



An Automatic Approximate Bayesian Computation Approach Using Metric Learning

W. González-Vanegas^(✉), A. Álvarez-Meza, and A. Orozco-Gutiérrez

Faculty of Engineerings, Automatic Research Group,
Universidad Tecnológica de Pereira, Pereira, Colombia
wilgonzalez@utp.edu.co

Abstract. Recent progress in Bayesian inference has allowed for accurate posterior estimations in complex situations with no idea about a likelihood function. Currently, Approximate Bayesian Computation (ABC) techniques have emerged as a widely used set of free-likelihood methods. Although there is a large number of different ABC-based approaches across the literature, all they have in common a hard dependence on free parameters selection, demanding for expensive tuning procedures such as grid search or cross-validation. Here, we introduce an Automatic Metric Learning-based ABC approach, termed AML-ABC. Namely, AML-ABC matches the simulation and observation spaces within an ABC-based framework. Attained results on a synthetic dataset and a real-world ecological system show that our approach is a competitive method compared to other non-automatic state-of-the-art ABC techniques.

Keywords: Approximate Bayesian Computation · Kernel methods · Metric learning · Non-linear dynamic system · Statistical inference

1 Introduction

Bayesian-based statistical inference tasks require the calculation of the likelihood function, which performs an important role as long as it states the probability of the observed data under a particular model. Therefore, the Bayes' theorem leverages the inclusion of a priori knowledge about the studied phenomenon into the posterior distribution. Indeed, straightforward models gather an analytic expression for the likelihood function facilitating the evidence assessment; then, the posterior can be precisely computed. Notwithstanding, for complex models, to find an exact formula for the likelihood function is often intractable [13].

To deal with this intractability, free-likelihood techniques like Approximate Bayesian Computation (ABC) have emerged. ABC-based methods assess an auxiliary model with different parameter values drawn from a prior distribution to compute simulations that are compared to the observed data [13]. In particular, this comparison can be performed using statistics that summarize and characterize the information over large features and observations [5, 10]. However, the

selection of proper and sufficient summary statistic could be difficult for complex models, demanding for alternative approaches that rely on kernel functions to embed and compare distributions into a Reproducing Kernel Hilbert Space (RKHS) [4, 9]. Recent advances in ABC-based inference have introduced kernel methods to accomplish more accurate posterior estimations. Authors in [6] developed a surrogate model as synthetic likelihood to define an adequate number of simulations within the ABC procedure via a Gaussian process-based framework. Mitrovic et al. [7] modeled the functional relationship between simulations and the optimal choice of summary statistics to encode the structure of a generative model using a kernel ridge regression for conditional distributions. Nonetheless, the techniques mentioned above require the estimation of different parameters related to the similarity computation among simulations to approximate the posterior. Then, expensive tuning procedures as grid search and cross-validation are carried out. Moreover, the user requires a vast knowledge concerning the ABC algorithm and the studied data to properly tune the free parameters, yielding to a high influence in the quality of the posterior approximation.

Here, we introduce an automatic version of an ABC algorithm to support Bayesian inference. Our approach, named AML-ABC, comprises a metric learning stage based on a Centered Kernel Alignment (CKA) to assess the matching between similarities defined over parameters and simulations [3]. Besides, a Mahalanobis distance is computed through CKA, and a graph representation is utilized to highlight local dependencies from both parameter and simulation spaces in ABC. Achieved results on synthetic and real-world inference problems demonstrate that our automatic extension of ABC infers competitive posteriors without requiring any manually fixing of free parameters.

The remainder of this paper is organized as follows: Sect. 2 describes the mathematical background. Section 3 describes the experimental set-up and the obtained result. Finally, the conclusions are outlined in Sect. 4.

2 Materials and Methods

ABC Fundamentals. In any Bayesian inference task, the central aim concerns the computation of a posterior $p(\theta|y)$, using a prior distribution $\zeta(\theta)$ and a likelihood function $p(y|\theta)$, where $y \in \mathcal{Y}$ stands for the observed data and $\theta \in \Theta$ holds the model parameters. Nonetheless, when the likelihood is intractable, neither exact nor sampled posterior $p(\theta|y) \propto p(y|\theta)\zeta(\theta)$ can be calculated. ABC approaches aim to facilitate such an inference via simulation of the likelihood through a generative model represented by a conditional probability $p(x|\theta)$, where $x \in \mathcal{X}$ is a random variable standing for the simulated data [13]. Fundamentally, an ABC-based framework relies on the acceptance and rejection of simulated samples x using a distance function $d_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}^+$. In turn, an approximate posterior can be estimated such that: $\hat{p}(\theta|y; \epsilon) \propto \hat{p}(y|\theta; \epsilon)\zeta(\theta)$, where $\hat{p}(y|\theta; \epsilon) = \int_{\mathcal{B}(y; \epsilon)} p(x|\theta)dx$, $\mathcal{B}(y; \epsilon) = \{x : d_{\mathcal{Y}}(x, y) < \epsilon\}$, and $\epsilon \in \mathbb{R}^+$. Note that setting the value of ϵ is a crucial stage for obtaining an accurate posterior.

On the other hand, most of the times it is difficult to apply a distance directly on \mathcal{Y} due to a large number of samples and features in real data. In such a case,

some strategies use a mapping $s = \vartheta(x)$ before calculating the distance, where $s \in \mathcal{S}$ is a feature space and $\vartheta : \mathcal{Y} \rightarrow \mathcal{S}$ [5]. However, using $\vartheta(x)$ often leaks information for complex models. Then, some ABC-based approaches approximate $\hat{p}(y|\theta; \epsilon)$ as the convolution of the true likelihood $p(y|\theta)$ and a kernel function $\kappa : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$, which imposes a constraint to the rejection of samples as the inner product $\kappa(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{H}}$ in a Reproducing Kernel Hilbert Space (RKHS), \mathcal{H} , where $\phi : \mathcal{Y} \rightarrow \mathcal{H}$ [9]. In practice, given N samples $\{x_n \sim P_{X_n}\}_{n=1}^N$ drawn from $p(x|\theta_n)$, with $\theta_n \sim \zeta(\theta)$, and the observation $y \sim P_Y$, a weighted sample set $\Psi = \{(\theta_n, w_n)\}_{n=1}^N$ is calculated by fixing:

$$w_n = \kappa_G(d_{\mathcal{H}}(P_{X_n}, P_Y); \epsilon) / \sum_{n=1}^N \kappa_G(d_{\mathcal{H}}(P_{X_n}, P_Y); \epsilon), \quad (1)$$

where κ_G is a Gaussian kernel with bandwidth ϵ . Eq. (1) is used to approximate $p(\theta|y)$ via posterior expectation as: $\hat{p}(\theta|y) = \sum_{n=1}^N w_n \kappa_G(d_e(\theta, \theta_n); \sigma_\theta)$, where d_e stands for the Euclidean distance and $\sigma_\theta \in \mathbb{R}^+$. Moreover, $d_{\mathcal{H}}(P_{X_n}, P_Y)$ represents a distance over distributions.

Automatic ABC Using Metric Learning. To avoid the influence of the ϵ value and the kernel parameters while computing the ABC-based posterior as in Eq. (1), we introduce an Automatic Metric Learning (AML) approach in the context of ABC, termed AML-ABC, for enhancing and automating the inference task. The idea behind AML-ABC is to include the information contained in the candidates $\{\theta_n\}_{n=1}^N$ to improve the comparison stage carried out over simulations and observations. Let $\Psi = \{\theta_n, x_n\}_{n=1}^N$ be the set of N candidates $\theta_n \in \mathbb{R}^P \sim \zeta(\theta)$ and their corresponding simulations $x_n \in \mathbb{R}^Q \sim p(x|\theta)$. Further, let the kernel function $\kappa_\theta : \Theta \times \Theta \rightarrow \mathbb{R}^+$ be a similarity measure between candidates in Θ , that define the kernel matrix $\mathbf{K}_\theta \in \mathbb{R}^{N \times N}$ holding elements:

$$\kappa_\theta(\theta_n, \theta_{n'}) = \begin{cases} \exp(-d_\Theta^2(\theta_n, \theta_{n'})), & \theta_n \in \Omega_{n'} \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where $\Omega_{n'}$ is a set holding the M -nearest neighbors of $\theta_{n'}$ in the sense of the distance $d_\Theta : \Theta \times \Theta \rightarrow \mathbb{R}^+$. In this paper, to avoid large variations among components of θ_n we rely on the Mahalanobis distance $d_\Theta^2(\theta_n, \theta_{n'}) = (\theta_n - \theta_{n'})^T \Sigma_\Theta^{-1} (\theta_n - \theta_{n'})$, where $\Sigma_\Theta \in \mathbb{R}^{P \times P}$ is the sample covariance matrix of $\{\theta_n\}_{n=1}^N$. Concerning the feature space \mathcal{S} , we assess the similarity via the kernel $\kappa_s : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}^+$, $\kappa_s(\vartheta(x_n), \vartheta(x_{n'})) = \exp(-d_{\mathcal{S}}^2(\vartheta(x_n), \vartheta(x_{n'})))$, to build the matrix $\mathbf{K}_s \in \mathbb{R}^{N \times N}$, where $d_{\mathcal{S}}^2 : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}^+$ and $\vartheta : \mathcal{Y} \rightarrow \mathcal{S}$ is a feature mapping. To perform the pairwise comparison between simulations in \mathcal{S} we use the Mahalanobis distance of the form [2]:

$$d_{\mathcal{S}}^2(\vartheta(x_n), \vartheta(x_{n'})) = (\vartheta(x_n) - \vartheta(x_{n'}))^T \mathbf{A} \mathbf{A}^T (\vartheta(x_n) - \vartheta(x_{n'})), \quad (3)$$

where $\Sigma_{\mathcal{S}}^{-1} = \mathbf{A} \mathbf{A}^T$ stands for the inverse covariance matrix of $\vartheta(x_n) \in \mathbb{R}^D$ and $\mathbf{A} \in \mathbb{R}^{D \times d}$. In this sense, we use the information concerning the similarity over candidates in Θ , represented via \mathbf{K}_θ , to state the notion of similarity over

simulations and observation in \mathcal{S} , represented via \mathbf{K}_s . In particular, we use a CKA-based measure between the above kernel matrices as [3]:

$$\hat{\rho}(\mathbf{K}_\theta, \mathbf{K}_s) = \frac{\langle \bar{\mathbf{K}}_\theta, \bar{\mathbf{K}}_s \rangle_F}{\sqrt{\langle \bar{\mathbf{K}}_\theta \bar{\mathbf{K}}_\theta \rangle_F \langle \bar{\mathbf{K}}_s \bar{\mathbf{K}}_s \rangle_F}}, \quad (4)$$

where $\bar{\mathbf{K}}$ stands for the centered kernel as $\bar{\mathbf{K}} = \tilde{\mathbf{I}}\mathbf{K}\tilde{\mathbf{I}}$, being $\tilde{\mathbf{I}} = \mathbf{I} - \mathbf{1}^\top \mathbf{1}/N$ the empirical centering matrix, $\mathbf{I} \in \mathbb{R}^{N \times N}$ is the identity matrix, and $\mathbf{1} \in \mathbb{R}^N$ is the all-ones vector. Moreover, The notation $\langle \cdot, \cdot \rangle_F$ represents the matrix-based Frobenius norm. In Eq. (4), $\hat{\rho}(\cdot, \cdot)$ is a data driven estimator that aims to quantify the similarity between the parameter space and the feature space. To find the projection matrix \mathbf{A} , we consider the following optimization problem:

$$\hat{\mathbf{A}} = \arg \max_{\mathbf{A}} \log(\hat{\rho}(\mathbf{K}_s(\mathbf{A}), \mathbf{K}_\theta)), \quad (5)$$

where the logarithm function is used for mathematical convenience. The optimization problem in Eq. (5) can be solved using a gradient descent-based approach [2]. Moreover, we form the weighted sample set $\Psi = \{(\theta_n, w_n)\}_{n=1}^N$ by fixing $w_n = \kappa_E(z, z_n) / \sum_{n=1}^N \kappa_E(z, z_n)$, where $\kappa_E : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is a similarity kernel defined as:

$$\kappa_E(z, z_n) = \begin{cases} \exp(-\|z - z_n\|_2^2), & z_n \in \Upsilon \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

where Υ is a set holding the M -nearest neighbors of $z = \vartheta(y)^\top \hat{\mathbf{A}}$ in the sense of the Euclidean distance. Algorithm 1 summarizes the AML-ABC approach.

Algorithm 1. AML-ABC algorithm

Input: Observed data: y , prior: $\zeta(\theta)$, mapping: ϑ , M -nearest neighbors, width: σ_θ .

Output: Posterior estimation: $\hat{p}(\theta|y)$.

Metric learning stage:

1: $\Psi' = \{(\theta'_n, x'_n)\}_{n=1}^N$; $\theta'_n \sim \zeta(\theta)$, $x'_n \sim p(x|\theta'_n)$ ▷ Draw training data.

2: $\hat{\mathbf{A}} = \arg \max_{\mathbf{A}} \log(\hat{\rho}(\mathbf{K}_s(\mathbf{A}), \mathbf{K}_\theta))$ ▷ Compute CKA based on ϑ , M , θ'_n , and x'_n .

Inference stage:

3: $\Psi = \{(\theta_n, x_n)\}_{n=1}^N$; $\theta_n \sim \zeta(\theta)$, $x_n \sim p(x|\theta_n)$ ▷ Draw simulated data.

4: $z = \vartheta(y)^\top \hat{\mathbf{A}}$ ▷ Project features of observed data

5: **for** $n = 1, \dots, N$ **do**

6: $z_n = \vartheta(x_n)^\top \hat{\mathbf{A}}$ ▷ Project features of simulated data

7: $\tilde{w}_n = \kappa_E(z, z_n)$ ▷ Compute the n -th weight value.

8: **end for**

9: $w_n = \tilde{w}_n / \sum_{n=1}^N \tilde{w}_n$ ▷ Normalize the weights

10: $\hat{p}(\theta|y) = \sum_{n=1}^N w_n \kappa_G(d_c(\theta, \theta_n; \sigma_\theta))$ ▷ Compute the posterior.

3 Experiments and Results

To test the AML-ABC performance, we consider two experiments following [9]: a toy problem concerning synthetic data from a mixture model and a Bayesian inference problem for a real ecological dynamic system. To accomplish an automatic inference, we find the number of M -nearest neighbors as the median of the optimal number of neighbours per point according to the Local Neighborhood Selection (LNS) algorithm introduced in [1]. Moreover, the K2-ABC method is selected as benchmark due to its nice performance over other methods [9].

For the toy problem, we analyze a mixture of uniform distributions of the form: $p(x|\boldsymbol{\pi}) = \sum_{c=1}^C \pi_c \mathcal{U}(c-1, c)$, where $\boldsymbol{\pi} = \{\pi_c\}_{c=1}^C$ are the mixing coefficients holding $\sum_{c=1}^C \pi_c = 1$, and C is the number of components. Here, the aim is to estimate the posterior $p(\boldsymbol{\pi}|y)$ for $C=5$, given synthetic observations y drawn from the mixture with true parameters (target): $\boldsymbol{\pi}^* = \{0.25, 0.04, 0.33, 0.04, 0.34\}$. For concrete testing, we draw $N = 1000$ samples from a symmetric Dirichlet prior, $\boldsymbol{\pi} \sim \text{Dirichlet}(\mathbf{1})$, and then used the mixture model to form the simulated data by drawing 400 observations for each prior candidate. Moreover, we employ a histogram with 10 bins as feature mapping in AML-ABC, and kernel widths $\gamma = 0.1$, $\epsilon = 0.001$ in K2-ABC [9]. As quantitative assessment, we use the Euclidean distance $\mathcal{E} = \|\boldsymbol{\pi}^* - \hat{\boldsymbol{\pi}}\|_2$, where $\hat{\boldsymbol{\pi}}$ is the expected value of the posterior using the weights $\{w_n\}_{n=1}^N$ obtained by each method.

For the real dataset experiment, we consider the problem of inferring the dynamics of an adult blowfly population as introduced in [14]. In particular, the population dynamics are modelled by a discretised differential equation of the form: $N_{t+1} = PN_{t-\tau} \exp(-N_{t-\tau}/N_0)e_t + N_t \exp(-\delta\epsilon_t)$, with N_{t+1} denoting the observation time at $t+1$ which is determined by the time-lagged observations N_t and $N_{t-\tau}$, where e_t and ϵ_t stand for Gamma distributed noise $e_t \sim \mathcal{G}(1/\sigma_p^2, \sigma_p^2)$ and $\epsilon_t \sim \mathcal{G}(1/\sigma_d^2, \sigma_d^2)$. Here, the aim is to estimate the posterior of the parameters $\boldsymbol{\theta} = \{P, N_0, \sigma_d, \sigma_p, \tau, \delta\}$ given observed data concerning a time series of 180 observations¹. We adopt Log-normal distributions for setting priors over $\boldsymbol{\theta}$ [6]: $\log P \sim \mathcal{N}(2, 2^2)$, $\log N_0 \sim \mathcal{N}(6, 1)$, $\log \sigma_d \sim \mathcal{N}(-0.5, 1)$, $\log \sigma_p \sim \mathcal{N}(-0.5, 1)$, $\log \tau \sim \mathcal{N}(2.7, 1)$, $\log \delta \sim \mathcal{N}(-1, 0.4^2)$. For AML-ABC we draw $N = 5000$ samples from the prior and then assess the model to form the simulated data by drawing 180 observations for each prior candidate. Besides, as feature mapping, we selected the 10 statistics used in [9]. Moreover, we use the Euclidean distance $\mathcal{E} = \|\vartheta(y) - \vartheta(x_n|\hat{\boldsymbol{\theta}})\|_2$ as quantitative assessment, where $x_n|\hat{\boldsymbol{\theta}}$ is a simulation from the model given the expected value of the posterior using each method. In particular, due to fluctuations produced by ϵ_t and e_t , we draw 100 simulation for each method and compute the median and standard deviation for \mathcal{E} [9].

Toy Problem Results. Since this is a controlled experiment with known parameters $\boldsymbol{\pi}^*$, we can find the best possible performance of our AML-ABC by running the inference stage in Algorithm 1 with $\tilde{w}_n = \kappa_E(\boldsymbol{\pi}^*, \boldsymbol{\pi}_n)$. We refer to this approach as *Best*. The previous setting is equivalent to think that the CKA between

¹ Available on the supplementary materials of [14].

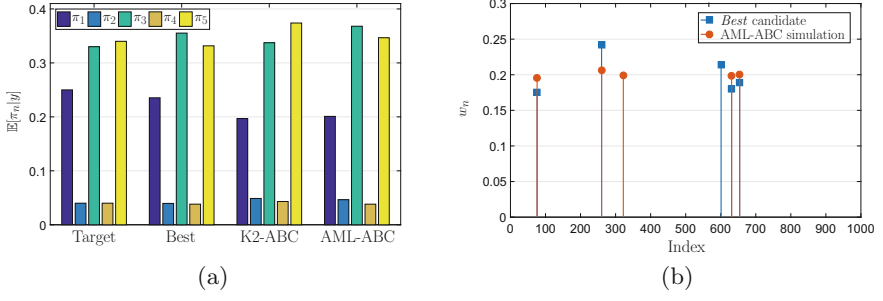


Fig. 1. Uniform mixture model results. (a) Estimated mean posterior of mixing coefficients using various methods (b) Weights of the 5 nearest neighbors in AML-ABC.

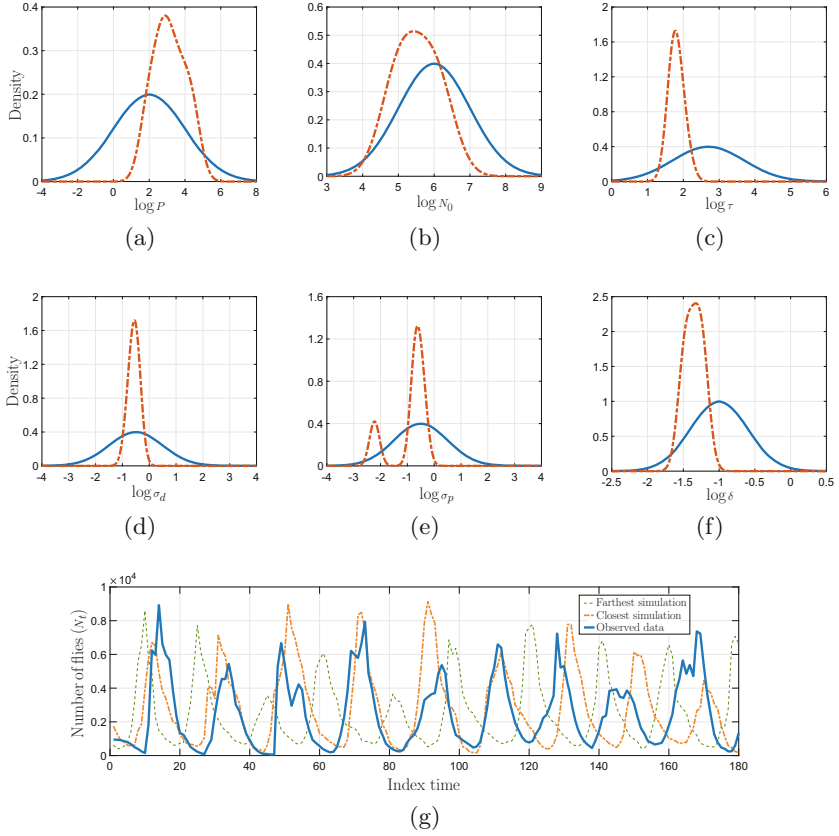


Fig. 2. Non-linear ecological dynamic system results. (a)–(f) Prior distribution (solid line) and AML-ABC-based posterior estimation (dashed line) of model parameters in the log-space. (g) Some realizations from the model using the expected value of the parameters found via AML-ABC.

Table 1. Performance of different ABC schemes over the blowfly dataset.

Method	Median (\mathcal{E}) \pm std (\mathcal{E})
Kernel ABC [8]	5.2 ± 3.0
Indirect score ABC [11]	2.1 ± 1.8
Semi-Automatic ABC [10]	1.9 ± 1.2
Synthetic Likelihood ABC [14]	1.7 ± 1.3
K2-ABC [9]	1.0 ± 0.8
AML-ABC	1.2 ± 0.4

\mathbf{K}_θ and \mathbf{K}_s is perfect ($\mathbf{K}_\theta = \mathbf{K}_s$). Figure 1 shows the *Best* performance along with K2-ABC and AML-ABC results over the uniform mixture problem. In Fig. 1(a), the expected value of the posterior computed for all methods is close to the target. In particular, we obtained $\mathcal{E}_{Best} = 0.030$, $\mathcal{E}_{K2-ABC} = 0.063$, and $\mathcal{E}_{AML-ABC} = 0.064$. These results show that our AML-ABC is a competitive estimator to K2-ABC with a significant advantage concerning the automatic selection of free parameters. In addition, to provide a better understanding of the AML-ABC effectiveness, in Fig. 1(b) we provide the weights for the 5 nearest neighbors (according to the LNS algorithm) used to compute the posteriors. As noted, the majority of the chosen simulations for AML-ABC match the selected candidates using the *Best*, although our approach never observes the target.

Real Dataset Results. Inferring the model parameters in this blowfly dataset is a very challenging task since the system dynamics can easily move from stable to chaotic regimes [6, 14]. This states an interesting scenario to test the performance and robustness of the AML-ABC. In Figs. 2(a) to (f), we provide the prior and the posterior approximation for each parameter fixing σ_θ according to [12]. Notice how our proposal updates the beliefs about the model parameters leading to more concentrated posteriors. In the case of $\log \sigma_p$, two modes reflect different intervals with probable values for driving the noise realization associated with egg production in the blowfly population. However, there is a predominant mode that states higher probabilities for this parameter. Moreover, Fig. 2(g) shows the closest and farthest simulation to the observed data from 100 realization used to compute \mathcal{E} , showing a posterior in stable regime. Finally, Table 1 shows the performance of AML-ABC compared to different ABC-based methods tested on the blowfly dataset by authors in [9], where clearly the proposed method is a quite competitive approach to K2-ABC.

4 Conclusions

We propose an automatic enhancement of the well-known ABC algorithm devoted to Bayesian inference called AML-ABC. In particular, we include a Metric Learning approach based on a CKA methodology to quantify the matching between parameter and simulation spaces. Then, a Mahalanobis distance

is learned through CKA and a graph representation is employed to reveal local relationships among parameter and simulation samples. Notably, our AML-ABC does not require the tuning of any free parameter. Besides, obtained results on a synthetic dataset and a real-world ecological system show the introduced AML-ABC is a competitive approach compared to other non-automatic state-of-the-art ABC methods. Future work includes the extension of AML-ABC for multi-dimensional problems and the inclusion of other dissimilarity measures, besides the Mahalanobis distance, to deal with complex and/or noisy data.

Acknowledgments. Research under grants provided by the project 1110-745-58696, funded by Colciencias, Colombia. Authors would like to thank the Master in Electrical Engineering program from Universidad Tecnológica de Pereira for partially funding this research. Moreover, author W. González-Vanegas was supported under the project E6-18-2, funded by Universidad Tecnológica de Pereira.

References

1. Álvarez-Meza, A., et al.: Global and local choice of the number of nearest neighbors in locally linear embedding. *Pattern Recogn. Lett.* **32**(16), 2171–2177 (2011)
2. Álvarez-Meza, A., et al.: Kernel-based relevance analysis with enhanced interpretability for detection of brain activity patterns. *Frontiers Neurosci.* **11**, 550 (2017)
3. Cortes, C., Mohri, M., Rostamizadeh, A.: Algorithms for learning kernels based on centered alignment. *JMLR* **13**, 795–828 (2012)
4. González-Vanegas, W., Alvarez-Meza, A., Orozco-Gutierrez, Á.: Sparse hilbert embedding-based statistical inference of stochastic ecological systems. In: Mendoza, M., Velastín, S. (eds.) *CIARP 2017. LNCS*, vol. 10657, pp. 255–262. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-75193-1_31
5. Joyce, P., Marjoram, P., et al.: Approximately sufficient statistics and Bayesian computation. *Stat. Appl. Genet. Molec.* **7**(1), 26 (2008)
6. Meeds, E., Welling, M.: GPS-ABC: Gaussian process surrogate approximate Bayesian computation. arXiv preprint [arXiv:1401.2838](https://arxiv.org/abs/1401.2838) (2014)
7. Mitrovic, J., Sejdinovic, D., Teh, Y.W.: DR-ABC: approximate Bayesian computation with kernel-based distribution regression. In: *International Conference on Machine Learning, ICML 2016*, vol. 48, pp. 1482–1491 (2016)
8. Nakagome, S., et al.: Kernel approximate Bayesian computation in population genetic inferences. *Stat. Appl. Genet. Mol. Biol.* **12**(6), 667–678 (2013)
9. Park, M., Jitkrittum, W., Sejdinovic, D.: K2-ABC: approximate Bayesian computation with kernel embeddings. arXiv preprint [arXiv:1502.02558](https://arxiv.org/abs/1502.02558) (2015)
10. Paul, F., Dennis, P.: Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate bayesian computation. *J. Roy. Stat. Soc. Ser. B (Stat. Methodol.)* **74**(3), 419–474 (2012)
11. Pigorsch, E.G.C.: Approximate Bayesian computation with indirect summary statistics. Technical report (2013)
12. Shimazaki, H., Shinomoto, S.: Kernel bandwidth optimization in spike rate estimation. *J. Comput. Neurosci.* **29**(1–2), 171–182 (2010)
13. Turner, B.M., Van Zandt, T.: A tutorial on approximate Bayesian computation. *J. Math. Psychol.* **56**(2), 69–85 (2012)
14. Wood, S.N.: Statistical inference for noisy nonlinear ecological dynamic systems. *Nature* **466**(7310), 1102–1104 (2010)