Approximate Bayesian Computation to calibrate Force-Fields

Antonietta Mira

Director, Data Science Lab, Institute of Computational Science
Università della Svizzera italiana, Switzerland
Professor of Statistics, Università dell'Insubria

Joint with:
R. Dutta and **ZF. Brotzakis**

Based on a paper in
Journal of Chemical Physics, 2018

Helsinki, ML4MS 2019 event

Big picture of statistical inference

GIVEN:

- Data $= x = (x_1, \ldots, x_n)$
- Model which describes data, $p_{x|\theta}(x|\theta)$
  indexed by parameters $= \theta = (\theta_1, \ldots, \theta_d)$
- Prior probability density for θ, p_θ

WANTED:

- Some probabilistic statement about θ and model
  - point estimation
  - confidence/credible intervals
  - hypothesis testing
  - prediction
  - model selection

Big picture of statistical inference

GIVEN:

- Data $= x = (x_1, \ldots, x_n)$
- Model which describes data, $p_{x|\theta}(x|\theta)$
  indexed by Parameters $= \theta = (\theta_1, \ldots, \theta_d)$
- Prior probability density function for θ, p_θ

WANTED:

- Some probabilistic statement about θ and model
  - point estimation
  - confidence/credible intervals
  - hypothesis testing
  - prediction
  - model selection

Two types of models

- Statistical model

$$p_{x|\theta}(x_i|\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x_i - \mu)^2\right), \quad \theta = (\mu, \sigma)$$

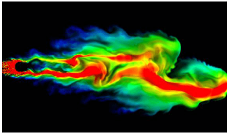
- Generative model
  $\rightarrow$ given $\theta = (\mu, \sigma)$
  $\rightarrow z_i \sim \mathcal{N}(0, 1)$
  $\rightarrow x_i = \mu + \sigma z_i$
  $\rightarrow x_i \sim p_{x|\theta}(x_i|\theta)$
- In some settings easier to specify a generative model
- Sometimes there is no 1:1 correspondence bwn statistical and generative model

Examples of generative models

$$\text{Model } \mathcal{M}(\theta)\text{:} = p_{x|\theta}(x|\theta) \rightarrow \text{Simulate Data: } x_{sim}$$

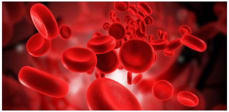
- Fluid dynamics:

  Angelos Cronis et. al. 2012



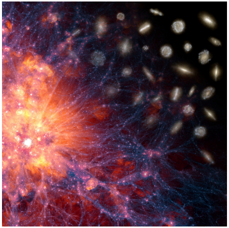
- Bio simulation:

  Dutta, Bastien, Mira et. al. 2018



- Simulation of galaxy:

  Gauss center for supercomputing, 2019



Other examples

- Evolutionary biology:
  Simulating species evolution
- Ecology:
  Simulating species migration over time
- Neuroscience:
  Simulating neural circuits
- Health science:
  Simulating the spread of an infectious disease
- Meteorology :
  Simulating weather prediction

Big picture of statistical inference

GIVEN:

- Data $= x = (x_1, \ldots, x_n)$
- Model which describes data, $p_{x|\theta}(x|\theta)$
  indexed by Parameters $= \theta = (\theta_1, \ldots, \theta_d)$
- Prior probability density function for θ, p_θ

WANTED:

- Some probabilistic statement about θ and model
  - point estimation
  - confidence/credible intervals
  - hypothesis testing
  - prediction
  - model selection

Likelihood-based statistical inference

- Likelihood function:

$$L(\theta) \propto p_{x|\theta}(x|\theta)$$

- Plays a central role in statistical inference
  - Maximum likelihood estimation:

$$\widehat{\theta}_{\text{MLE}} = \text{argmax}_{\theta} L(\theta)$$

  - Bayesian inference:

$$p_{\theta|x}(\theta|x) \propto L(\theta) p_{\theta}(\theta)$$

- *Likelihood function not available for generative models*

We do inference for LHD free generative models

Successful collaborations

- Network Science
  → To detect source and spreading of fake news / epidemics
- Health
  → To personalize clinical tests of cardiovascular diseases
- Dynamic Queuing Network
  → To better manage passengers in airports
- Physics → To calibrate Force-Fields (+ UQ) to reproduce properties measured by simulations or experiments

On-going collaborations

- Hydrology
- Modeling of solar dynamo
- ....

Details of the setting

- We consider MD simulations, which sample the phase space by integrating the deterministic Newtons equations of motion giving access to dynamic and thermodynamic properties for which LHD is not available analytically (unlike forces and energies)
- Uncertainty: Model, Parameters, Computational (finite number of molecules and simulation time), Measurement
- The accuracy of the underlying molecular mechanics Force-Field used to solve the equations of motion defines the approximation in the phase space exploration
- Lennard-Jones Force-Field parameters of helium and (rigid non-polarizable) TIP4P Force-Field of water
- Simulated (LAMMPS, GROMACS) and experimental data collected using Neutron and X-ray diffraction

Final results:

- strong correlation pattern between the Force-Field parameters
- posterior distribution allows uncertainty quantification
- calibrate + predict + select Force-Field formalisms (TIP3P, TIP4P, TIP5P)

Approximate Bayesian Computation (ABC) references

- ABC in population genetics, MA Beaumont, W Zhang, DJ Balding - Genetics, 2002
- Comparative evaluation of a new effective population size estimator based on approximate Bayesian computation DA Tallmon, G Luikart, MA Beaumont - Genetics, 2004
- Inferring population history with DIY ABC: a user-friendly approach to ABC, JM Cornuet, F Santos, MA Beaumont, CP Robert, JM Marin, . . . - Bioinformatics, 2008
- COMPUTER PROGRAMS: onesamp: a program to estimate effective population size using ABC, DA Tallmon, A Koyuk, G Luikart, MA Beaumont - Molecular Ecology Resources, 2008
- Adaptive ABC, MA Beaumont, JM Cornuet, JM Marin, CP Robert - Biometrika, 2009
- Approximate Bayesian computation without summary statistics: the case of admixture, VC Sousa, M Fritz, MA Beaumont, L Chikhi - Genetics, 2009
- Review: Marin, Statistics and Computing, 2012

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, $\boldsymbol{x_{sim}}$, resembling the observed data, $\boldsymbol{x_{obs}}$

- Simulated data resemble the observed data if some discrepancy measure $\Delta_{\boldsymbol{\theta}}(\boldsymbol{x_{sim}}, \boldsymbol{x_{obs}})$ is small

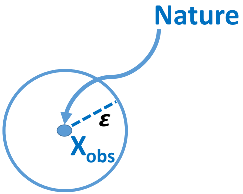
Approximate Bayesian Computation (ABC)

Starting point is Bayes' theorem:

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

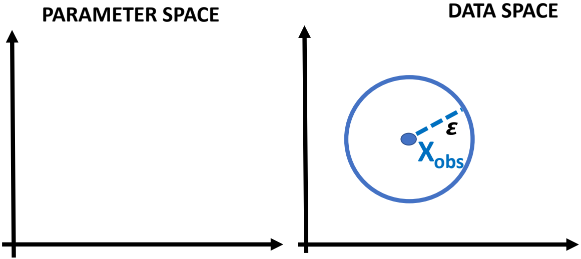
- $x = x_{obs}$
- $p(\theta|x) =$ posterior
- $p(x|\theta) =$ likelihood
- $p(\theta) =$ prior
- $p(x) =$ evidence

Rejection ABC scheme

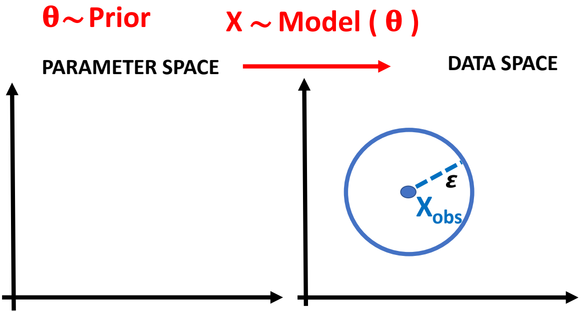


Rejection ABC scheme

$\theta \sim$ Prior



Rejection ABC scheme



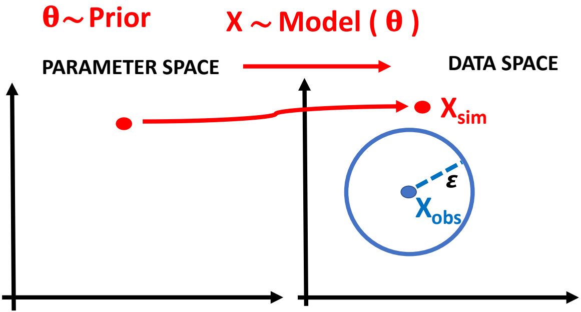
$\theta \sim$ Prior

$X \sim$ Model (θ)

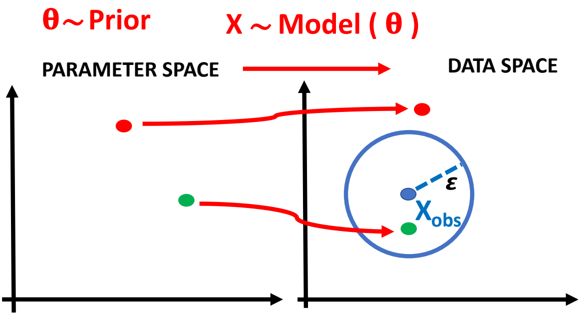
PARAMETER SPACE

DATA SPACE

Rejection ABC scheme



Rejection ABC scheme



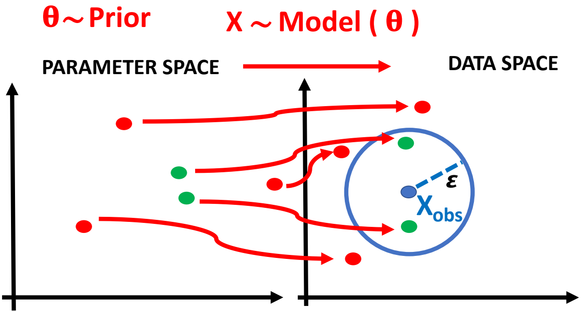
$\theta \sim$ Prior

$X \sim$ Model (θ)

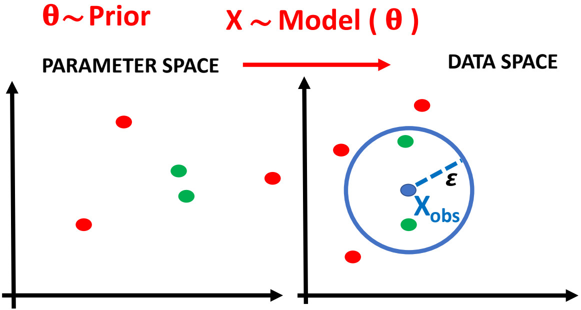
PARAMETER SPACE

DATA SPACE

Rejection ABC scheme



Rejection ABC scheme



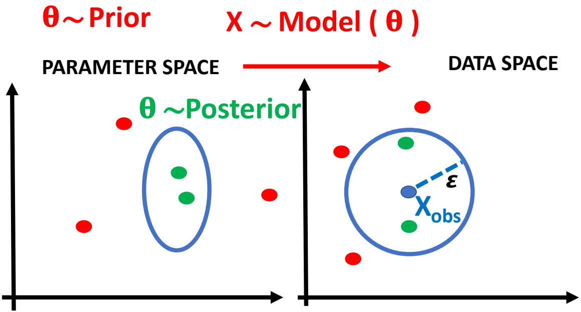
$\theta \sim$ Prior

$X \sim$ Model (θ)

PARAMETER SPACE

DATA SPACE

Rejection ABC scheme



Rejection ABC

- ABC rejection sampler is the simplest form of ABC

ABC rejection sampler

- Sample parameter θ from the prior $p(\theta)$
- Simulate dataset x_{sim} under the given model specified by θ: $x_{sim} \sim p(\cdot|\theta)$
- Accept θ if $\Delta_{\theta}(x_{sim}, x_{obs}) \leq \epsilon$

- Distance $\Delta(x_{sim}, x_{obs})$ measures the discrepancy between the simulated data x_{sim} and the observed data y

- The accepted θ are approximately distributed according to the desired posterior and, crucially, obtained without the need of explicitly evaluating the LHD

Rejection ABC

- It may be unfeasible to compute the distance $\Delta(x_{sim}, x_{obs})$ for high-dimensional data
- Lower dimensional summary statistic $S(x_{obs})$ to capture the relevant information in x
- Comparison is done between $S(x_{sim})$ and $S(x_{obs})$: accept θ if $\Delta(S(x_{sim}), S(x_{obs})) \leq \epsilon$
- If S is sufficient wrt θ, then it contains all information in y about θ (by definition), and using $S(x_{obs})$ in place of the full dataset does not introduce any error
- For most models it may be impossible to find sufficient statistics S, in which case application relevant summary statistics need to be used
- Use of non-sufficient summary statistics introduces a further level of approximation

A more advanced ABC: Adaptive Population Monte Carlo ABC - APMCABC

Step 1. (re-)sample a set of parameters $\boldsymbol{\theta}$ either from the prior or from an already existing set of parameters

$\rightarrow$ 5000 parameter values

Step 2. **Update** each parameter using the perturbation kernel

$\rightarrow$ given perturbed parameter
simulate from model and generate pseudo-data

compute the distance between simulated and observed data,
and either accept parameter if the distance $< \epsilon$
or repeat Step 2.

Step 3. For each accepted parameter calculate a weight

Step 4. Normalize the weights
Calculate covariance matrix for next perturbation kernel

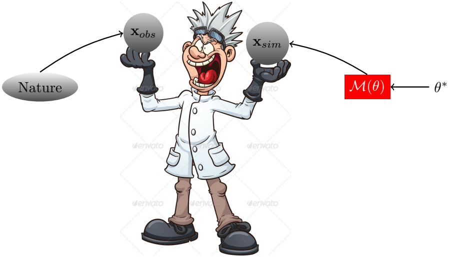
Repeat (Step 1→Step 4) while decreasing ϵ

ABC boosted by HPC: ABCpy

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

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

$\epsilon \downarrow 0$



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

ABC boosted by HPC: ABCpy

- Each fwd data simulation is costly (from minutes to hours)
- ABC algorithms are parallelizable
- Development of ABCpy

ABC boosted by 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)
- Reproducibility
- 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)

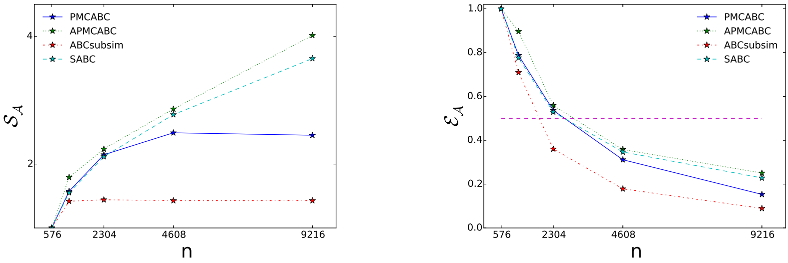
ABCpy: A brief

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]

- So which one should we use?

Comparison of algorithms: HPC perspective



(a) Speedup

(b) Efficiency

The best algorithm in terms of speedup + efficiency is APMCABC

ABCpy: A brief

- For summary selection: Semi-automatic summary selection [Fearnhead and Prangle, 2012]

- Specialized distances: Classifier-ABC [Gutmann et. al. 2018], 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]

Calibration of Force-Field Helium (Kulakova et. al. 2016)

- The LJ potential is given by

$$V_{LJ}(\sigma_{LJ},\epsilon_{LJ}) = \sum_i \sum_j 4\epsilon_{LJ}\left(\left(\frac{\sigma_{LJ}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{LJ}}{r_{ij}}\right)^{6}\right) \quad (1)$$

  $\epsilon_{LJ}(zJ) =$depth of the potential well
  $\sigma_{LJ}(nm) =$finite distance at which inter-particle potential $= 0$
  $r_{ij}(nm) =$distance between the i and j particles

- Non-bonded Force-Field parameters: $\theta = (\sigma_{LJ},\epsilon_{LJ})$
- Generative model (fwd simulated with LAMMPS):

$$\mathcal{M}_{LJ}[\theta = \theta^*] \rightarrow \{(\text{coordinates}(t))\,,\ t = 0,\ldots,t_{end}\}$$

- **Summary statistics**:
  $\mathcal{F}_{LJ} : \boldsymbol{x} \rightarrow f_B(t) = \langle exp\{-H(t)/(k_B T)\}\rangle$
  $H(t) =$ enthalpy contribution of a helium atom at time t
  $T =$ temperature of the system
  $k_B =$ Boltzmann constant
  $\langle\ \rangle =$ ensemble average over all atoms in the system at time t

Calibration of Force-Field Helium (Kulakova et. al. 2016)

- **Discrepancy measure**:

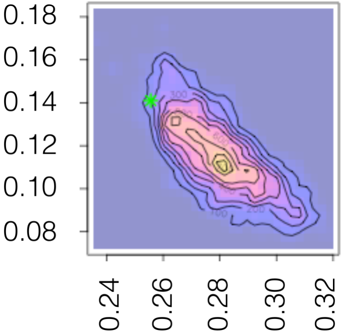
$$\begin{aligned} d_{LJ}(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) &:= d_{LJ}\left(\mathcal{F}_{LJ}(\boldsymbol{x}^{(1)}), \mathcal{F}_{LJ}(\boldsymbol{x}^{(2)})\right) \\ &:= d_{LJ}\left(f_B^{(1)}, f_B^{(2)}\right) \\ &= KL\left(f_B^{(1)}, f_B^{(2)}\right) \\ &= \int \chi^{(1)}(z) \log \frac{\chi^{(1)}(z)}{\chi^{(2)}(z)} dz \end{aligned}$$

  where $\chi^{(1)}(z)$ and $\chi^{(2)}(z)$ are, respectively, the probability density functions of $f_B^{(1)}$ and $f_B^{(2)}$

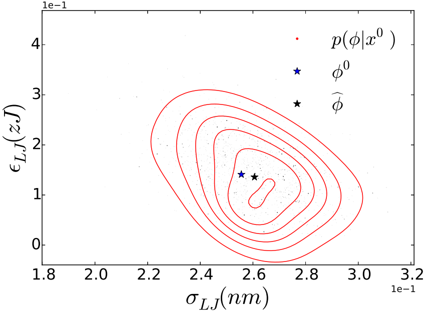
- **Priors**: independent continuous uniform
  $\sigma_{LJ} \sim U[0.1(nm), 0.8(nm)]$
  $\epsilon_{LJ} \sim U[0.01(zJ), 1.0(zJ)]$
- **Perturbation kernel**: truncated two-dimensional Gaussian centered at current value with covariance learned from previous particles

ABCsubsim vs APMCABC: Posterior distribution

- No assumption of Gaussianity on likelihood functions
- After running both algorithms for 6 steps and 5000 particles



(c) ABCsubsim, Kulakova 2016



(d) APMCABC, Dutta et. al. 2018

Comparison: APMCABC vs ABCsubsim to calibrate Lennard-Jones FF of Helium, after $N_{\text{step}} = 6$

Euclidean distance bwn Bayes estimate and true parameter value used to simulate the dataset $d_E(\widehat{\theta}, \theta^0)$

Final ABC threshold value, δ_{final}

Algorithm	$d_E(\widehat{\theta}, \theta^0)$	δ_{final}
APMCABC	0.00744	0.0138
ABCsubsim	0.03365	0.67

Calibration of TIP4P Force-Field Water

- Water structure and dynamics regulates biological and physicochemical processes
- TIP4P = rigid nonpolarizable, 4 interaction site, FF with all bonds and angles constrained using the LINCS algorithm
- Potential energy = LJ + electrostatic interactions:

$$
\begin{aligned}
U_{noncov}(\sigma_{TP}, \epsilon_{TP}) &= \sum_i \sum_j 4\epsilon_{TP} \left[\left(\frac{\sigma_{TP}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{TP}}{r_{ij}} \right)^{6} \right] \\
&+ \sum_i \sum_j \sum_\alpha \sum_\beta \frac{q_{i\alpha} q_{j\beta}}{r_{ij}}
\end{aligned}
$$

α and β = indices of the partial charges q of each molecule
Non-bonded FF parameters: $\theta = (\sigma_{LJ}, \epsilon_{LJ})$ repulsion and attraction of the Van der Waals forces

Calibration of TIP4P Force-Field Water

- Generative model (fwd simulated with GROMACS):

$$\mathcal{M}_{TP}[\theta = \theta^*] \to \{(\text{coordinates}(t))\,,\ t = 0, \ldots, t_{end}\}$$

- After compiling the TIP4P FF with $\boldsymbol{\theta^*}$, we perform an energy minimization (steepest descend), followed by an NPT simulation (LINCS)

Calibration of TIP4P Force-Field Water

- LHD assumed Gaussian in existing works and fit is on forces and energy

- ABC → No assumption of Gaussianity on LHD and fit is on termodynamical properties

- From experimental studies it is not possible to track the time dependent position of water molecules, but we can learn their properties, e.g. different radial distribution functions, using different diffraction techniques

- Radial distribution functions and self-diffusion coefficient (Neutron and X-ray diffraction) considered as data

Calibration of TIP4P: summary statistics

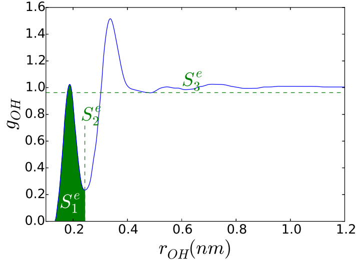
Summary statistics: characteristic quantities of the structure and dynamics of liquids

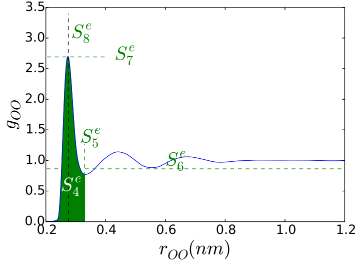
$$\mathcal{F}_{TP} : \boldsymbol{x} \to (S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9)$$

First compute:

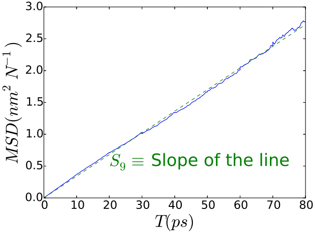
- the distribution functions (g_{OH} and g_{OO}) of the radial for the $O - H$ (r_{OH}) and $O - O$ (r_{OO}) atoms
- the (M)ean (S)quare (D)isplacement (MSD) from the coordinates of the dynamical system

Calibration of TIP4P: distribution functions of radials for $O - H$ and $O - O$





Calibration of TIP4P: (M)ean (S)quare (D)isplacement



Calibration of TIP4P: summary statistics on distribution functions of $O - H$ radial

Then compute the summary statistics as follows:

- S_1: Estimate of the number of hydrogen bonds per water molecule - The area under the curve r_{OH} vs g_{OH} until the first minimum
- S_2: Estimate of the donor acceptor hydrogen bond distance - Value of r_{OH} (nm) at the first minimum of the radial distribution function g_{OH}
- S_3: Mean of g_{OH}

Calibration of TIP4P: summary statistics summary statistics on distribution functions of $O - O$ radial

- S_4: Estimate of number of water molecules in the first hydration shell - The area under the curve r_{OO} vs g_{OO} until the first minimum
- S_5: Estimate of the max distance of the first hydration shell - Value of r_{OO} (nm) at the first min of the radial distribution function g_{OO}
- S_6: Mean of g_{OO}
- S_7: The height of g_{OO} at the first max of g_{OO}
- S_8: Value of r_{OO} (nm) at the first max of the radial distribution g_{OO}
- S_9: Slope of the line, fitted to MSD, which is an estimate of 6 $\times$ self-diffusion coefficient

TIP4P: discrepancy, prior, perturnation kernel

Discrepancy measure:

$$
\begin{aligned}
d_{TP}(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}) &:= d_{TP}\left(\mathcal{F}_{TP}(\boldsymbol{x}^{(1)}), \mathcal{F}_{TP}(\boldsymbol{x}^{(2)})\right) \\
&= \frac{1}{9}\sum_{i=1}^{9} |S_i^{(1)} - S_i^{(2)}|
\end{aligned}
$$

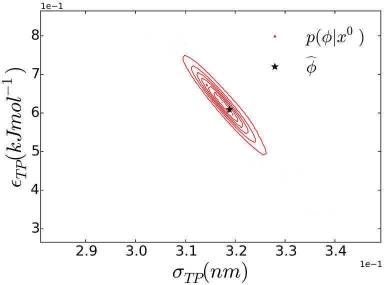
Priors:

Independent continuous uniform
$\sigma_{TP} \sim U[0.281(nm), 0.53(nm)]$
$\epsilon_{TP} \sim U[0.2(kJmol^{-1}), 0.9(kJmol^{-1})]$

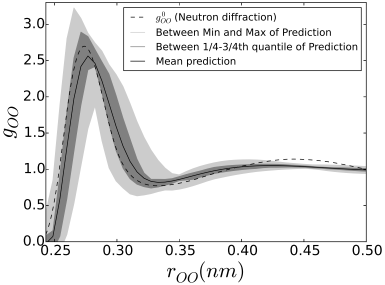
Outside this range the TIP4P model of water in GROMACS is extremely chaotic and simulated data set can not be obtained in reasonable time

Perturbation kernel: as before

Calibration of TIP4P Force-Field Water with APMC-ABC



(e) Posterior distribution



(f) Posterior prediction

Posterior distribution and prediction for the experimentally obtained radial distribution function of $O - O$

The experimental dataset is mostly within the prediction band

Validation of TIP4P Force-Field Water with APMC-ABC

We compare values of a set of properties not used for calibration

- Heat capacity ($Cp\ calmol^{-1}K^{-1}$)
- Density ($\rho\ gcm^{-3}$) of water at $298K$ and ice at $250K$
- Isothermal compressibility ($\kappa_T\ 10^{-6}/bar$)
- Dielectric constant (ξ) of water at $298K$

	Prop.	Expt.	TIP4P	Neutron diff.	X-ray diff.
Ice ($250K$)	Cp	8.3	14.7	12.47	20.02
	ρ	0.92	0.937	0.913	1
Water ($298K$)	Cp	18	20	20.1	18.3
	ρ	0.997	0.988	0.958	0.854
	κ_T	45.3	59	57.5	79.1
	ξ	78.5	50	47	43

Neutron closer than X-ray diffraction (does not have radial distribution function of $O - H$)

Conclusions

- **ABC**: very powerful methodology for sound statistical inference in mechanistic network models + processes

- **ABCpy**: python framework for ABC
  - Download from Github
  - For a quick start look at ABCpy documentation
  - Some simulation models, calibrated using ABCpy
  - Can be run on Piz Daint HPC, provided by CSCS

- **References**:
  - *Proc. Platform for Advanced Scientific Computing*, 2017
  - *Journal of Chemical Physics*, 2018
  - *Proc. Royal Society. A*, 2018
  - *Frontiers in Physiology*, 2018

Thanks

- 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 CSCS
- M. Parrinello and A. Laio (choice of FF and summary statistics)