The Journal of Open Source Software

# pyABC: Efficient and robust easy-to-use approximate Bayesian computation

**Yannik Schälte**[1,2,3], **Emmanuel Klinger**[2,3,4], **Emad Alamoudi**[1], and **Jan Hasenauer**[1,2,3]

**1** Faculty of Mathematics and Natural Sciences, University of Bonn, Bonn, Germany **2** Institute of Computational Biology, Helmholtz Center Munich, Neuherberg, Germany **3** Center for Mathematics, Technical University Munich, Garching, Germany **4** Department of Connectomics, Max Planck Institute for Brain Research, Frankfurt, Germany

**DOI:** 10.21105/joss.04304

**Software**

- Review
- Repository
- Archive

**Editor:** Øystein Sørensen

**Reviewers:**

- @mattpitkin
- @hpesonen
- @blakeaw

**Submitted:** 26 March 2022
**Published:** 25 June 2022

## Summary

The Python package pyABC provides a framework for approximate Bayesian computation (ABC), a likelihood-free parameter inference method popular in many research areas. At its core, it implements a sequential Monte-Carlo (SMC) scheme, with various algorithms to adapt to the problem structure and automatically tune hyperparameters. To scale to computationally expensive problems, it provides efficient parallelization strategies for multi-core and distributed systems. The package is highly modular and designed to be easily usable. In this major update to pyABC, we implement several advanced algorithms that facilitate efficient and robust inference on a wide range of data and model types. In particular, we implement algorithms to accurately account for measurement noise, to adaptively scale-normalize distance metrics, to robustly handle data outliers, to elucidate informative data points via regression models, to circumvent summary statistics via optimal transport based distances, and to avoid local optima in acceptance threshold sequences by predicting acceptance rate curves. Further, we provide, besides previously existing support of Python and R, interfaces in particular to the Julia language, the COPASI simulator, and the PEtab standard.

## Statement of Need

Mathematical models are important tools to describe and study real-world systems, allowing to understand underlying mechanisms (Gershenfeld & Gershenfeld, 1999; Kitano, 2002). They are commonly subject to unknown parameters that need to be estimated using observed data (Tarantola, 2005). The Bayesian framework allows doing so by updating prior beliefs about parameters by the likelihood of data given parameters. However, especially for complex stochastic models, evaluating the likelihood is often infeasible (Jagiella et al., 2017; Tavaré et al., 1997; D. J. Wilkinson, 2009). Thus, likelihood-free methods such as ABC have been developed (Andrieu & Roberts, 2009; Beaumont et al., 2002; Gourieroux et al., 1993; Price et al., 2018; Pritchard et al., 1999). ABC is widely applicable, as it only requires an executable forward process model, simulating data given model parameters. In a nutshell, ABC circumvents likelihood evaluation by accepting parameters if a distance between corresponding simulated and observed data is below a threshold (Sisson et al., 2018) (Figure 1). ABC is often combined with a sequential Monte-Carlo (ABC-SMC) scheme using importance sampling, which gradually reduces the acceptance threshold and thus improves the posterior approximation, while maintaining high acceptance rates (Del Moral et al., 2006; Sisson et al., 2007).

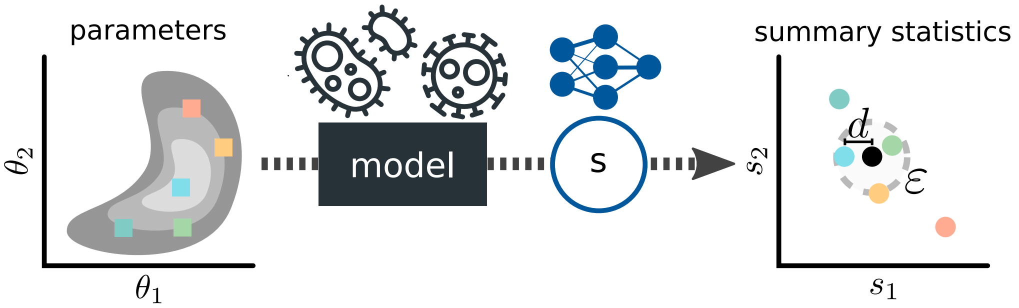

**Figure 1: Basic ABC algorithm.** Parameters $\theta \sim \pi(\theta)$ are sampled from the prior or a proposal distribution, and passed to a simulator model (here exemplarily biological systems) generating potentially stochastic simulated data according to the likelihood $y \sim \pi(y|\theta)$. These are optionally passed through a summary statistics function (here exemplarily a neural network model as employed in Schälte & Hasenauer (2022)) giving a low-dimensional representation $s(y)$. Summary statistics of simulated and observed data are compared via a distance metric $d$, and the underlying parameters accepted if the distance is below an acceptance threshold $\varepsilon$.

While conceptually simple and widely applicable, ABC is computationally expensive, as it requires simulating the forward model many times for different parameters. Its practical performance relies on a number of factors. pyABC implements at its core an ABC-SMC scheme based on Toni & Stumpf (2010) and facilicates robust and efficient inference for a broad spectrum of applications via robust methods and self-tuned choices of hyperparameters, reducing the need for manual tuning. An article on core features of pyABC was previously published (Emmanuel Klinger et al., 2018), discussing in particular adaptive transition kernels (Filippi et al., 2013), population sizes (E. Klinger & Hasenauer, 2017), and wall-time efficient parallelization via dynamic scheduling. pyABC is in use in a variety of fields, e.g. to model virus transmission on cellular (Imle et al., 2019) and population level (Kerr et al., 2021), neuron circuits (Bittner et al., 2021), cancer (Colom et al., 2021), gene expression (Coulier et al., 2021), axolotl regeneration (Costa et al., 2021), universe expansion (Bernardo & Said, 2021), cardiac electrophysiology (Cantwell et al., 2019), and bee colonies (Minucci et al., 2021).

Besides pyABC, there exist several other software packages implementing different algorithms in different languages, each with their own strengths, including notably in Python ABCpy (Dutta et al., 2017) and ELFI (Lintusaari et al., 2018), in Julia GpABC (Tankhilevich et al., 2020), and in R EasyABC (Jabot et al., 2013). In particular, ABCpy, ELFI and GpABC, which are actively maintained at the time of writing, implement various likelihood-free methods, including and beyond ABC-SMC, such as Bayesian optimization, synthetic likelihoods, or Gaussian process emulation. In contrast, pyABC focuses on providing a broadly applicable, efficient and robust implementation of ABC-SMC with various tailored algorithms. Exclusive to pyABC are, at the time of writing and to the best of our knowledge, the following features: While ABCpy and ELFI also allow distributed execution via static scheduling, only pyABC implements dynamic scheduling, further improving wall-time efficiency (Emmanuel Klinger et al., 2018). Further, most methods described in the below section are only implemented in pyABC: While e.g. ELFI implements basic adaptive distances based on Prangle (2017), only pyABC implements the robuster approach Schälte et al. (2021). While e.g. ABCpy implements basic regression-based summary statistics based on Fearnhead & Prangle (2012), only pyABC implements the approaches in Schälte & Hasenauer (2022) overcoming limitations in the method by Fearnhead & Prangle (2012), improving the posterior approximation and introducing sensitivity weights. Further, only pyABC implements the exact method under measurement noise Schälte & Hasenauer (2020).

## New Features

The methods discussed in the following have been newly implemented in pyABC (version 0.12), with details in the API documentation and Jupyter example notebooks accessible via the online documentation:

*Variations in different data scales and robustness to outliers.* In ABC, a distance metric is used to quantify differences between simulated and observed data. When simulations for different data points vary on different scales, highly variable ones dominate the acceptance decision. Prangle (2017) introduces a method to, in an ABC-SMC framework, iteratively update distance weights to normalize contributions. A further problem are outliers, i.e. errors in the measurement process that the model does not account for (Ghosh & Vogt, 2012; Motulsky & Christopoulos, 2003). Schälte et al. (2021) show that an approach adapted from Prangle (2017) can robustly identify outliers and reduce their impact. The approaches by Prangle (2017) and Schälte et al. (2021) are now implemented in pyABC.

*Identify informative data.* Instead of operating on the full data, in ABC often summary statistics, i.e. low-dimensional data representations, are employed (Blum et al., 2013). A particular line of approaches uses as statistics the outputs of inverse regression models of parameters on data, e.g. via linear regression (Fearnhead & Prangle, 2012), neural networks (Jiang et al., 2017), or Gaussian processes (Borowska et al., 2021). In Schälte & Hasenauer (2022), such approaches are combined with adaptive scale-normalization, and extended to achieve a higher-order posterior approximation. Further, inverse regression models are used to, instead of constructing summary statistics, inform robust sensitivity weights accounting for informativeness. All of the above approaches are now implemented in pyABC, with regression models interfaced via scikit-learn (Pedregosa et al., 2011).

*Accurate handling of noise.* The approximation error of ABC methods is often unclear. R. D. Wilkinson (2013) shows that ABC can be considered as giving exact inference under an implicit distance-induced noise model. In Schälte & Hasenauer (2020), this insight is used to develop an efficient ABC-SMC based exact inference scheme in the presence of measurement noise. The framework is now integrated into pyABC.

*Optimal transport distances.* Besides the above-mentioned adaptive distances, pyABC now in particular implements Wasserstein distances, which consider an optimal transport problem between distributions and may allow to circumvent the use of summary statistics (Bernton et al., 2019).

*Acceptance threshold selection.* Efficiency and convergence of ABC-SMC algorithms further depend on the acceptance threshold sequence. Silk et al. (2013) discuss that common schemes based on quantiles of previous values (Drovandi & Pettitt, 2011) can fail in the presence of local minima, and propose a method based on analyzing predicted acceptance rate curves. pyABC now implements a modified version, using importance sampling instead of unscented transform to predict the acceptance rate as a function of the threshold.

*Interoperability.* Not only algorithms, but also accessibility and interoperability determine the usefulness of a tool. Besides natural support of Python, and previously established support of R, pyABC now also provides an efficient interface to models written in Julia (Bezanson et al., 2017), to biochemical pathway models defined in SBML or COPASI format by using the COPASI toolbox (Hoops et al., 2006), and supports the PEtab inference standard (Schmiester, 2021), currently only for the ODE simulator AMICI (Fröhlich et al., 2021). Finally, it allows to connect to models written in arbitrary languages and frameworks via file exchange.

## Availability and Development

pyABC is being developed open-source under a 3-clause BSD license. The code, designed to be highly modular and extensible, is hosted on GitHub and continuously tested. Extensive documentation is hosted on Read the Docs, including API documentation and numerous Jupyter notebooks containing tutorials, outlining features, and showcasing applications.

## Acknowledgements

We thank many collaboration partners and pyABC users for valuable input, in particular Frank Bergmann for the COPASI wrapper, and Elba Raimúndez for fruitful discussions. This work was supported by the German Federal Ministry of Education and Research (BMBF) (FitMultiCell/031L0159 and EMUNE/031L0293) and the German Research Foundation (DFG) under Germany's Excellence Strategy (EXC 2047 390873048 and EXC 2151 390685813).

## References

Andrieu, C., & Roberts, G. O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, *37*(2), 697–725. https://doi.org/10.1214/07-AOS574

Beaumont, M. A., Zhang, W., & Balding, D. J. (2002). Approximate Bayesian Computation in Population Genetics. *Genetics*, *162*(4), 2025–2035. https://doi.org/10.1093/genetics/162.4.2025

Bernardo, R. C., & Said, J. L. (2021). Towards a model-independent reconstruction approach for late-time hubble data. *Journal of Cosmology and Astroparticle Physics*, *2021*(08), 027. https://doi.org/10.1088/1475-7516/2021/08/027

Bernton, E., Jacob, P. E., Gerber, M., & Robert, C. P. (2019). Approximate Bayesian computation with the Wasserstein distance. *J. Roy. Stat. Soc. B (Statistical Methodology)*, *81*(2), 235–269. https://doi.org/10.1111/rssb.12312

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, *59*(1), 65–98. https://doi.org/10.1137/141000671

Bittner, S. R., Palmigiano, A., Piet, A. T., Duan, C. A., Brody, C. D., Miller, K. D., & Cunningham, J. (2021). Interrogating theoretical models of neural computation with emergent property inference. *Elife*, *10*, e56265. https://doi.org/10.7554/eLife.56265

Blum, M. G., Nunes, M. A., Prangle, D., & Sisson, S. A. (2013). A comparative review of dimension reduction methods in approximate Bayesian computation. *Statistical Science*, *28*(2), 189–208. https://doi.org/10.1214/12-STS406

Borowska, A., Giurghita, D., & Husmeier, D. (2021). Gaussian process enhanced semi-automatic approximate Bayesian computation: Parameter inference in a stochastic differential equation system for chemotaxis. *Journal of Computational Physics*, *429*, 109999. https://doi.org/10.1016/j.jcp.2020.109999

Cantwell, C. D., Mohamied, Y., Tzortzis, K. N., Garasto, S., Houston, C., Chowdhury, R. A., Ng, F. S., Bharath, A. A., & Peters, N. S. (2019). Rethinking multiscale cardiac electrophysiology with machine learning and predictive modelling. *Computers in Biology and Medicine*, *104*, 339–351. https://doi.org/10.1016/j.compbiomed.2018.10.015

Colom, B., Herms, A., Hall, M., Dentro, S., King, C., Sood, R., Alcolea, M., Piedrafita, G., Fernandez-Antoran, D., Ong, S., & others. (2021). Mutant clones in normal epithelium outcompete and eliminate emerging tumours. *Nature*, *598*(7881), 510–514. https://doi.org/10.1038/s41586-021-03965-7

Costa, E. C., Otsuki, L., Albors, A. R., Tanaka, E. M., & Chara, O. (2021). Spatiotemporal control of cell cycle acceleration during axolotl spinal cord regeneration. *Elife*, *10*, e55665. https://doi.org/10.7554/eLife.55665

Coulier, A., Hellander, S., & Hellander, A. (2021). A multiscale compartment-based model of stochastic gene regulatory networks using hitting-time analysis. *The Journal of Chemical Physics*, *154*(18), 184105. https://doi.org/10.1063/5.0010764

Del Moral, P., Doucet, A., & Jasra, A. (2006). Sequential Monte Carlo samplers. *J. R. Stat. Soc. B*, *68*(3), 411–436. https://doi.org/10.1111/j.1467-9868.2006.00553.x

Drovandi, C. C., & Pettitt, A. N. (2011). Estimation of parameters for macroparasite population evolution using approximate Bayesian computation. *Biometrics*, *67*(1), 225–233. https://doi.org/10.1111/j.1541-0420.2010.01410.x

Dutta, R., Schoengens, M., Onnela, J.-P., & Mira, A. (2017). ABCpy: A user-friendly, extensible, and parallel library for approximate Bayesian computation. *Proceedings of the Platform for Advanced Scientific Computing Conference*, 8:1–8:9. https://doi.org/10.1145/3093172.3093233

Fearnhead, P., & Prangle, D. (2012). Constructing summary statistics for approximate Bayesian computation: Semi-automatic approximate Bayesian computation. *J. R. Stat. Soc. B*, *74*(3), 419–474. https://doi.org/10.1111/j.1467-9868.2011.01010.x

Filippi, S., Barnes, C. P., Cornebise, J., & Stumpf, M. P. (2013). On optimality of kernels for approximate Bayesian computation using sequential Monte Carlo. *Stat. Appl. Genet. Mol.*, *12*(1), 87–107. https://doi.org/10.1515/sagmb-2012-0069

Fröhlich, F., Weindl, D., Schälte, Y., Pathirana, D., Paszkowski, Ł., Lines, G. T., Stapor, P., & Hasenauer, J. (2021). AMICI: high-performance sensitivity analysis for large ordinary differential equation models. *Bioinformatics*, *btab227*. https://doi.org/10.1093/bioinformatics/btab227

Gershenfeld, N. A., & Gershenfeld, N. (1999). *The nature of mathematical modeling*. Cambridge university press.

Ghosh, D., & Vogt, A. (2012). Outliers: An evaluation of methodologies. *Joint Statistical Meetings*, 3455–3460.

Gourieroux, C., Monfort, A., & Renault, E. (1993). Indirect inference. *J. Appl. Econ.*, *8*(S1), S85–S118. https://doi.org/10.1002/jae.3950080507

Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., & Kummer, U. (2006). COPASI – a COmplex PAthway SImulator. *Bioinformatics*, *22*(24), 3067–3074. https://doi.org/10.1093/bioinformatics/btl485

Imle, A., Kumberger, P., Schnellbächer, N. D., Fehr, J., Carrillo-Bustamante, P., Ales, J., Schmidt, P., Ritter, C., Godinez, W. J., Müller, B., & others. (2019). Experimental and computational analyses reveal that environmental restrictions shape HIV-1 spread in 3D cultures. *Nature Communications*, *10*(1), 2144. https://doi.org/10.1038/s41467-019-09879-3

Jabot, F., Faure, T., & Dumoulin, N. (2013). EasyABC: Performing efficient approximate Bayesian computation sampling schemes using R. *Methods Ecol. Evol.*, *4*(7), 684–687. https://doi.org/10.1111/2041-210X.12050

Jagiella, N., Rickert, D., Theis, F. J., & Hasenauer, J. (2017). Parallelization and high-performance computing enables automated statistical inference of multi-scale models. *Cell Systems*, *4*(2), 194–206. https://doi.org/10.1016/j.cels.2016.12.002

Jiang, B., Wu, T., Zheng, C., & Wong, W. H. (2017). Learning summary statistic for approximate Bayesian computation via deep neural network. *Statistica Sinica*, 1595–1618. https://doi.org/10.5705/ss.202015.0340

Kerr, C. C., Stuart, R. M., Mistry, D., Abeysuriya, R. G., Rosenfeld, K., Hart, G. R., Núñez, R. C., Cohen, J. A., Selvaraj, P., Hagedorn, B., & others. (2021). Covasim: An agent-based model of COVID-19 dynamics and interventions. *PLOS Computational Biology*, *17*(7), e1009149. https://doi.org/10.1371/journal.pcbi.1009149

Kitano, H. (2002). Systems biology: A brief overview. *Science*, *295*(5560), 1662–1664. https://doi.org/10.1126/science.1069492

Klinger, E., & Hasenauer, J. (2017). A scheme for adaptive selection of population sizes in Approximate Bayesian Computation - Sequential Monte Carlo. In J. Feret & H. Koeppl (Eds.), *Computational methods in systems biology. CMSB 2017* (Vol. 10545). Springer, Cham. https://doi.org/10.1007/978-3-319-67471-1_8

Klinger, Emmanuel, Rickert, D., & Hasenauer, J. (2018). pyABC: Distributed, likelihood-free inference. *Bioinformatics*, *34*(20), 3591–3593. https://doi.org/10.1093/bioinformatics/bty361

Lintusaari, J., Vuollekoski, H., Kangasrääsio, A., Skytén, K., Jarvenpää, M., Marttinen, P., Gutmann, M. U., Vehtari, A., Corander, J., & Kaski, S. (2018). Elfi: Engine for likelihood-free inference. *Journal of Machine Learning Research*, *19*(16), 1–7.

Minucci, J. M., Curry, R., DeGrandi-Hoffman, G., Douglass, C., Garber, K., & Purucker, S. T. (2021). Inferring pesticide toxicity to honey bees from a field-based feeding study using a colony model and Bayesian inference. *Ecological Applications*, *31*(8), e02442. https://doi.org/10.1002/eap.2442

Motulsky, H., & Christopoulos, A. (2003). *Fitting models to biological data using linear and nonlinear regression: A practical guide to curve fitting*. GraphPad Software Inc., San Diego CA.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Prangle, D. (2017). Adapting the ABC distance function. *Bayesian Analysis*, *12*(1), 289–309. https://doi.org/10.1214/16-BA1002

Price, L. F., Drovandi, C. C., Lee, A., & Nott, D. J. (2018). Bayesian synthetic likelihood. *J. Comp. Graph. Stat.*, *27*(1), 1–11. https://doi.org/10.1080/10618600.2017.1302882

Pritchard, J. K., Seielstad, M. T., Perez-Lezaun, A., & Feldman, M. W. (1999). Population growth of human Y chromosomes: A study of Y chromosome microsatellites. *Molecular Biology and Evolution*, *16*(12), 1791–1798. https://doi.org/10.1093/oxfordjournals.molbev.a026091

Schälte, Y., Alamoudi, E., & Hasenauer, J. (2021). Robust adaptive distance functions for approximate Bayesian inference on outlier-corrupted data. *bioRxiv*. https://doi.org/10.1101/2021.07.29.454327

Schälte, Y., & Hasenauer, J. (2020). Efficient exact inference for dynamical systems with noisy measurements using sequential approximate Bayesian computation. *Bioinformatics*, *36*(Supplement 1), i551–i559. https://doi.org/10.1093/bioinformatics/btaa397

Schälte, Y., & Hasenauer, J. (2022). Informative and adaptive distances and summary statistics in sequential approximate Bayesian computation. *bioRxiv*. https://doi.org/10.1101/2022.03.18.484896

Schmiester, Y. A. B., Leonard AND Schälte. (2021). PEtab—interoperable specification of parameter estimation problems in systems biology. *PLOS Computational Biology*, *17*(1), 1–10. https://doi.org/10.1371/journal.pcbi.1008646

Silk, D., Filippi, S., & Stumpf, M. P. H. (2013). Optimizing threshold-schedules for sequential approximate Bayesian computation: Applications to molecular systems. *Stat. Appl. Genet.*

*Mol. Biol.*, *12*(5), 603–618. https://doi.org/10.1515/sagmb-2012-0043

Sisson, S. A., Fan, Y., & Beaumont, M. (2018). *Handbook of approximate Bayesian computation*. Chapman; Hall/CRC.

Sisson, S. A., Fan, Y., & Tanaka, M. M. (2007). Sequential Monte Carlo without likelihoods. *Proc. Natl. Acad. Sci.*, *104*(6), 1760–1765. https://doi.org/10.1073/pnas.0607208104

Tankhilevich, E., Ish-Horowicz, J., Hameed, T., Roesch, E., Kleijn, I., Stumpf, M. P., & He, F. (2020). GpABC: A Julia package for approximate Bayesian computation with Gaussian process emulation. *Bioinformatics*, *36*(10), 3286–3287. https://doi.org/10.1093/bioinformatics/btaa078

Tarantola, A. (2005). *Inverse problem theory and methods for model parameter estimation*. SIAM.

Tavaré, S., Balding, D. J., Griffiths, R. C., & Donnelly, P. (1997). Inferring coalescence times from DNA sequence data. *Genetics*, *145*(2), 505–518. https://doi.org/10.1093/genetics/145.2.505

Toni, T., & Stumpf, M. P. H. (2010). Simulation-based model selection for dynamical systems in systems and population biology. *Bioinformatics*, *26*(1), 104–110. https://doi.org/10.1093/bioinformatics/btp619

Wilkinson, D. J. (2009). Stochastic modelling for quantitative description of heterogeneous biological systems. *Nat. Rev. Genet.*, *10*(2), 122–133. https://doi.org/10.1038/nrg2509

Wilkinson, R. D. (2013). Approximate Bayesian computation (ABC) gives exact results under the assumption of model error. *Stat. Appl. Gen. Mol. Bio.*, *12*(2), 129–141. https://doi.org/10.1515/sagmb-2013-0010