Reliable detection of CVD: ABC with HPC

Ritabrata Dutta

Data Science Center, Institute of Computational Science
Università della Svizzera italiana, Switerland
(joint work with: Prof. Bastien Chopard, Prof. Antonietta Mira and Prof. Karim Zouaoui-Boudjeltia)

June 14, 2018, Swiss National Supercomputing Center, ETH-zürich

Cardio-vascular diseases (CVD)

Worldwide deaths because of
Cardiovascular Diseases
TRAVCURE
17.3 Million

Are you Walking a **risky** path?

Only way to know: Clinical tests

Existing clinical tests do not perform well:

Personalized clinical test:

- Incapable to consider inter-individual variability.

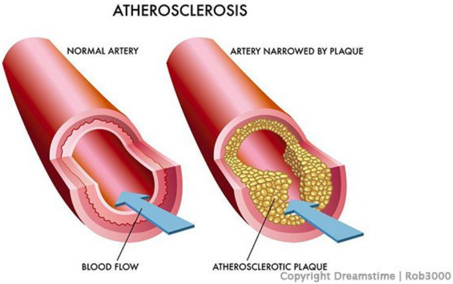
Mechanism of platelets:

- Do not consider different stages of platelet activation
- Do not consider the molecular dynamics involved in platelet interactions.

Why platelets are so important?

Platelet Adhesion and Aggregation

- **Platelets** are transported by blood
- They play a major role in many physiological porcesses: e.g. thrombus formation



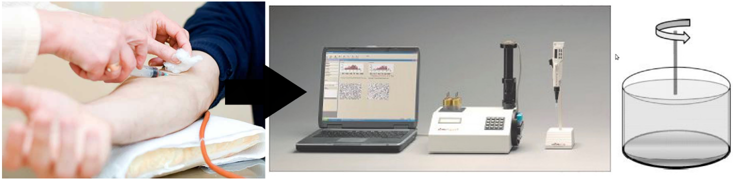
Copyright Dreamstime | Rob3000

- Their **adhesion and aggregation rate** on the vessel walls are important properties, whose values may indicate pathologies
- How can we determine these quantities?

Step 1: Numerical model [Chopard et al., (2017)]

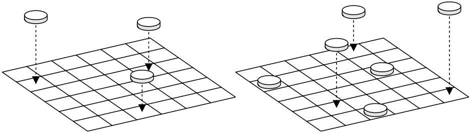
Numerical model of platelet deposition, adhesion and aggregation

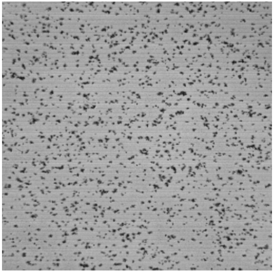
- Collect blood from a patient.
  Prof Karim Zouaoui-Boudjeltia, ULB and CHU-Charleroi, Belgium.



- Observe the pattern of platelet deposition, adhesion and aggregation in Impact-R machine
- Numerical model can replicate the pattern.
  Prof Bastien Chopard, U. Geneva, Switzerland. *R. Soc. open sci. 4:170219, 2017.*

Observed data: Platelet deposition

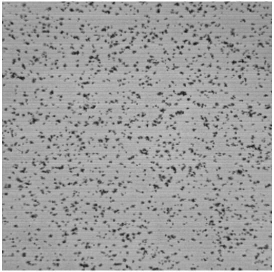




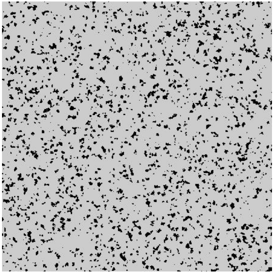
- Due to the diffusion produced by the shear flow, platelets reach the bottom of the device and deposit on the substrate at some rate that we would like to compute.

Numerical model of the adhesion-aggregation process

Experiment



Simulation



- pre-activated, non-activated platelets, **albumin**
- in suspension, or deposited (cluster)
- Experiments give the number of particles in suspension as a function of time ($t = 0$, $t = 20$, $t = 60$, $t = 120$ and $t = 300$ seconds).
- Experiments also give the number of clusters and their size at the same time points.

Step 2: Parameters of the numerical model

- Numerical model depends on 5 different parameters: aggregation rate, adhesion rate, aggregation rate, deposition rate of albumin, attenuation factor
- Your blood has an **unique value** of these five parameters.
- Simulated data matches observed data (from your blood) only when parameter values are correct.

How can we **estimate** values of these parameters?

Suggested Clinical Test

Claim: The values of these adhesion and aggregation rates are precisely the information needed to assess various possible pathological situations and quantifying their severity regarding CVD.

Meaning we can tell from values of these parameters, whether you will have CVD or not!

Question: How do we know what are the values for those parameters given your blood?

Answer: Approximate Bayesian computation (ABCpy) + High performance computing (CSCS)

Numerical model: A generative model

$$\mathcal{M}[\boldsymbol{\theta} = \boldsymbol{\theta}^*] \rightarrow \{(\mathcal{S}_{agg-clust}(t), \mathbb{N}_{agg-clust}(t), \mathbb{N}_{plat}(t), \mathbb{N}_{act-plat}(t))\}_{t=0}^{T}$$

- Forward simulate from $\mathcal{M}$, given value of parameters $\boldsymbol{\theta} = \boldsymbol{\theta}^*$, $\mathbb{N}_{plat}(0)$ and $\mathbb{N}_{act-plat}(0)$

$\boldsymbol{\theta}$

p_{Ag} : aggregation rate

p_{Ad} : adhesion rate

p_T : aggregation rate

p_F : deposition rate of albumin

a_T : attenuation factor

$\boldsymbol{x_{obs}}$

$\mathcal{S}_{agg-clust}(t)$: avg. size of the agg. clusters

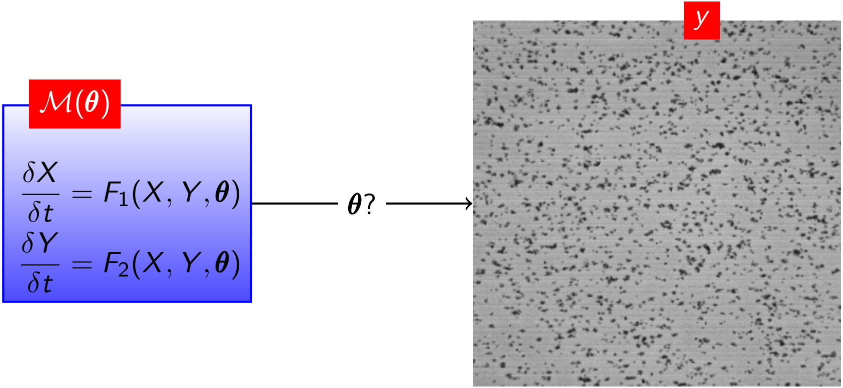
$\mathbb{N}_{agg-clust}(t)$: no. of the agg. clusters

$\mathbb{N}_{plat}(t)$: no. of pre-activated plat.

$\mathbb{N}_{act-plat}(t)$: no. of non-activated plat.

per mm^2, still in suspension at time t.

Our interest: Inverse problem



$\mathcal{M}(\boldsymbol{\theta})$

$$\frac{\delta X}{\delta t} = F_1(X, Y, \boldsymbol{\theta})$$

$$\frac{\delta Y}{\delta t} = F_2(X, Y, \boldsymbol{\theta})$$

$\boldsymbol{\theta}$?

y

- Question: inference on the parameters defining the DE system

Stochastic model

- Stochasticity in forward simulation:

$$\begin{tikzcd} & \boldsymbol{x_{obs}}^1 \text{ w.p. } p_1 \\ \mathcal{M} : \boldsymbol{\theta} \arrow[ur] \arrow[dr] & \\ & \boldsymbol{x_{obs}}^2 \text{ w.p. } p_2 \end{tikzcd}$$

- Stochasticity in inverse problem:

$$\begin{tikzcd} \mathcal{M} : \boldsymbol{\theta}_1 \arrow[dr, "\text{w.p. } q_1'"] & \\ & \boldsymbol{x_{obs}} \\ \mathcal{M} : \boldsymbol{\theta}_2 \arrow[ur, "\text{w.p. } q_2'"'] & \end{tikzcd}$$

- No unique solution of $\boldsymbol{\theta}$, due to stochastic nature.
- So we need to quantify the uncertainty in $\boldsymbol{\theta}$ give $\boldsymbol{x_{obs}}$

Bayes Theorem

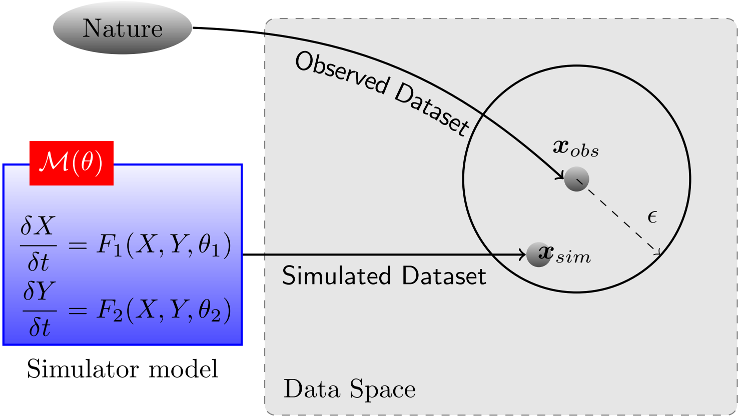
- Bayesian approach to uncertainty quantification

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{x})}$$

  - $p(\boldsymbol{\theta})$: (prior) A function of $\boldsymbol{\theta}$, assigning prior knowledge about $\boldsymbol{\theta}$
  - $p(\boldsymbol{x}|\boldsymbol{\theta})$: (likelihood) A function of $\boldsymbol{\theta}$, assigning likelihood of $\boldsymbol{\theta}$ given $\boldsymbol{x}$
  - $p(\boldsymbol{\theta}|\boldsymbol{x})$: (posterior) Probability of $\boldsymbol{\theta}$, given $\boldsymbol{x}$
  - $p(\boldsymbol{x})$: (evidence) $\int p(\boldsymbol{\theta}|\boldsymbol{x})p(\boldsymbol{\theta})d\boldsymbol{\theta}$
- The stochastic nature of inverse problem is captured by likelihood function $p(\boldsymbol{x}|\boldsymbol{\theta})$.

Likelihood (LHD: $p(\boldsymbol{x}|\boldsymbol{\theta})$) is computationally expensive or infeasible, ruling out both likelihood-based and posterior-based inference

ABC



Within the black ball centered around $\boldsymbol{x_{obs}}$: $\Delta_{\boldsymbol{\theta}} < \epsilon$, for some $\epsilon > 0$

Approximate Bayesian Computation (ABC)

ABC avoids direct evaluation of the LHD and approximates it by generating pseudo-data (synthetic observations) by forward simulation from the model

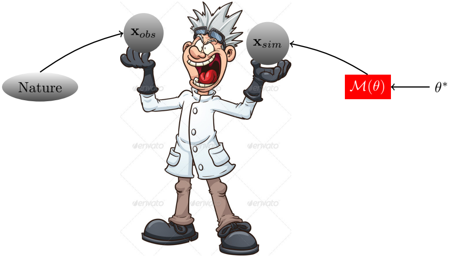
- Basic idea: Identify the values of $\boldsymbol{\theta}$ which produce simulated data, $\boldsymbol{x}_{\mathbf{sim}}$, resembling the observed data, $\boldsymbol{x}_{\boldsymbol{obs}}$
- Simulated data resemble the observed data if some discrepancy measure $\Delta_{\boldsymbol{\theta}}(\boldsymbol{x}_{\mathbf{sim}}, \boldsymbol{x}_{\boldsymbol{obs}})$ is small

ABC: Boosted by HPC

Accept θ^*: if $\Delta_{\boldsymbol{\theta}}(\mathbf{x}_{obs}, \mathbf{x}_{sim}) < \epsilon$

Reject θ^*: if $\Delta_{\boldsymbol{\theta}}(\mathbf{x}_{obs}, \mathbf{x}_{sim}) > \epsilon$

$\epsilon \downarrow 0$



Nature

$\mathbf{x}_{obs}$

$\mathbf{x}_{sim}$

$\mathcal{M}(\theta)$

θ^*

ABCpy: Efficient ABC algorithms with HPC (PASC'2017)

Sketch of an ABC algorithm

i. (re-)sample a set of parameters $\boldsymbol{\theta}$ either from the prior or from an already existing set of parameters

→ 1000 samples/parameter values

ii. **Update** each sample: perturb it using the perturbation kernel,

→ simulate the model and generate data,

compute the distance between generated and observed data, and either accept it if the distance $< \epsilon$, or repeat the whole second step

iii. For each sample calculate its weight

iv. normalize the weights, calculate a co-variance matrix and a quantile

Repeat (i→iv): by decreasing ϵ

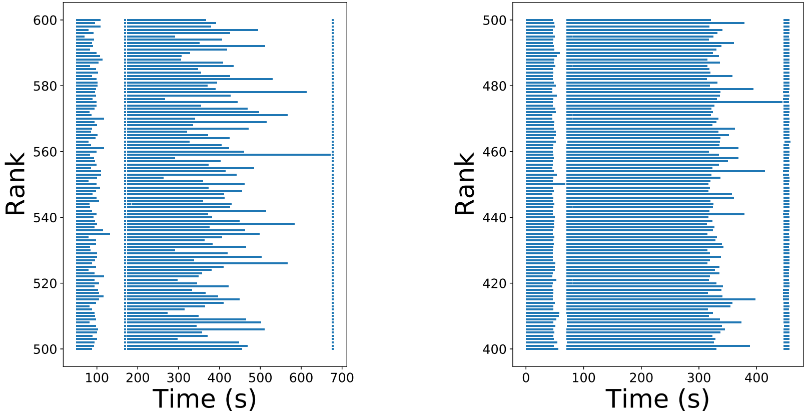
ABC with HPC: ABCpy

- Each simulation of data task is costly (for our model 10 minutes)
- But generally ABC algorithms are parallelizable
- Next → Development of ABCpy

ABC with HPC: ABCpy

- ABCPy: A python suite of ABC, user friendly and modular [Dutta et. al. 2017a]
- Super-computers: Developed in collaboration with Swiss Super Computing Center (CSCS)
- Usability: In collaboration with CSCS, we offer to infer model/parameter of your problem using the most powerful super computer of Europe (CRAY)
- Map-Reduce: For parallelization we use *Map-reduce* scheme of Spark, MPI and dynamic allocation MPI (implemented by us to mitigate imbalance in ABC)

ABCpy: Solving imbalance in ABC

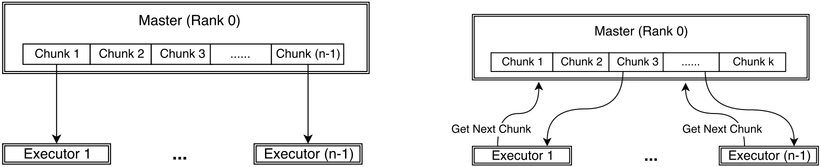


(a) MPI(straight-forward)

(b) MPI(dynamic-allocation)

Figure: Imbalance of ABC algorithms using MPI(straight-forward) and MPI(dynamic-allocation) backend

ABCpy: Solving imbalance in ABC



(a) MPI(straight-forward)

(b) MPI(dynamic-allocation)

Figure: MPI(straight-forward) and MPI(dynamic-allocation) backend [Dutta et. al. 2017b]

ABCpy: A brief

Implemented ABC algorithms

- For inference:
  1. Rejection ABC [Tavaré et. al. 1997]
  2. Population Monte Carlo ABC PMC-ABC [Beaumont 2010]
  3. Sequential Monte Carlo ABC SMC-ABC [Del Moral et al 2012]
  4. Replenishment SMC ABC RSMC-ABC [Drovandi et al 2011]
  5. Adaptive Population Monte Carlo ABC APMC-ABC [Lenormand et al 2013]
  6. ABC with subset simulation ABCsubsim [Chiachio et al 2014]
  7. Simulated Annealing ABC SABC [Albert et al 2015]
  8. (Coming soon) Surrogate modeling based ABC

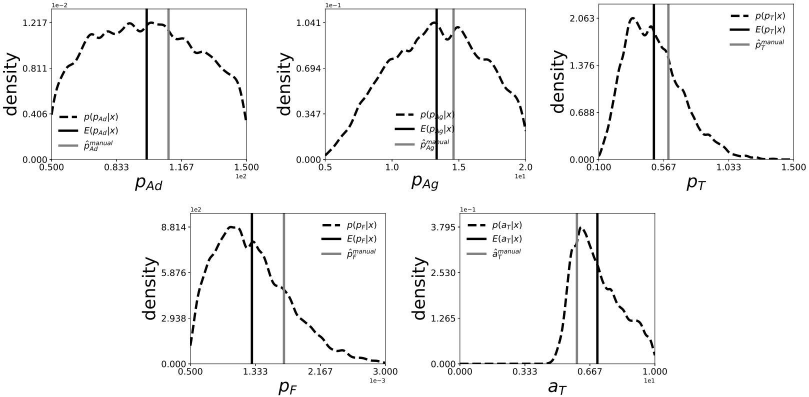
ABCpy: A brief

- For summary selection: Semi-automatic summary selection [Fearnhead and Prangle, 2012]
- Specialized distances: Classifier-ABC [Gutmann et. al. 2017], with automatic summary selection
- Model selection: Random forest ensemble model selection [Pudlo et. al., 2015]
- Additional: Population Monte Carlo to perform pseudo-marginal approach using approximate likelihoods:
  1. Synthetic Likelihood [Woods 2010]
  2. Penalised Logistic Regression [Dutta et. al. 2017c]

ABC with HPC

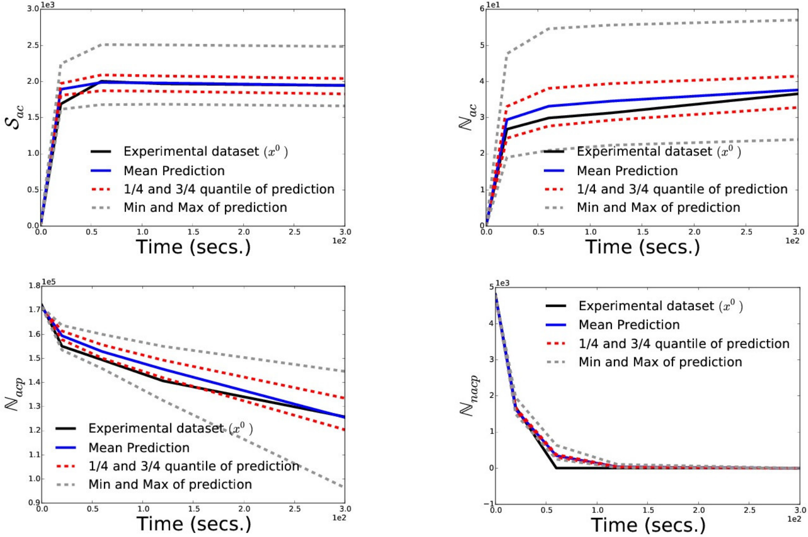
- Lets use ABC to solve inverse problem in our application: ABCpy+HPC+Problem = ‘Exciting results’!

Collective dataset of 7 patients



- Marginal posterior distribution (black-dashed)
- Bayes Estimate (black-solid) of $\boldsymbol{\theta}$
- Manual estimate (gray-solid) of $\boldsymbol{\theta}$

Posterior Prediction: Model prediction uncertainty



What did we achieve?

Your blood

→ Numerical model + Impact-R + ABCpy + HPC
→ value of these parameters

‘Reliable’ detection of CVD

(At this moment we are performing clinical trial in cooperation with CHU Charleroi.)

→ **Personalized medicine**

What to be further developed

Cost of diagnosis: Depends on,

1. Cost of simulation of a model (10 minutes)
2. Cost of HPC infrastructure to run ABC
3. Around 100 euros for each patient.

Can we make cheaper?

1. Use GPU acceleration for both numerical model and ABCpy
2. Couple GPU acceleration with present parallelization of ABCpy - Nested parallelization?
3. Develop efficient ABC algorithms
4. Further develop numerical model

If you are interested ...

- ABCpy can be run on Daint, provided by CSCS
- ABCpy can be downloaded from Github
- For a quick look into ABCpy documentation
- Some simulation models, calibrated using ABCpy

Thanks

- Paper can be found: arxiv.org/abs/1710.01054

- Funding: Swiss National Science Foundation Grant No. 105218_163196
- Partial funding: Horizon 2020 research and innovation programme for the CompBioMed project under grant agreement 675451
- HPC Infrastructure: CADMOS and Swiss Super Computing Center
- Experimental data: CHU Charleroi for supporting the experimental work used in this study