CSIC

CONSEJO SUPERIOR DE INVESTIGACIONES CIENTÍFICAS

100 CENTENARIO MBG MISIÓN BIOLÓGICA DE GALICIA

UNIVERSITÄT BONN

Benchmarking parallel optimization methods in computational systems biology

David R. Penas[1], Stephan Grein[2], Daniel Weindl[3], Julio R. Banga[1], Jan Hasenauer[2]

(1) *Computational Biology Lab, MBG-CSIC, Pontevedra 36143, Galicia, Spain*

(2) *Life and Medical Sciences (LIMES) Institute, Rheinische Friedrichs-Wilhelms-Universität Bonn, 53115, Bonn, Germany*

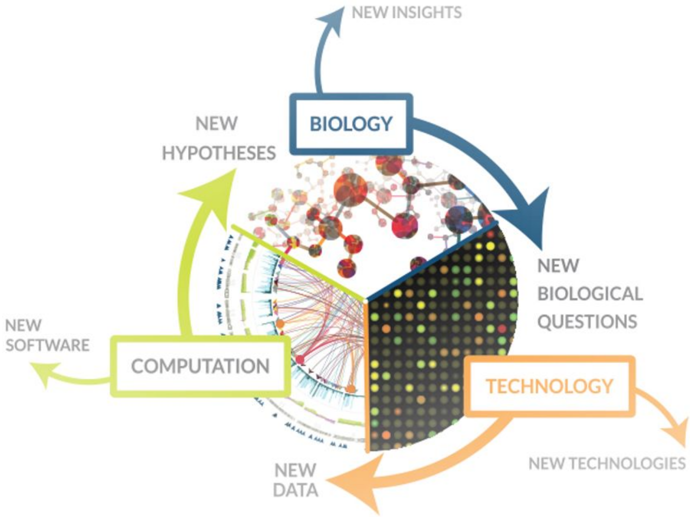
(3) *Computational Health Center, Helmholtz Zentrum München Deutsches Forschungszentrum für Gesundheit und Umwelt (GmbH), Neuherberg, Germany*

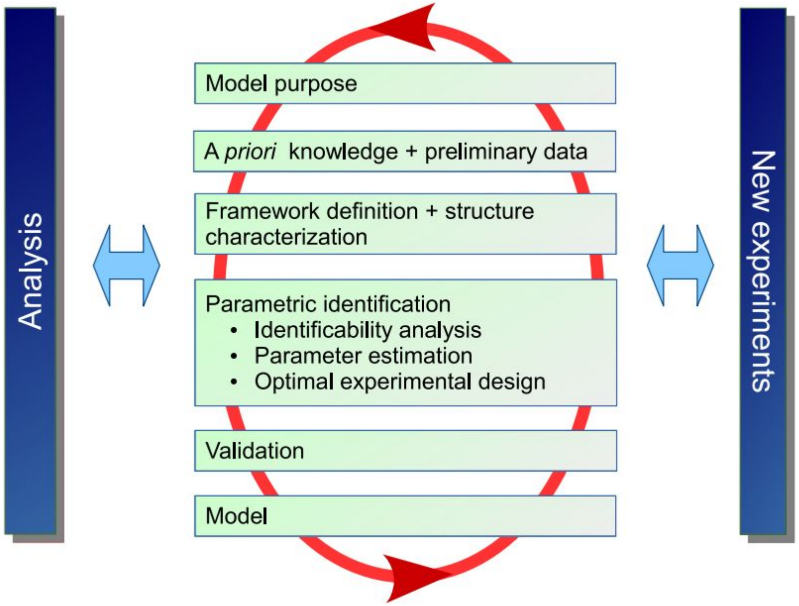
Large-scale dynamic models in systems biology

Systems Biology:

New knowledge about complex biological systems by combining:

- experimental data
- mathematical modeling
- advanced computational techniques





Analysis

Model purpose

A *priori* knowledge + preliminary data

Framework definition + structure characterization

Parametric identification
- Identificability analysis
- Parameter estimation
- Optimal experimental design

Validation

Model

New experiments

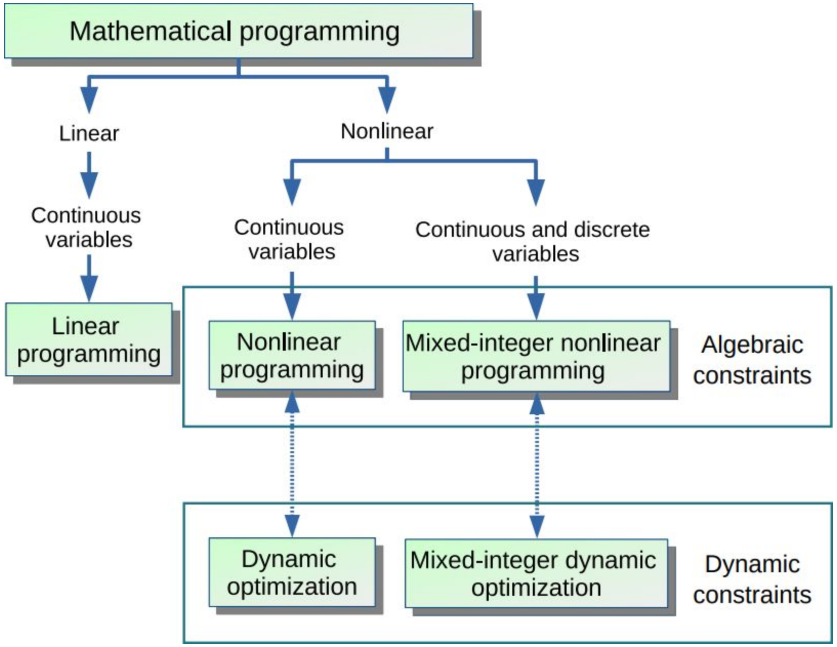
Model calibration or parameter estimation

They can be formulated as mathematical optimization problems:

- finding the parameters of a dynamic model that give the best fit to a set of time-series experimental data
- minimizing a cost function that measures the goodness of this fit

Class of optimization problems?

- RMSE
- NLP-DO
  - non-convex
  - bound constraints
  - multiple local optima
  - NP-hard
- Solvers
  - metaheuristics
  - multistart strategies
- HPC techniques



Mathematical programming

Linear

Nonlinear

Continuous variables

Continuous variables

Continuous and discrete variables

Linear programming

Nonlinear programming

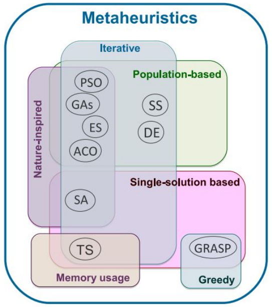
Mixed-integer nonlinear programming

Algebraic constraints

Dynamic optimization

Mixed-integer dynamic optimization

Dynamic constraints



Systematic comparison of optimizers

- Comparison of different solvers, executed on a parallel machine

- The main question we want to address: **Given a set of `P` problems and a set of `S` solvers, and given pre-specified parallel computational resources (`MP` machine processors and a budget of computational time, `T` hours), which are the most efficient and robust solvers?**

- **Efficient:** faster in getting the best solution

- **Robust:** able to solve most problems within the computational budget defined by `MP*T`

Parallel optimization methods

- **pyPESTO: Parameter EStimation TOolbox for python**

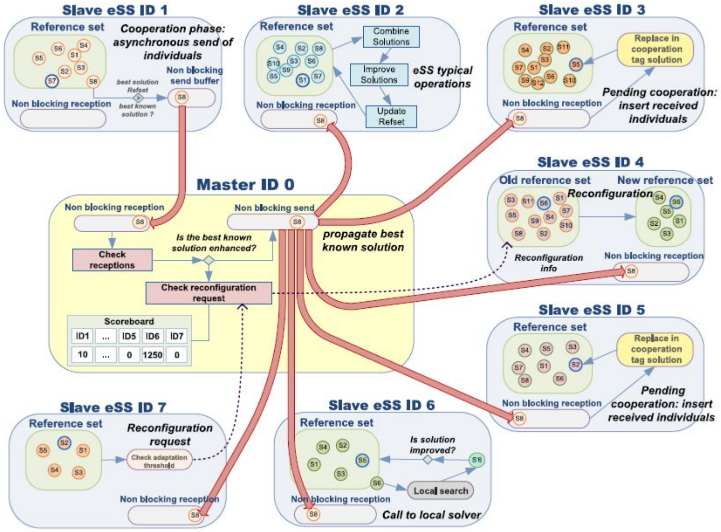
S. Leonard, D. Weindl, and J. Hasenauer (2021)."Efficient gradient-based parameter estimation for dynamic models using qualitative data. *Bioinformatics* 37. 23: 4493-4500.

- **parPE: parameter estimation for PEtab**

L. Schmiester, Y. Schälte, F. Fröhlich, J. Hasenauer, and D. Weindl. (2019) Efficient parameterization of large-scale dynamic models based on relative measurements. *Bioinformatics* 36, 594–602.

- **saCeSS: parameter estimation for PEtab**

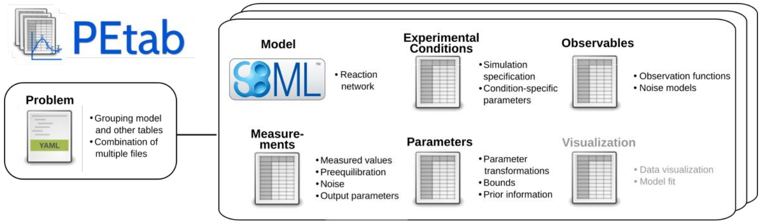
D.R. Penas, P. González, J.A. Egea, R. Doallo, and J.R. Banga, (2017). Parameter estimation in large-scale systems biology models: a parallel and self-adaptive cooperative strategy. *BMC bioinformatics*, *18*(1), 1-24.

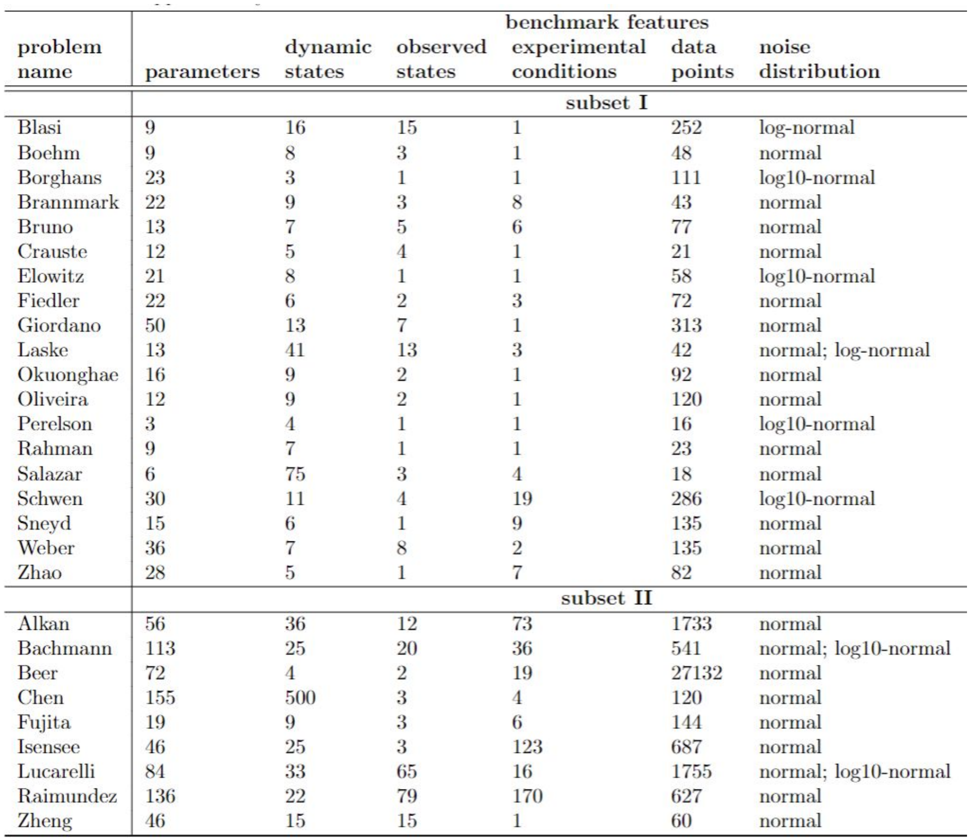


ID1	...	ID5	ID6	ID7
10	...	0	1250	0

PEtab benchmarks considered

- **PEtab** - a data format for specifying parameter estimation problems in systems biology.





	benchmark features					
problem name	parameters	dynamic states	observed states	experimental conditions	data points	noise distribution
				subset I		
Blasi	9	16	15	1	252	log-normal
Boehm	9	8	3	1	48	normal
Borghans	23	3	1	1	111	log10-normal
Brannmark	22	9	3	8	43	normal
Bruno	13	7	5	6	77	normal
Crauste	12	5	4	1	21	normal
Elowitz	21	8	1	1	58	log10-normal
Fiedler	22	6	2	3	72	normal
Giordano	50	13	7	1	313	normal
Laske	13	41	13	3	42	normal; log-normal
Okuonghae	16	9	2	1	92	normal
Oliveira	12	9	2	1	120	normal
Perelson	3	4	1	1	16	log10-normal
Rahman	9	7	1	1	23	normal
Salazar	6	75	3	4	18	normal
Schwen	30	11	4	19	286	log10-normal
Sneyd	15	6	1	9	135	normal
Weber	36	7	8	2	135	normal
Zhao	28	5	1	7	82	normal
				subset II		
Alkan	56	36	12	73	1733	normal
Bachmann	113	25	20	36	541	normal; log10-normal
Beer	72	4	2	19	27132	normal
Chen	155	500	3	4	120	normal
Fujita	19	9	3	6	144	normal
Isensee	46	25	3	123	687	normal
Lucarelli	84	33	65	16	1755	normal; log10-normal
Raimundez	136	22	79	170	627	normal
Zheng	46	15	15	1	60	normal

- Benchmarks were divided in two subsets:
  - **I-easier**
    - **Processors:** 12.
    - **Execution time:** 3 hours.
  - **II-harder**
    - **Processors:** 24.
    - **Execution time:** 9 hours.

Comparison methodology & infrastructure

- We have tested 37 solvers:
  - 3 based on **SaCeSS** configurations
  - 1 using the parallel multistart implemented in **parPE+Ipopt**
  - 33 solvers using the parallel **pyPESTO** multistart method with different solvers

- Each solver was executed 10 times (jobs) for each problem

- All solvers were run in a parallel setup, we used the same number of parallel processors in each job.
- In adition, we used the same computational infrastructure (**Finisterrae III**), and the same **MPI** implementation (**openmpi 4.1.1**) and compilers (**gcc 10.1.0**)

Summary of results: the best solution achieved.

Solutions 0-1% (or larger) from best

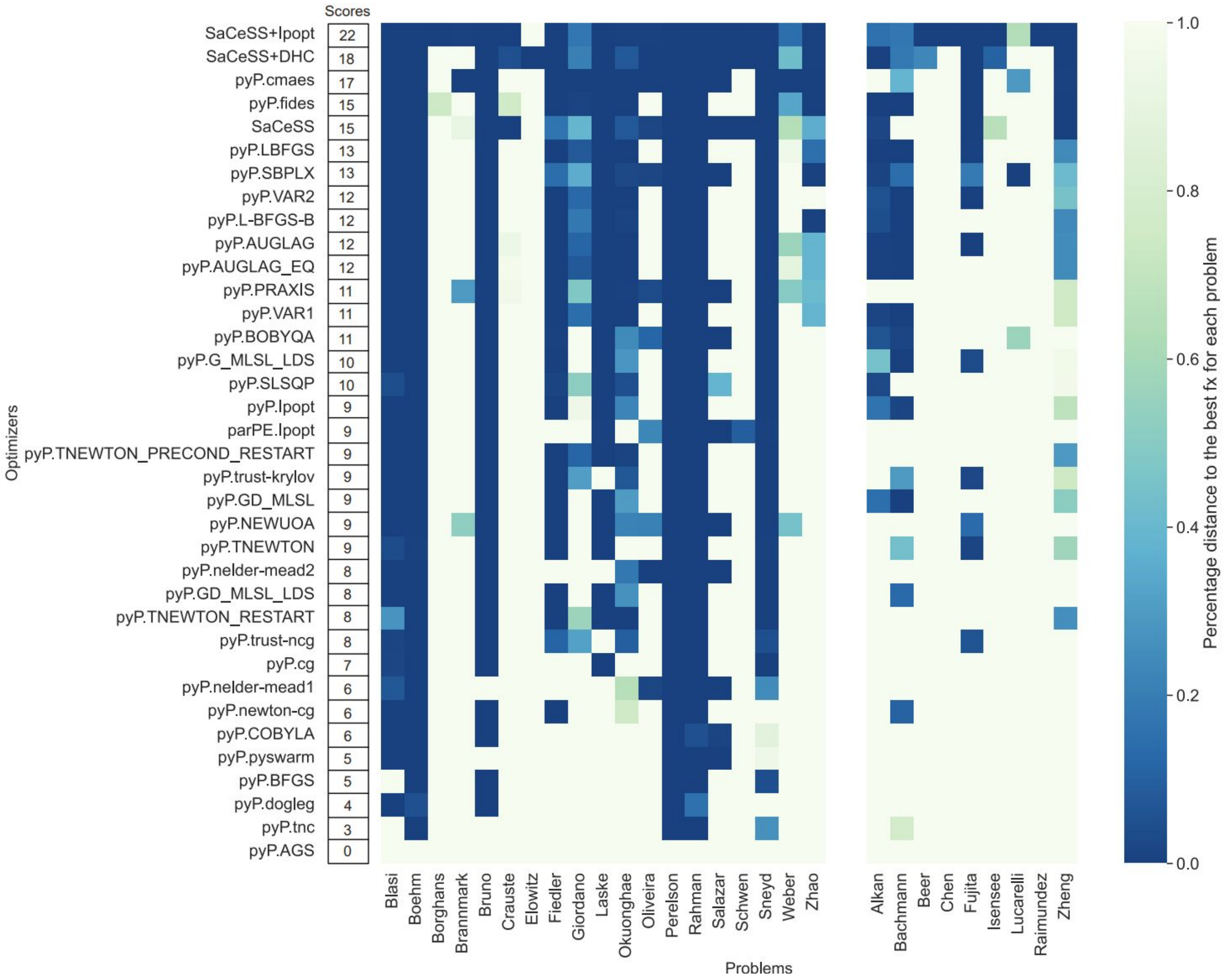
The heatmap with the percentage distance to the best fx for each solver and in each problem

Calculating distance of solver1 in problem P1:

distance_solver1 =

(best_cost_P1 - best_cost_solver1)

/ best_cost_P1 * 100



Conclusions

- Considering most problems solved (best or near-best solution achieved), **`SaCeSS+Ipopt`** is the winner

- **`SaCeSS+DHC`** and **`SaCeSS`** without local solver also have a good performance in terms of the best solution reached and robustness

- Multistart of **`fides`** and **`cmaes`** was competitive in some benchmarks, with a small dispersion in the results. An integration between **`SaCeSS+fides`** looks promising.

- Cooperative methods such as **`SaCeSS`** are a good alternative to solve complex problems in the **calibration of large-scale dynamic models in systems biology**

Acknowledgements

Can Path Pro

This work was supported by the European Union’s Horizon 2020 research and innovation program under grant agreement no. 686282, CanPathPro.

THE FRAMEWORK PROGRAMME FOR RESEARCH AND INNOVATION

HORIZON 2020

BIODYNAMICS

This research has received support from grant PID2020-117271RB-C22 (BIODYNAMICS) funded by MCIN/AEI/ 10.13039/501100011033

GOBIERNO DE ESPAÑA

MINISTERIO DE CIENCIA E INNOVACIÓN

AGENCIA ESTATAL DE INVESTIGACIÓN

CESGA

GALICIA SUPERCOMPUTING CENTER

CSIC

CONSEJO SUPERIOR DE INVESTIGACIONES CIENTÍFICAS

100 CENTENARIO

MBG

MISIÓN BIOLÓGICA DE GALICIA

UNIVERSITÄT BONN