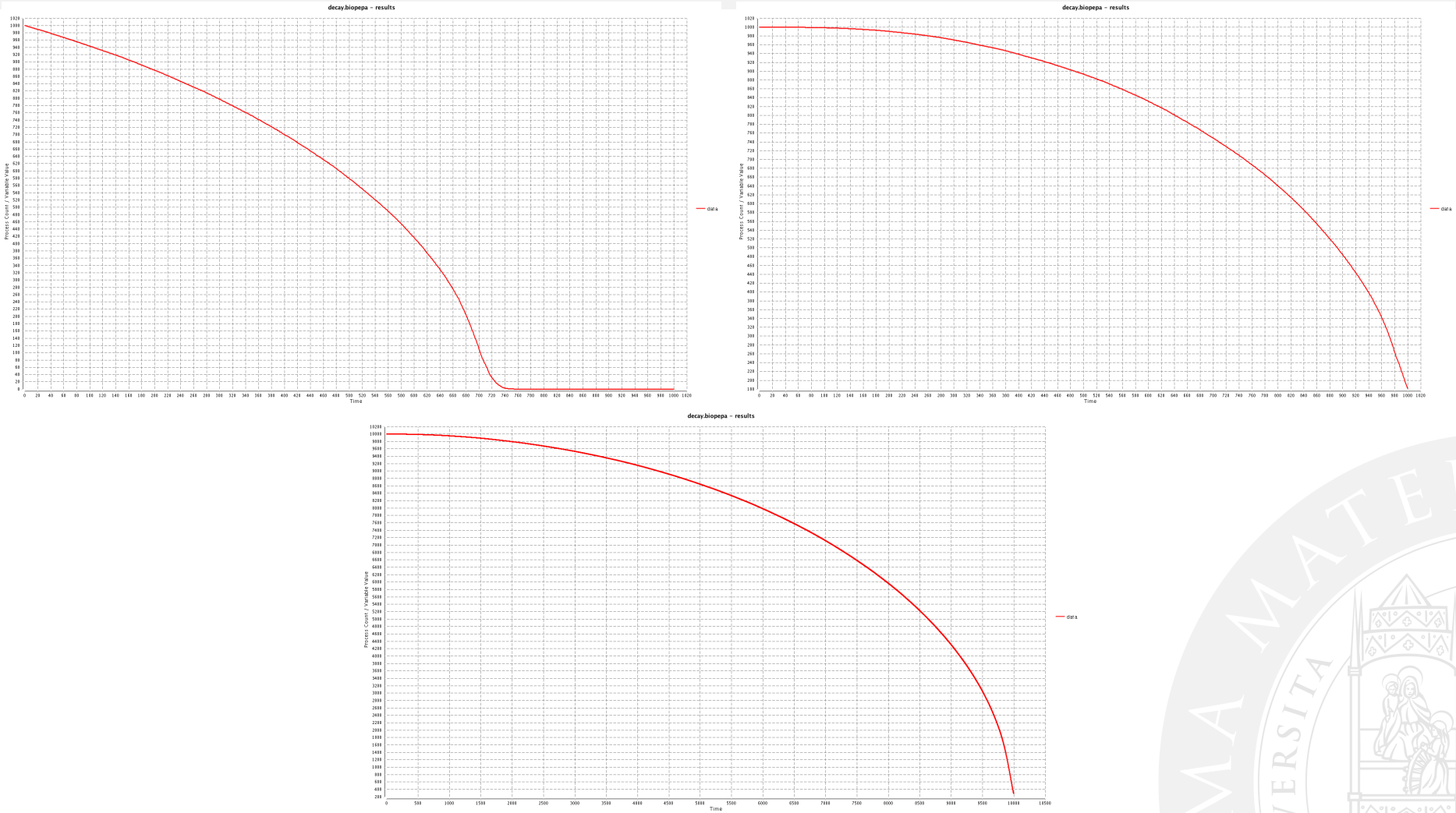


Figure 2 : Rate constant from 0.5 (left plot) to 0.005 (right plot), thus time from ≈ 700 to over $1,000$. Data quantity from $1,000$ (left/right plots) to $10,000$ (bottom plot), time too increased proportionally from 1000 to over $10,000$ time steps.

Custom Rate III

- *Opposite*, "slow-then-fast" trend
- Decreasing rate constant *still* leads to a delay in decay completion—compare left plot to right plot
- Changing the quantity of `data` to decay *now affects* decay time proportionally—compare right plot to bottom plot

"Local-to-Global"

- This dramatic behaviour twist is due to the new factors we added to the kinetic rate expression: *direct proportionality* to `time` and *inverse proportionality* to `data`
- Notice to keep independency of data quantity, we can simply remove factor $\frac{1}{\texttt{data}}$ from the rate expression.

Mitigating the issue

Adding/removing factors to the local mechanism (the "law" manipulating each `data` item) leads to a well-defined change in the emergent, global behaviour achieved (the evolution of the whole population of `data` items), greatly enhancing *flexibility* & *controllability*.

Outline

Usual fMA-based Rate

Artificial Chemical Feed

$$data + food \xrightarrow{r_{feed}} data + data$$

BioPEPA Encoding

```
1 FEED_CONSTANT = 0.5;
2 r_feed = [fMA(FEED_CONSTANT)];
3 // same rate name ==> same chemical reaction
4 data = (r_feed, 1) >>; // product (1 unit produced)
5 food = (r_feed, 1) <<; // reactant (1 unit consumed)
```

- "Fast-then-slow" feed
- A lower rate leads to a slower feeding process
- A higher quantity of data *does not affect* time taken to complete the feeding process

Custom Rate

BioPEPA Encoding

```
1 FEED_CONSTANT = 0.5;
2 r_feed = [fMA(FEED_CONSTANT) + H(food) * time/food];
3 // same rate name ==> same chemical reaction
4 data = (r_feed, 1) >>; // product (1 unit produced)
5 food = (r_feed, 1) <<; // reactant (1 unit consumed)
```

- *Opposite*, "slow-then-fast" trend
- As for the fMA-only rate, lower rate leads to a slower feeding process
- As for the fMA-only rate, higher quantity of data *does not affect* time taken to complete the feeding process

fMA Vs. Custom Rate

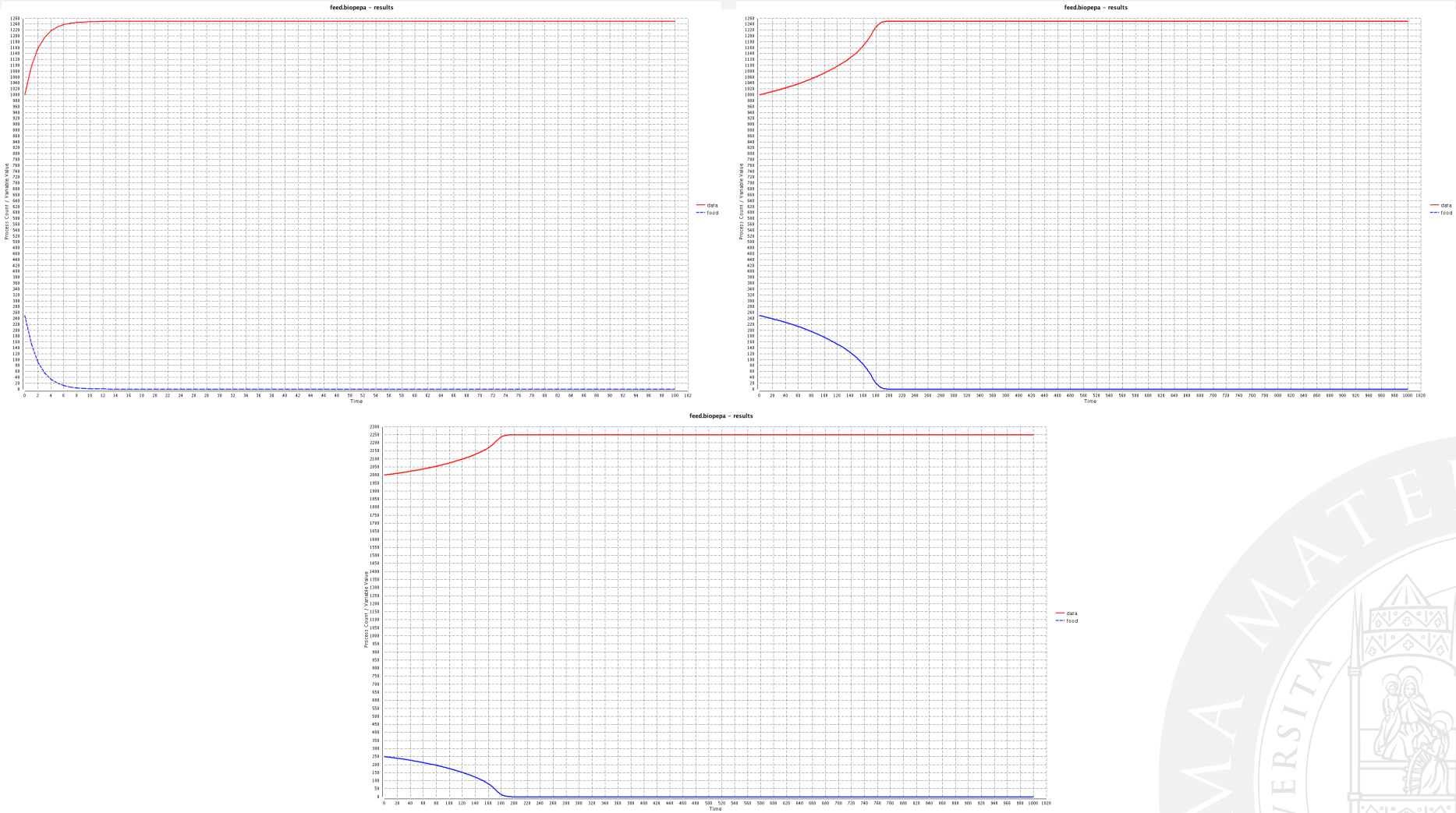


Figure 3 : `fMA`-only rate (left plot) is sensitive to rate constant change and to data quantity likewise decay; custom kinetic rate (right and bottom plots) too (bottom plot has twice the data of right plot, but saturation time is the same), due to its inverse proportionality to `food`, not `data`. Changing `food` affects saturation time of both rate expressions.

Outline

Usual fMA-based Rate I

Artificial Chemical Activation/Inhibition

$$data + on \xrightarrow{r_{activation}} on + data\_on$$
$$data + off \xrightarrow{r_{inhibition}} off + data\_off$$

BioPEPA Encoding

```
1 ACTIVATION_CONSTANT = 0.5;
2 r_activation = [fMA(ACTIVATION_CONSTANT)];
3 data = (r_activation, 1) <<;
4 on = (r_activation, 1) (+); // activator enzyme (not consumed)
5 data_on = (r_activation, 1) >>;
6 // inhibition ==> replace (+) with (-)
```

Usual fMA-based Rate II

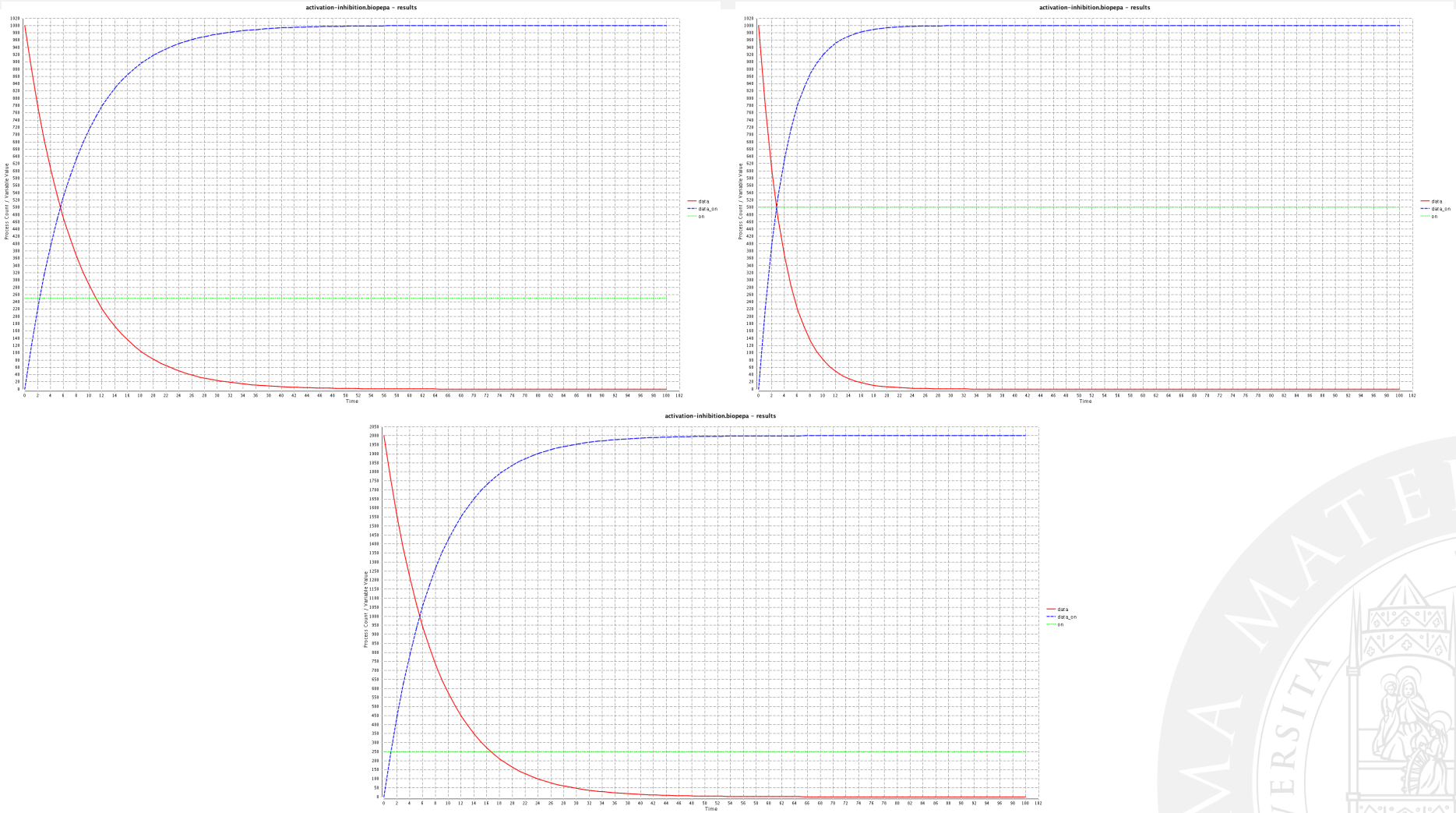


Figure 4 : In right plot activator enzyme quantity (`on`) is twice that of left plot, causing a faster activation process (from time 6 to time 3). In bottom plot `data` quantity is twice that of other plots, but activation time is the same as that of left plot, thus here `data` quantity does not affect timing. Acting on rate constant speeds up or slows down the process as usual.

Usual fMA-based Rate III

- "Fast-then-slow" activation
- *Dependent* on the quantity of activator enzyme—compare right plot to bottom plot
- *Independent* of the quantity of reactant to activate—compare left plot to bottom plot
- Timing can be tuned by changing the rate constant

Species' Role

Notice we are comparing `food` with `data`, not with `on`. Although `food` and `on` play a similar "role" in the artificial chemical reaction – "activators" – they have a completely different "chemical nature": `food` is a *reactant*, whereas `on` is an *enzyme*.

Custom Rate I

BioPEPA Encoding

```
1 ACTIVATION_CONSTANT = 0.5;
2 r_activation = [fMA(ACTIVATION_CONSTANT) +
3                 H(data) * time/data_on];
4 data = (r_activation, 1) <<;
5 on = (r_activation, 1) (+); // activator enzyme (not consumed)
6 data_on = (r_activation, 1) >>;
```

Custom Rate II

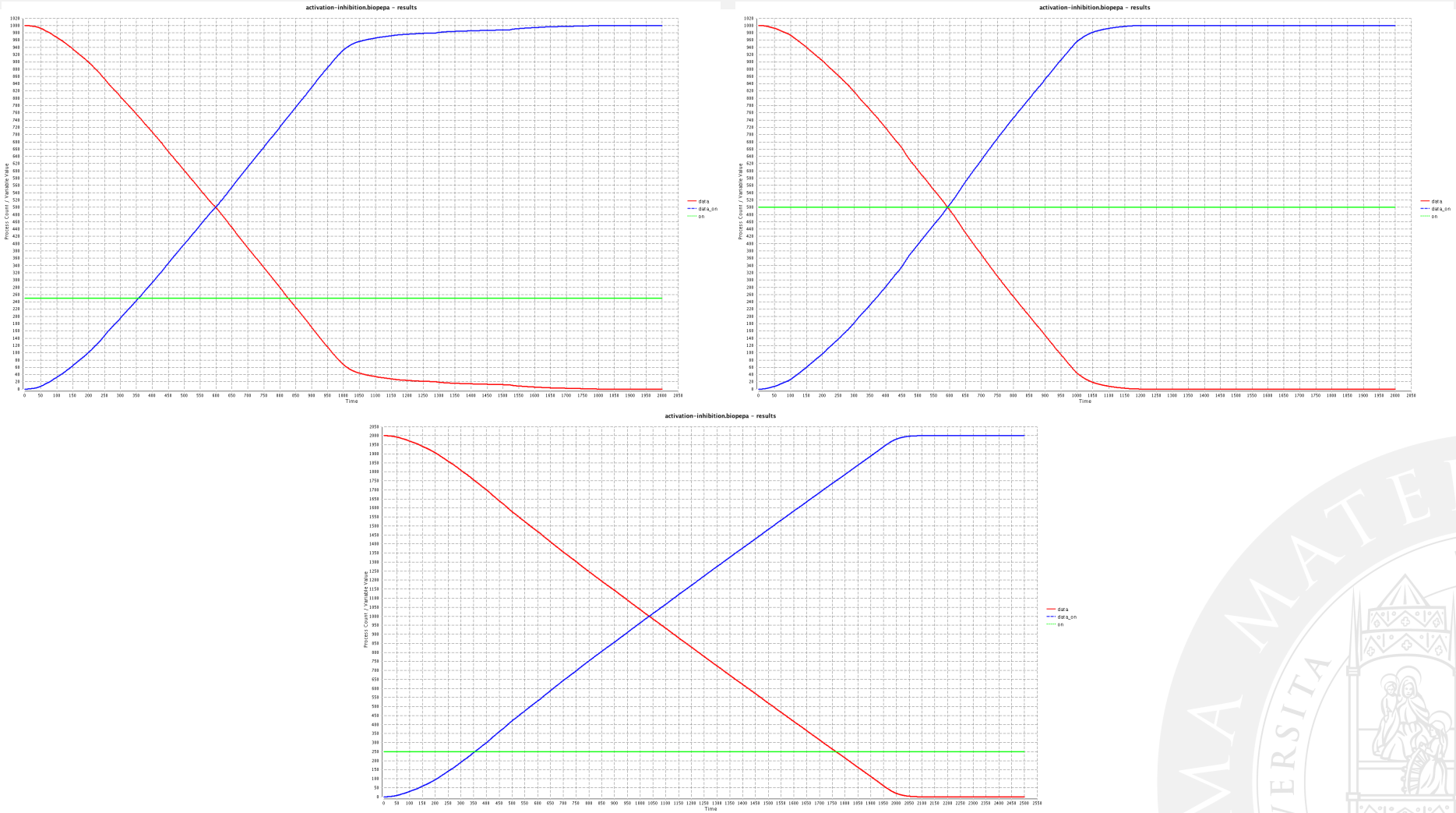


Figure 5 : Increasing the on enzyme quantity no longer affects the activation process time (right plot has twice on than left plot, but "crossing" time step is still ≈ 600), whereas increasing data to activate does (bottom plot has twice data than left plot, thus time taken until crossing point increased to ≈ 1050). Acting on the rate constant speeds up or slows down

Custom Rate III

- Different, "more linear" trend
- Dependency on the quantity of the activator enzyme (`on`) *is lost*
- *Direct proportionality* to the quantity of data to undergo the activation process

More Flexibility & Controllability

- Notice not only the emergent, global behaviour changed, but also sensitivity to the parameters involved in kinetic rate computation
- Also, we put in a kinetic rate computation a *product* of the artificial chemical reaction in process, which is something impossible to find in (real-world) "chemistry"

Custom Rate IV

BioPEPA Encoding

```
1 ACTIVATION_CONSTANT = 0.5;
2 r_activation = [fMA(ACTIVATION_CONSTANT) +
3                 H(data) * time/data];
4 data = (r_activation, 1) <<;
5 on = (r_activation, 1) (+); // activator enzyme (not consumed)
6 data_on = (r_activation, 1) >>;
```

Custom Rate V

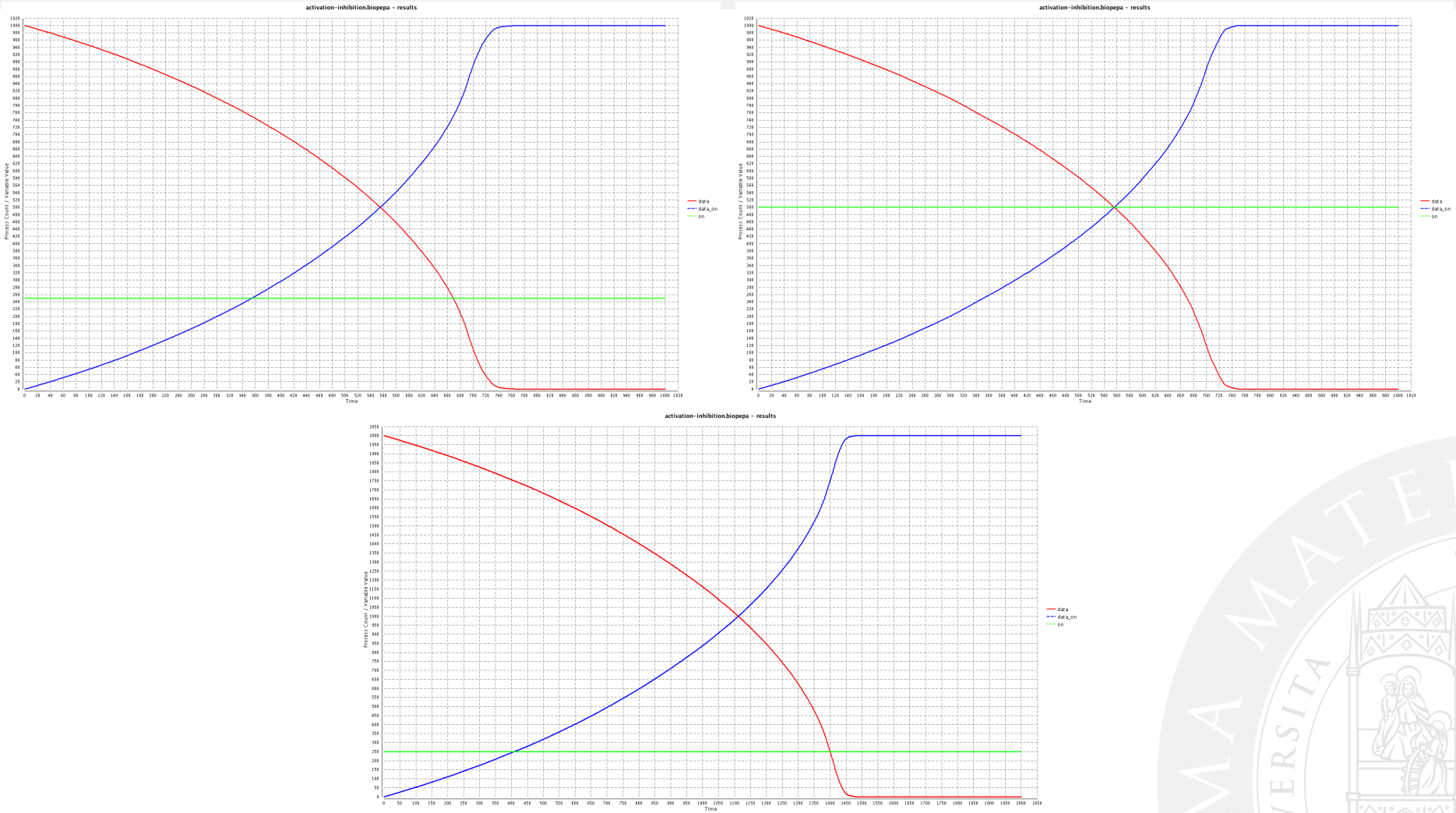


Figure 6 : Increasing on enzyme quantity does not affect the activation process time (right plot has twice than left plot, but crossing time step is still ≈ 560), whereas increasing data to activate does (bottom plot has twice than left plot, thus time taken until crossing point increased to over 1100). Acting on rate constant speeds/slows up/down the process as usual.

Custom Rate VI

- *Opposite*, "slow-then-fast" activation
- *Independent* of the enzyme quantity—same as Figure 5
- *Directly proportional* to `data` quantity—same as Figure 5

Outline

Usual fMA-based Rate I

Artificial Chemical Aggregation

$$part^1 + part^2 \xrightarrow{r_{aggregation}} whole$$

BioPEPA Encoding

```
1 AGGREGATION_CONSTANT = 0.0005;
2 r_aggregation = [fMA(AGGREGATION_CONSTANT)];
3 part1 = (r_aggregation, 1) <<;
4 part2 = (r_aggregation, 1) <<;
5 whole = (r_aggregation, 1) >>;
```

Usual fMA-based Rate II

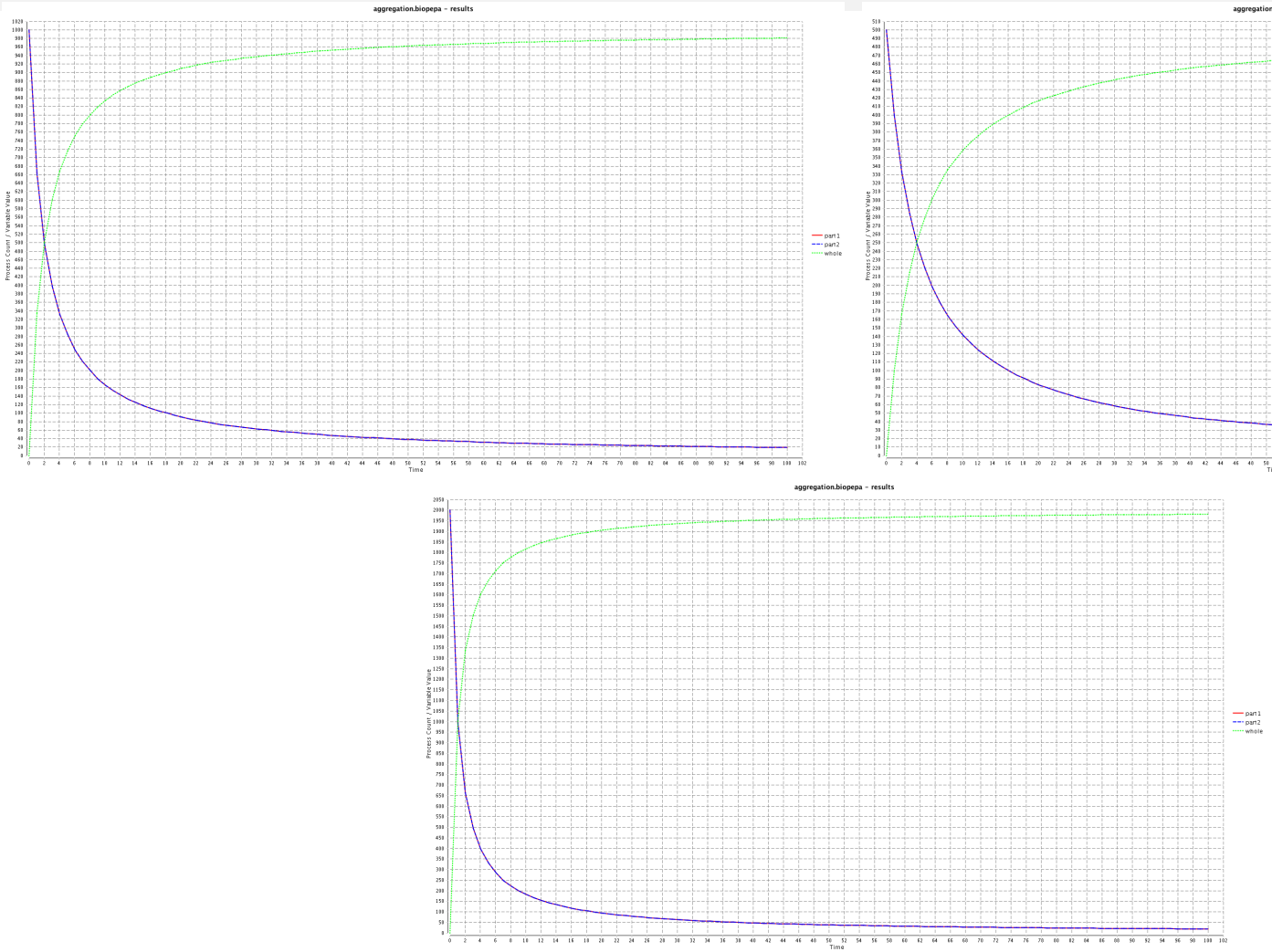


Figure 7 : In right plot reactants $\texttt{part}^1/\texttt{part}^2$ are half w.r.t. left plot, causing a slower aggregation process (step 2 in left plot, step 4 in right plot). In bottom plot instead, their quantity is twice that of the first (2000 units), thus aggregation time faster (half time w.r.t. left plot). Acting on the rate constant speeds/slows up/down the process as usual.

Usual fMA-based Rate III

- "Fast-then-slow" aggregation
- Increasing (decreasing) $\texttt{part}^1/\texttt{part}^2$ decreases (increases) time taken to aggregate parts into `whole`—opposite w.r.t. Figure 5

Custom Rate I

BioPEPA Encoding

```
1 AGGREGATION_CONSTANT = 0.0005;
2 r_aggregation = [fMA(AGGREGATION_CONSTANT) +
3                  H(part1) * H(part2) * time/whole];
4 part1 = (r_aggregation, 1) <<;
5 part2 = (r_aggregation, 1) <<;
6 whole = (r_aggregation, 1) >>;
```

Custom Rate II

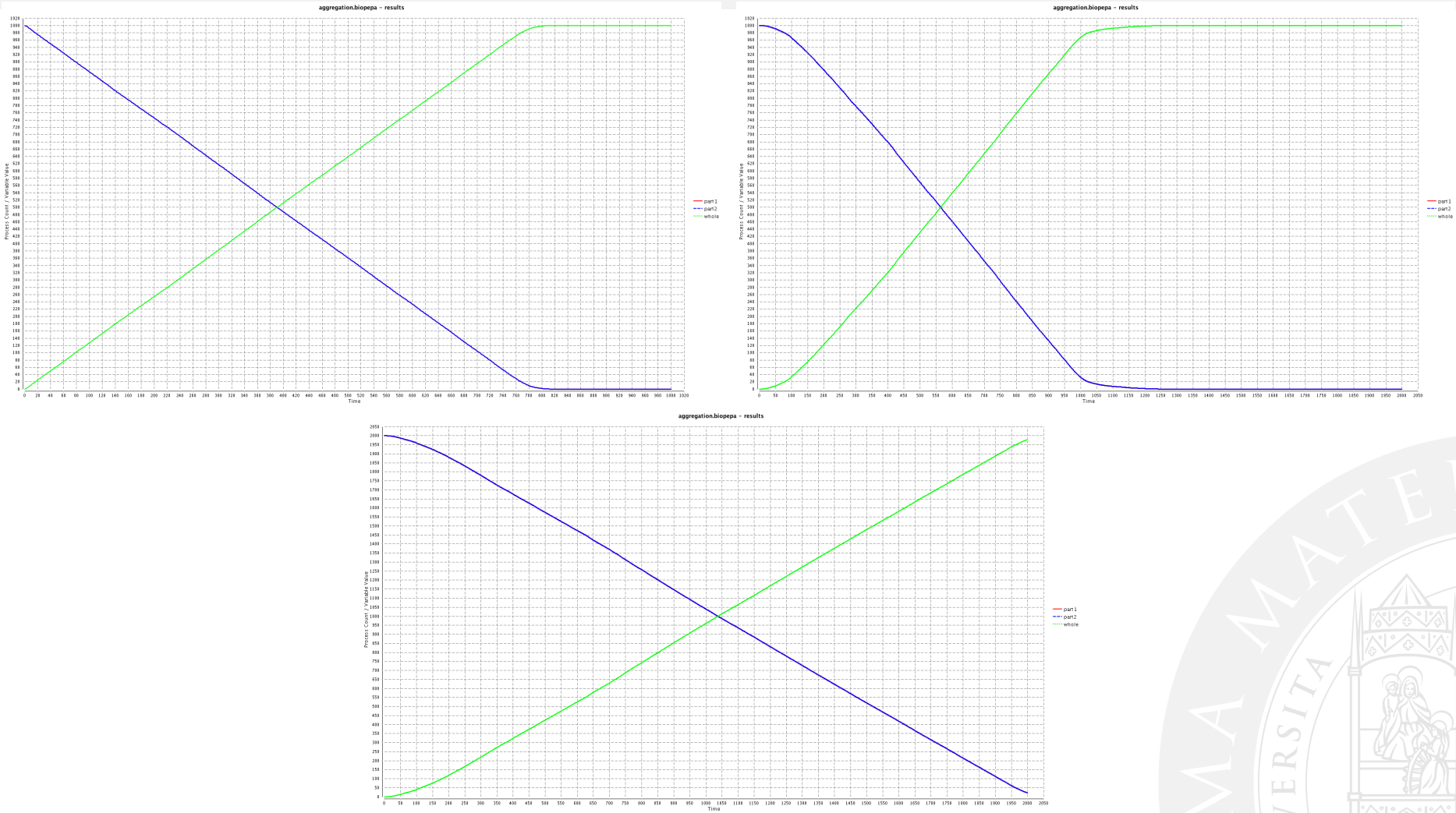


Figure 8 : Now increasing reactants quantity (bottom plot has twice "parts" than right plot) increases time taken to complete aggregation (from 550 time steps in right plot, to 1050 in bottom plot). Instead, acting on the rate constant speeds/slows up/down the process as usual ("crossing point" from ≈ 390 in left plot, to 550 in right plot).

Custom Rate III

BioPEPA Encoding

```
1 AGGREGATION_CONSTANT = 0.0005;
2 r_aggregation = [fMA(AGGREGATION_CONSTANT) +
3                  H(part1) * H(part2) * time/part1];
4 part1 = (r_aggregation, 1) <<;
5 part2 = (r_aggregation, 1) <<;
6 whole = (r_aggregation, 1) >>;
```

Custom Rate IV

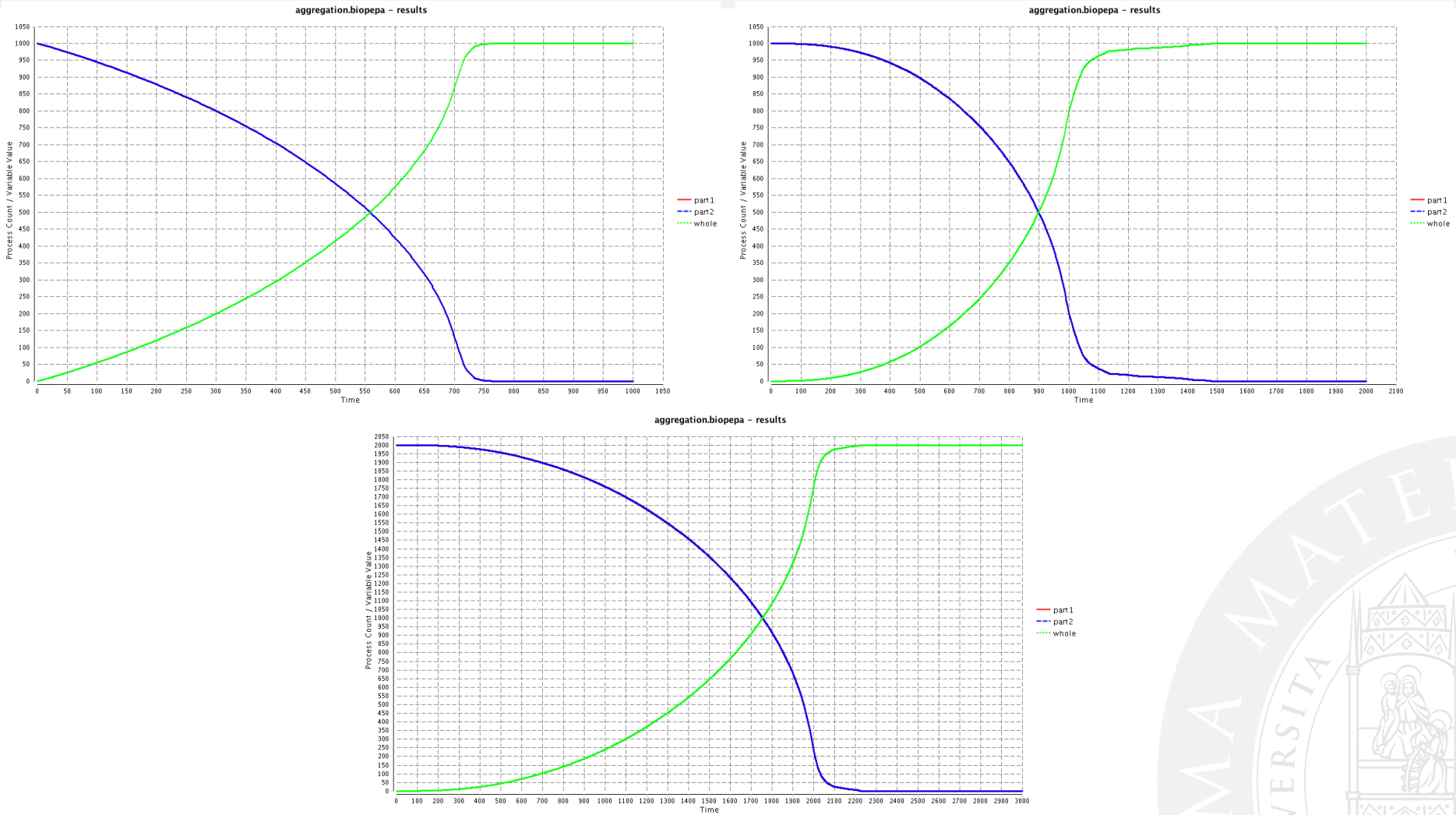


Figure 9 : Parameters of the local mechanism (the aggregation reaction) have the same effect on the emergent, self-organising behaviour achieved as described for previous custom rate. In particular, right plot has a slower rate than left plot, thus time scale almost doubled, and bottom plot has twice parts than right plot, thus, again, time scale almost doubled.

Custom Rate V

- The exhibited trends and sensitivity to local mechanisms' parameters are mostly the same as that of custom activation
- This should not surprise the reader: being the "role" played by `whole` in aggregation primitive the same as that played by `data_on` in activation – that is, *products* of the reaction – as well as the role played by $\texttt{part}^1$ and $\texttt{part}^2$ in aggregation the same as that played by `data` in activation – that is, *reactants* – we can expect for them both the same global behaviour to emerge and the same sensitivity to parameters

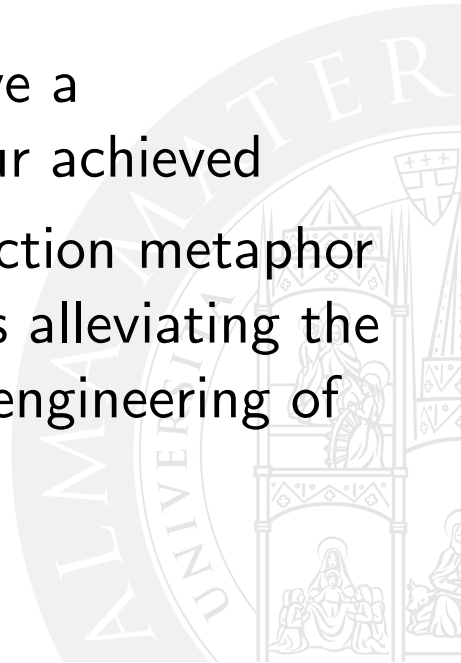
Predictability

This is another nice property of custom kinetic rates in artificial chemical reactions, making predictions about the global, emergent behaviour achievable easier to do.

Outline

Conclusion

- Novel approach in dealing with the "local-to-global" issue in engineering self-organising systems:
  1. model self-organisation primitives as artificial chemical reactions
  2. design custom kinetic rates
  3. adjust rates' parameters according to the emergent, global behaviour desired
- Factors chosen for custom kinetic rate expressions have a well-defined, controllable effect on the global behaviour achieved
- This is made possible by adoption of the chemical reaction metaphor while implement self-organisation primitives, and helps alleviating the "local-to-global" issue, ultimately leading to a better engineering of self-organising behaviours



Ongoing Work

- Extending the pool of self-organisation primitives considered to those implying spatial aspects—e.g. diffusion and repulsion/attraction
- Considering also self-organisation "design patterns", that is, more complex behaviours obtainable by *composition* of self-organisation primitives [Fernandez-Marquez et al., 2012]

Thanks

Thank you for your attention :)

(Friendly) Questions are welcome ;)

References I

Beal, J. and Bachrach, J. (2006).
Infrastructure for engineered emergence on sensor/actuator networks.
Intelligent Systems, IEEE, 21(2):10–19.

Cardelli, L. (2008).
On process rate semantics.
Theoretical computer science, 391(3):190–215.

Casadei, M. and Viroli, M. (2013).
Toward approximate stochastic model checking of computational fields for pervasive computing systems.
In Pitt, J., editor, *Self-Adaptive and Self-Organizing Systems Workshops (SASOW)*, pages 199–204. IEEE CS.
2012 IEEE Sixth International Conference (SASOW 2012), Lyon, France, 10-14 September 2012. Proceedings.

Ciocchetta, F. and Hillston, J. (2009).
Bio-PEPA: A framework for the modelling and analysis of biological systems.
Theoretical Computer Science, 410(33–34):3065 – 3084.

References II

Fernandez-Marquez, J., Marzo Serugendo, G., Montagna, S., Viroli, M., and Arcos, J. (2012).
Description and composition of bio-inspired design patterns: a complete overview.
Natural Computing, pages 1–25.

Gardelli, L., Viroli, M., and Omicini, A. (2006).
On the role of simulations in engineering self-organising MAS: The case of an intrusion detection system in TuCSoN.
In *Engineering Self-Organising Systems*, volume 3910 of *LNAI*, pages 153–168. Springer.

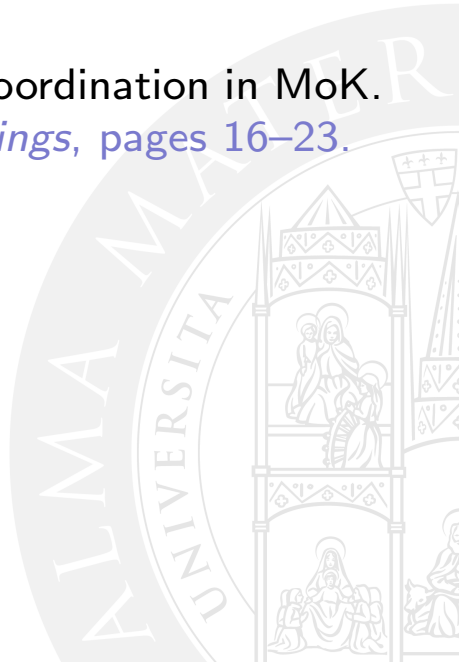
Gardelli, L., Viroli, M., and Omicini, A. (2009).
Combining simulation and formal tools for developing self-organizing MAS.
In Uhrmacher, A. M. and Weyns, D., editors, *Multi-Agent Systems: Simulation and Applications*, Computational Analysis, Synthesis, and Design of Dynamic Systems, chapter 5, pages 133–165. CRC Press.

Gillespie, D. T. (1977).
Exact stochastic simulation of coupled chemical reactions.
The Journal of Physical Chemistry, 81(25):2340–2361.

References III

Hermanns, H. (2002).
Interactive Markov chains: and the quest for quantified quality.
Springer-Verlag.

Mariani, S. (2013).
Parameter engineering vs. parameter tuning: the case of biochemical coordination in MoK.
In *From Objects to Agents*, volume 1099 of *CEUR Workshop Proceedings*, pages 16–23.
Sun SITE Central Europe, RWTH Aachen University.



On the "Local-to-Global" Issue in Self-Organisation

Chemical Reactions with Custom Kinetic Rates

Stefano Mariani

s.mariani@unibo.it

DISI

Alma Mater Studiorum—Università di Bologna

Talk @ Fundamentals of Collective Adaptive Systems (SASO Workshop)
Imperial College, London, UK - 8th September 2014