

Gaussian process enhanced semi-automatic approximate Bayesian computation: parameter inference in a stochastic differential equation system for chemotaxis

Agnieszka Borowska^{a,*}, Diana Giurghita^a, Dirk Husmeier^{a,*}

^a*School of Mathematics and Statistics, University of Glasgow, Glasgow G12 8QQ, UK*

Abstract

Chemotaxis is a type of cell movement in response to a chemical stimulus which plays a key role in multiple biophysical processes, such as embryogenesis and wound healing, and which is crucial for understanding metastasis in cancer research. In the literature, chemotaxis has been modelled using biophysical models based on systems of nonlinear stochastic partial differential equations (NSPDEs), which are known to be challenging for statistical inference due to the intractability of the associated likelihood and the high computational costs of their numerical integration. Therefore, data analysis in this context has been limited to comparing predictions from NSPDE models to laboratory data using simple descriptive statistics. We present a statistically rigorous framework for parameter estimation in complex biophysical systems described by NSPDEs such as the one of chemotaxis. We adopt a likelihood-free approach based on approximate Bayesian computations with sequential Monte Carlo (ABC-SMC) which allows for circumventing the intractability of the likelihood. To find informative summary statistics, crucial for the performance of ABC, we propose to use a Gaussian process (GP) regression model. The interpolation provided by the GP regression turns out useful on its own merits: it relatively accurately estimates the parameters of the NSPDE model and allows for uncertainty quantification, at a very low computational cost. Our proposed methodology allows for a considerable part of computations to be completed before having observed any data, providing a practical toolbox to experimental scientists whose modes of operation frequently involve experiments and inference taking place at distinct points in time. In an application to externally provided synthetic data we demonstrate that the correction provided by ABC-SMC is essential for accurate estimation of some of the NSPDE model parameters and for more flexible uncertainty quantification.

Keywords: Biophysics, statistical inference, approximate Bayesian

*Corresponding authors:

Email addresses: Agnieszka.Borowska@glasgow.ac.uk (Agnieszka Borowska), Dirk.Husmeier@glasgow.ac.uk (Dirk Husmeier)

1. Introduction

Cell migration is a complex phenomenon which is frequently analysed as a biophysical system, e.g. using a reaction–diffusion model. Chemotaxis is a type of directed cell movement in which cells are steered by chemical signals [1]. It is of high interest to biophysicists as cell migration in response to a chemical gradient plays a key role in a range of critical processes, such as wound healing, embryogenesis and cancer metastasis. For this reason it has been a subject of active research, see e.g. Neilson et al. [2], Tweedy et al. [3], MacDonald et al. [4] and references therein. Neilson et al. [2] attribute chemotaxis to pseudopod formation and Tweedy et al. [3] demonstrate the impact of cell shapes on cell movement. MacDonald et al. [4] focus on approximating the solution to coupled bulk–surface reaction-diffusion equations by devising a novel finite element method. The previous literature has formalised biophysical hypotheses using a model based on a system of nonlinear stochastic partial differential equations (NSPDEs). NSPDE models are known to be challenging for statistical inference due to the intractability of the associated likelihood and the high computational costs of the numerical integration. Therefore, data analysis in this context has been limited to comparing predictions from NSPDE models to laboratory data using simple descriptive statistics, such as correlations of principal components extracted from Fourier descriptors of cell contours in Tweedy et al. [3], or assessing visual similarity between features extracted from simulated and real cells by Neilson et al. [2]. We aim to fill the gap for rigorous statistical inference in complex biophysical systems described by NSPDEs focusing on the model for chemotaxis.

We point out the main difference of our approach to more main-stream methods in physical modelling. Research in physics often starts from a mechanistic model and then aims to understand how different parameter regimes are related to certain features in the data, e.g. attractor characteristics, fractal dimensions and potential phase transitions. This type of research focuses on the model and views features in the data as emergent properties. Our research follows the opposite direction. Given the data, we aim to find the parameters of the model that are most consistent with the observations. Conceptually, these parameters are those that maximize the likelihood of the data (classical statistics) or lie in the high posterior probability region (Bayesian statistics). For complex nonlinear stochastic models, as studied in the present paper, the likelihood is analytically intractable, and data features, like attractor characteristics, are extracted for approximate inference. Thus, the features play a different role in our approach compared to a standard approach in physics, as rather than starting from the model to predict these features, we start from the features to calibrate the model.

One approach to parameter estimation in complex stochastic models is provided by the broad family of pseudo-marginal methods, see Andrieu and Roberts

[5], Beaumont [6] and Andrieu et al. [7]. Even if it is not possible to analytically evaluate the likelihood in the model of interest, like in the case of the NSPDE models for cell movement, these algorithms are guaranteed to converge to the exact posterior distribution provided that an unbiased and non-negative likelihood estimator can be devised [5]. However, obtaining such an estimator in the case of NSPDE models turns out computationally prohibitive due to the complexity and dimensionality of the system in question. Moreover, even if we were able to devise an estimator with both desired properties, it would likely be characterised by a huge variance (due to the high dimensionality of the system), leading to a slowly mixing Markov chain Monte Carlo algorithm, see Deligiannidis et al. [8].

A different approach to parameter inference in NSPDE models is provided by the so-called likelihood-free methods. These are simulation-based algorithms typically used to perform inference in complex stochastic models for which the likelihood is hard or impossible to evaluate but for which forward simulations are relatively easy to carry out. This category includes applications involving physical models which summarise and explore dynamic states of complex biophysical process. More specifically, approximate Bayesian computation (ABC) [9] and the methods based on synthetic likelihood [10, 11] have recently gained popularity in molecular biology and biophysics, with applications ranging from model calibration and exploration of leukocyte cell migration inside zebrafish embryos [12] to parameter inference in models describing cell motility and proliferation [11].

In ABC algorithms the likelihood calculation is replaced by simulating artificial data from the model given different parameter values and comparing these simulated datasets with the observed data. For high dimensional outputs, the comparison is typically made in terms of summary statistics, which are features extracted from the artificial and the observed datasets. If a chosen metric of the difference between the summary statistics is below a given threshold, the parameters used to generate the artificial dataset are considered to be a sample from an approximate posterior distribution. However, extracting low-dimensional summary statistics from the data turns out to be one of the most difficult aspects of ABC as, in general, this task is problem-specific and hence hard to generalise. A seminal step towards making summary statistics construction automatic was achieved by Fearnhead and Prangle [13], who propose a general methodology for constructing summary statistics called “semi-automatic ABC”. Their framework relies on using linear regression to combine several features extracted from the data into a single summary statistic, on which the ABC inference is based. An obvious limitation of this approach is restricting the space of summary statistics to its linear manifold, as this assumption is likely to provide very crude parameter estimates in particular for highly-nonlinear problems ubiquitous in biology and physics.

In the present paper, we first propose a new efficient approach to obtaining summary statistics for ABC based on a Gaussian process (GP) regression model, see Rasmussen and Williams [14] for a detailed treatment of GPs. Our framework extends the one of Fearnhead and Prangle [13] and is motivated by the

challenges of formal statistical inference for the biophysical cell migration model of Tweedy et al. [3]. Given a highly non-linear relationship between parameter and observation space in the cell migration problem, a flexible GP regression framework provides more accurate and efficient summary statistics for ABC. Second, the interpolation provided by the GP regression turns out useful on its own merits as it alone allows us to fairly accurately estimate the parameters of the NSPDE model, including approximate uncertainty quantification (though under the Gaussianity restriction), at a very low computational cost. Third, we demonstrate that the correction provided by the ABC simulations is required for more flexible uncertainty quantification (compared to using the GP regression only) and for accurate estimation of some of the NSPDE model parameters, in particular for those for which the prior calibration domain turns out too narrow compared to the ground-truth values.

We note that combining insights from GP regression and ABC analysis has already been proposed by e.g. Wilkinson [15]. However, Wilkinson [15] does not address the issue of constructing summary statistics for ABC, which – due to the high-dimensional observation space in the NSPDE model – is of crucial interest to us. A further difference is that the work of Wilkinson [15] is based on the history matching approach [see 16]. This technique is computationally efficient but greedy: it sequentially rules out parameter regions which are inconsistent with the data and may fail for multimodal distributions. Moreover, it does not provide a full description of the posterior distribution as it outputs “non-implausible regions” in which the parameters are weighted equally.

Our proposed methodology allows for a considerable part of computations to be completed before having observed any data. This provides a practical toolbox to experimental scientists whose modes of operation often involve experiments and inference taking place at distinct points in time, with long time necessary to set up equipment and develop a model and potentially little time for statistical inference. The methodological framework we discuss in the present paper allows this time, required for the acquisition of the data, to be exploited for carrying out precomputations relevant to inference from data before actually seeing any data. This substantially reduces the computational costs of inference that would otherwise have to be incurred at the time when the data become available. In an application to externally provided synthetic data we demonstrate that our computational inference framework can tackle heterogeneous high-dimensional data stemming from a biophysical model with a substantial level of complexity in a time-efficient manner.

The rest of this paper is organised as follows. In Section 2 we present the NSPDE model of cell movement developed by Neilson et al. [2] and Tweedy et al. [3] and we discuss the NSPDE simulator generating data from this system. We further discuss a typical pipeline in experimental sciences, in which experiments and inference are likely to be time-divided, and link it to the desirable pipeline of an inference scheme. We use the insights gained to develop our methodological framework, presented in Section 3, where we discuss the three essential steps in turn: feature extraction to reduce the dimensionality of the outputs from the NSPDE simulator; predicting the parameters of the NSPDE model using

GP regressions fitted to the collected features; ABC simulations with summary statistics constructed using the GP regressions. The latter ABC step allows us to expand the analysis outside the initial design on which GP regressions were trained. We report the results in Section 4, separately for GP regressions and for ABC. Section 5 concludes with an outline of further research.

2. Biophysical model of cell migration

Cell migration due to chemotaxis can be modelled with a system of nonlinear stochastic partial differential equations (NSPDEs) describing the behaviour of migrating cells using a pseudopod-centred mechanism, see Neilson et al. [2] and Tweedy et al. [3]. The boundary of a cell is assumed to move in response to an auto-activating internal process, disturbed by external chemoattractants. We adhere to the system defined and discussed by Neilson et al. [2] and adopt the implementation following that of Tweedy et al. [3]. In particular, we use the NSPDE simulator detailed by Tweedy [17]. The most important notation conventions used in this and the following sections can be found in Nomenclature section at the end of the present paper.

2.1. NSPDE system and simulator

The NSPDE system of Tweedy et al. [3] models cell shape behaviour by a time-evolving contour $\Gamma(t)$ responding to three biochemical species: local activator $a(\gamma, t)$ (LA), local inhibitor $b(\gamma, t)$ (LI) and global inhibitor $c(t)$ (GI). The former two are local in nature, i.e. they are defined for each point $\gamma \in \{\mathbb{R} | 0 < \gamma < |\Gamma(t)|\}$ around the contour for each time point t , where $|\Gamma|$ denotes the total length of the boundary. The evolution of the three species is specified as follows

$$\frac{\partial a(\gamma, t)}{\partial t} = D_a \frac{\partial^2 a(\gamma, t)}{\partial \gamma^2} + \frac{s(\gamma, t) \left(\frac{a(\gamma, t)^2}{c(t)} + b_a \right)}{(k_M + b(\gamma, t))(1 + s_a a(\gamma, t)^2)} - d_a a(\gamma, t), \quad (1)$$

$$\frac{\partial b(\gamma, t)}{\partial t} = D_b \frac{\partial^2 b(\gamma, t)}{\partial \gamma^2} + k_b a(\gamma, t) - d_b b(\gamma, t), \quad (2)$$

$$\frac{\partial c(t)}{\partial t} = \frac{r_c}{|\Gamma|} \oint_{\Gamma} a(\gamma, t) d\gamma - r_c c(t), \quad (3)$$

$$s(\gamma, t) = (1 + d_r R) + \left(\frac{C(\gamma, t)}{C(\gamma, t) + k_d} \right) (1 + d_r R), \quad (4)$$

$$\frac{\partial \Gamma(t)}{\partial t} = (f_a a(\gamma, t) - \lambda(A(t) - A_0)) \hat{\Gamma}(\gamma, t), \quad (5)$$

where $s(\gamma, t)$ in (4) represents the strength of the chemosensation stimulus around the point γ and time t in terms of external concentration $C(\gamma, t)$, while (5) defines the evolution of the cell boundary $\Gamma(t)$. R in (4) is a random noise term, uniformly drawn from $[0, 1]$, introduced to break the symmetry of the system to allow for pseudopod formation. In (5), $\hat{\Gamma}(\gamma, t)$ denotes the unit vector that is outward normal to the boundary at point γ , $A(t)$ is the cell area at time

t and A_0 is the so-called “natural area” of the cell, see Tweedy et al. [3]. The remaining quantities ($f_a, r_c, k_b, d_b, D_b, k_M, s_a, b_a, D_a, d_a, d_r, \lambda, k_d$) are parameters of the model; we present most of them in Table 1 and refer to Neilson et al. [2] and Tweedy et al. [3] for the explanation of the remaining ones.

The details of the stochastic system (1)–(5) are presented in Tweedy et al. [3]. In essence, the equations of the Meinhardt [18] model describing an auto-activating/autonomous formation and modulation of pseudopods in response to an external chemotactic gradient are coupled with an evolving cell boundary. The Meinhardt [18] model uses three reaction-diffusion equations, (1), (2) and (3), to describe an autocatalytic and saturating LA, a destabilising LI locally absorbing peaks of LA, and a rapidly equilibrating GI to allow for a “competition” between all membrane points, respectively. Hence, the model adds LI to the standard local-excitation, global inhibition model [19] to destabilise dominant regions of activation.

The Meinhardt [18] model considers a static cell perimeter and does not allow for cell movement. Neilson et al. [2] introduce the cell surface evolving according to (5) to the Meinhardt [18] model. To this end, the evolving surface finite element method of Dziuk and Elliott [20] is applied under the assumption that each point on the cell membrane moves outwards in the direction specified by the normal to boundary at that point [21], with the LA concentration specifying the protrusion rate. To maintain an approximately fixed cell area over time, the outward movements are counteracted by retractions based on the local curvature of the cell boundary.

The random noise $s(\gamma, t)$ is generated from a stochastic process given by the stochastic equation in (4). The first summand on the right hand side of (4) represents a noisy autocatalytic activation while the second summand relates to noisy chemotactic signal. The derivation of (4) is provided in Neilson et al. [22], who argue that this formulation is biologically more realistic than the one originally used by Meinhardt [18] (requiring localising the highest chemoattractant concentration and based on a cosine function to model the behaviour of the chemical signal). The reason why a stochastic component is necessary in the model is that it introduces perturbations and instabilities into the system required to break its symmetry. The asymmetries are the driving forces of pseudopod formation. We note that deterministic systems can provide limits for large stochastic systems via averaging effects, here, however, we are concerned with a single cell model so those effects are not available.

The three reaction-diffusion equations of this model (1)–(3) describe dynamic pattern formation, where the concentration of the local activator is proportional to the outward force acting on the membrane, while the retraction force is provided by membrane tension.

The system (1)–(5) does not admit an analytical solution and hence needs to be solved numerically. Tweedy et al. [3] employ the finite element method (FEM), in which the NSPDE space is discretised using a finite element mesh and the NSPDEs are numerically integrated over the resulting discretisation. In particular, in the approach adopted by Tweedy et al. [3], the finite element points are independently uniformly drawn from the membrane in each time

period, so the recorded samples do not represent the same points in time.

Our aim is to infer the following 10 parameters of the system

$$\boldsymbol{\theta} = (f_a, r_c, k_b, d_b, D_b, k_M, s_a, b_a, D_a, d_a)^T,$$

in which we follow Tweedy [17]. The remaining parameters of the system are treated as fixed in the NSPDE simulator. We present parameter meanings and their default values [from 17] in Table 1. By default, the NSPDE simulator describes a simulation of duration of 100000 arbitrary time units u and outputs samples every $100u$ (evenly spaced), which results in the output of time dimension $T = 1000$.

Table 1: Parameters of the system (1)–(5) to be inferred for *Fully Observed Data* and *Partially Observed Data* as specified by Tweedy [17] together with their boundaries we assumed for preparing the calibration samples, (see Section 2.2, the last paragraph, for a discussion of the challenge in specifying the boundaries and Section 4.1 for a discussion of the chosen boundaries).

Parameter	Meaning	Default value $\tilde{\theta}$	Design boundaries	
			Lower	Upper
f_a	rate of the outward force from LA	0.0015	0.0008	0.003
r_c	response speed of GI	0.07	0.035	0.14
k_b	birth rate of LI	0.0028	0.0014	0.0056
d_b	death rate of LI	0.013	0.0065	0.020
D_b	diffusivity of LI	0.045	0.0225	0.09
k_M	Michaelis-Menten constant for LA	0.16	0.08	0.32
s_a	saturation of autocatalysis of LA	7.0E-5	3.5E-5	1.4E-4
b_a	basal production level of LA	0.1	0.05	0.2
D_a	diffusivity of LA	0.025	0.0125	0.05
d_a	death rate of LA	0.02	0.01	0.04

2.2. Output data and their challenges

The NSPDE simulator generates five outputs: a time series of XY coordinates of finite element nodes representing the cell membrane; a univariate time series of the GI level; and multivariate time series of LA, LI, S values over the membrane. Figure 1 illustrates outputs from the NSPDE simulator generated using the default parameter values. Below we discuss the challenges encountered when working with the NSPDE simulator and its outputs.

Stochastic system. The system in question is stochastic, which means that each time we run the NSPDE simulator, a different realisation is obtained, even for the same $\boldsymbol{\theta}$. Figure 2 illustrates this property by presenting two outputs obtained with the default parameter setting, with darker contours being recorded later in time. One can see that qualitatively both runs are very similar (in terms of the cell shape and cell movement over time), despite different “nominal”, or actual, locations of the membrane (e.g. in Simulation 1, the cell moves from around 300 to 500 along the x-axis, while in Simulation 2 it moves from around 250 to 400 in the x-axis). A standard approach to deal with stochastic problems is provided

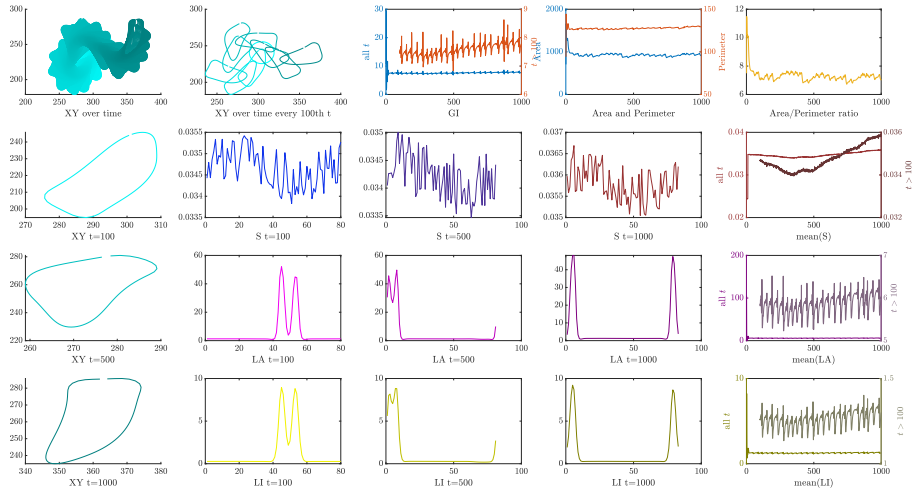


Figure 1: Outputs generated using the default parameter values $\bar{\theta}$. From left to right, top row: cell membrane evolution in time for $t = 1, 2, \dots, 1000$ (the shade indicates the time point, from early – light, to late – dark); cell membrane evolution in time for $t = 1, 101, \dots, 901$; time series of GI for $t = 1, 2, \dots, 1000$ (left axis) and for $t \geq 100$ (right axis); time series of area (left axis) and perimeter (right axis); time series of area-perimeter ratios. Second row: cell contour at $t = 100$; value of stochastic signal s over the cell membrane at $t = 100$; value of s over the cell membrane at $t = 500$; value of s over the cell membrane at $t = 1000$; time series of means over space of s for $t = 1, 2, \dots, 1000$ (left axis) and for $t \geq 100$ (right axis). Third row: cell contour at $t = 500$; value of LA over the cell membrane at $t = 100$; value of LA over the cell membrane at $t = 500$; value of LA over the cell membrane at $t = 1000$; time series of means over space of LA for $t = 1, 2, \dots, 1000$ (left axis) and for $t \geq 100$ (right axis). Bottom row: cell contour at $t = 1000$; value of LI over the cell membrane at $t = 100$; value of LI over the cell membrane at $t = 500$; value of LI over the cell membrane at $t = 1000$; time series of means over space of LI for $t = 1, 2, \dots, 1000$ (left axis) and for $t \geq 100$ (right axis).

by likelihood-based inference, however, as discussed in the Introduction, for high-dimensional and complex systems, such as the biophysical cell migration model, likelihood computation becomes infeasible. In consequence, we proceed with likelihood-free inference based on features extracted from the data. The extracted features need to be similar for stochastic outputs generated using the same parameters hence they cannot depend on e.g. “nominal” values of the cell membrane locations. On the other hand, appropriate features need to allow us to differentiate between distinct parameter values.

Varying output dimensions. The FEM implementation of Tweedy et al. [3] allows the number of finite element nodes to vary across simulations and even within a single simulation. These authors add or remove the nodes in consecutive iterations depending on their density on the evolving line. For instance, when the cell front expands, nodes are added to maintain discretisation accuracy, while as the cell rear retracts nodes need to be removed as they become too dense. In roughly 35% of our simulations (constituting our training calibration sample, to be discussed later) the spacial dimension of the generated matrices was constant over time and equal to the initial size of 79 finite element points.

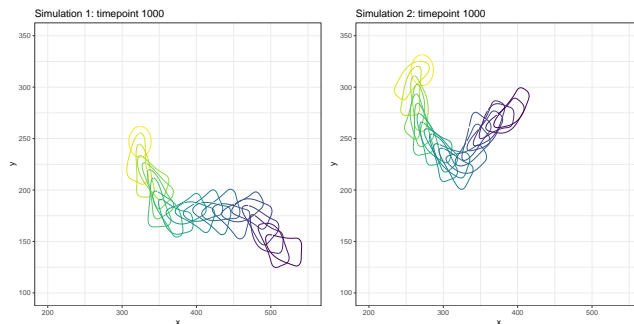


Figure 2: Two different realisations from the NSPDE simulator obtained using the default parameter values θ : cell contours (XY coordinates) plotted every 100th time point.

For most parameters, however, cells were growing (up to 407 points) as shown in Figure 3. In several rare cases we observed shrinking of cells (down to 58 points).

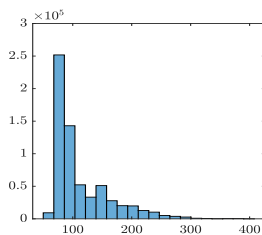


Figure 3: Histogram of spacial dimension sizes (x-axis) in the training calibration sample for the time-varying cases (observed in approximately 65% of simulations).

Time-unaligned nodes. As discussed above, Tweedy et al. [3] solve the NSPDE system (1)–(5) using FEM by independently drawing points from the cell membrane in each time period. In consequence, the samples of membrane points do not represent the same points in time and hence it is not possible to track the evolution of a given point on the membrane to perform time-series analysis. Moreover, since the sampled membrane points cannot be interpreted as physical landmarks on the boundary, the Procrustes method [see 23, for an overview], commonly applied to analyse shapes and their evolution, was impractical in our case. Supplementary Material B illustrates the problems related to applying the Procrustes method in more detail.

Unknown parameter ranges. The NSPDE simulator takes 10 parameters as input, however neither Neilson et al. [2] nor Tweedy et al. [3] explicitly specify biophysically plausible ranges for these parameters. These ranges might be (roughly) clear to researchers working on chemotaxis, however they are unknown to anyone taking the model (1)–(5) at face value. The only information

clearly available *a priori*, before running any experiments using the NSPDE simulator, is the default parameter vector θ as specified by Tweedy [17] (see Table 1).

2.3. Practical considerations

Next to the output challenges discussed in the previous section there are two further points that need to be considered, related to practice of laboratory work. First, modes of operation of experimental scientists often involve experiments and inference taking place at distinct point in time. This means that typically a long time is necessary to set up equipment and experiments as well as to develop a biophysical model. Given time-limited projects, this often leaves researchers with relatively little time for statistical inference, which by definition requires data being available. A framework practical to experimentalists should therefore complete as much of the relevant computations as possible before any data become available, only relying on the theoretical model.

Second, the outputs from the system (1)–(5) differ with respect to their “observability”. Ideally, one would be able to measure all the outputs, not only the evolving cell membrane but also chemical signals, including the chemosensation at membrane $s(\gamma, t)$. This, however, might not be feasible in all cases, due to equipment limitations or physiological properties of the studied environment. Therefore, we propose to consider two cases of data observability which we call *Fully Observed Data* and *Partially Observed Data*. In the former, we assume that the researcher is able to accurately record all the outputs from the system, including the chemical signals. In the latter, we consider a possibly more realistic scenario, in which one can only record the evolving cell boundary. This reflects a common situation in microscopy without access to spectroscopy.

3. GP-enhanced semi-automatic ABC

In this section we present our proposed methodological inference framework, summarised in Figure 4. Our final objective is parameter estimation and uncertainty quantification using ABC-SMC (step 4, discussed in Section 3.3). ABC-SMC is a state-of-the-art computational algorithm for approximate inference based on features extracted from the data. Even though the algorithm itself is general, the initial feature extraction phase (step 2) is typically problem-specific and may be heuristic and reliant on intuition and domain knowledge. In this sense feature extraction is similar to analysing the outputs from NSPDE model simulations (step 1, discussed in Section 2.2), which is model-dependent. In Section 3.1 we discuss how we tackle the challenge of obtaining features from the high-dimensional outputs from the biophysical cell movement model (1)–(5). We note that the devised procedure is quite complex (hence we refer to step 2 in the context as “feature construction” rather than “feature extraction”) and necessarily tied to the biophysical model under investigation, hence might not be directly transferable to other physical models. We therefore recommend skipping Section 3.1 at a first reading. We start discussing our essential

methodological contribution, general and transferable to other physical models, in Section 3.2, where we describe how the problem-specific features from step 2 are best combined and weighted so as to improve the approximate parameter inference and uncertainty quantification in step 3.

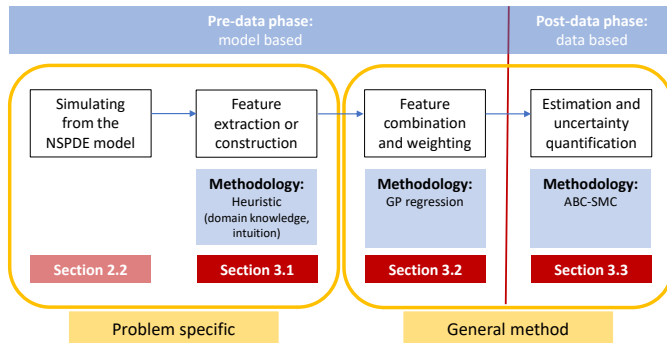


Figure 4: General pipeline of the GP-enhanced semi-automatic ABC framework. Simulating from any NSPDE model and obtaining features from the resulting outputs is typically problem specific. General statistical methodology is then used to combine the extracted features with GP regression for parameter estimation and uncertainty quantification with ABC-SMC. Additionally, we distinguish the pre-data (model-based) phase and the post-data (data-based) phase. The former allows the user to perform considerable part of the analysis and computations prior to observing any data solely based on simulations from the model. In the latter, ABC simulations are run to make inference over a particular dataset of interest.

Figure 5 illustrates our methodological pipeline applied to the particular case of the biophysical NSPDE model (1)–(5), to a high extent motivated by the timeline of a typical experimental project (see Section 2.3). We note, however, that it emerges as a special case of the general procedure in Figure 4 with the last two steps (GP regression and ABC-SMC simulations) being directly transferable to other physical models. Conceptually, feature construction in step 2 is self-evident. Yet given the high-dimensional and stochastic output from the NSPDE simulator (approximately half a million per single forward simulation), efficient feature extraction is a challenging task in this case. Here, by “efficiency” we mean that on the one hand we need to considerably reduce the dimensionality of the outputs to a few dozens of variables feasible for GP regression, while on the other hand still capture the key characteristics of the output data. Therefore, feature construction plays an important role in our application to the biophysical NSPDE model (1)–(5). Another application specific element is using the GP regression to obtain preliminary parameter estimates so that ABC-SMC can be used to refine these results. There are two main advantages of the final ABC correction over using the estimates from the GP regression alone. First, the ABC-SMC is essentially not restricted to the initial compact domain represented by the training calibration sample, which means that extrapolating beyond the calibration sample is more robust. Second, uncertainty quantification with ABC is more flexible as the ABC-posterior distribution is not limited to the Gaussian distribution (light-tailed and symmetric), which is an inherent consequence of

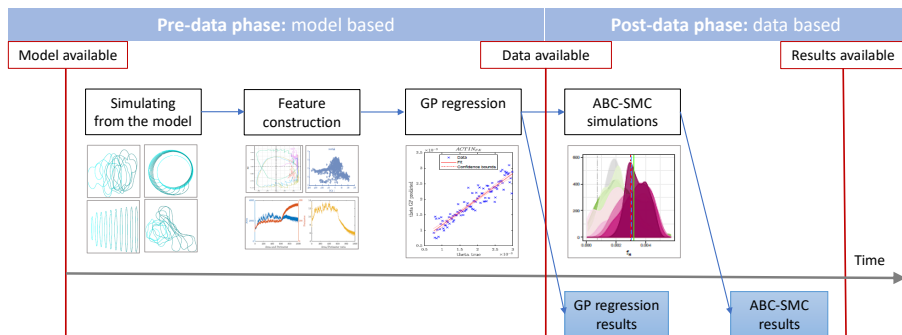


Figure 5: Pipeline of the GP-enhanced semi-automatic ABC framework in the context of the biophysical cell movement model (1)–(5). The model-based phase allows for performing a considerable part of the analysis prior to observing any data by just using the model. In this case, the data-based phase also uses the fitted GP regressions to infer the parameters for the given dataset. Importantly, all the ABC simulations can only be carried out with the data available.

the chosen form of the regression model (6).

3.1. Feature extraction

As mentioned above, we recommend skipping the details in this section at a first reading. In general, there are three main requirements which the extracted features need to satisfy in order for us to consider them “appropriate”. First, the features should capture changes in the output as parameters change. Second, the features should be similar for stochastic outputs generated using the same parameters. Third, for features describing shapes, they should be invariant with respect to scale, rotation and translation. In addition to these requirements, we need to distinguish between cell-contour-based features, which could be used for both *Fully Observed Data* and *Partially Observed Data*, and chemical-signals-based features, which we could use only for *Fully Observed Data*.

3.1.1. Fourier-transform-based features

Cell shapes generated from the NSPDE simulator are highly variable, both for a single parameter set as well as across different parameter configurations, see Figure 11 and Supplementary Material A. Moreover, the lack of alignment of the finite element nodes over time makes the nominal, absolute location values (e.g. the mean value of the XY coordinates or the location of the peak of the LA over the membrane) mostly meaningless. Therefore, it is necessary to adopt alternative measures to appropriately summarise the observed patterns. Such a means is provided by Fourier analysis, which is suited to capture the properties of the shape only, irrespective of its scale, translation or rotation. For illustration, consider the two cell shapes in Figure 6a: they should lead to exactly the same features because one cell is just a scaled, rotated and shifted version of the other. The same holds for the two LA signals in Figure 6b which differ only with respect to the location of the signal peak. We note that Tweedy et al. [3] employed Fourier shape descriptors to embed extremely variable cell shapes into a low dimensional manifold.

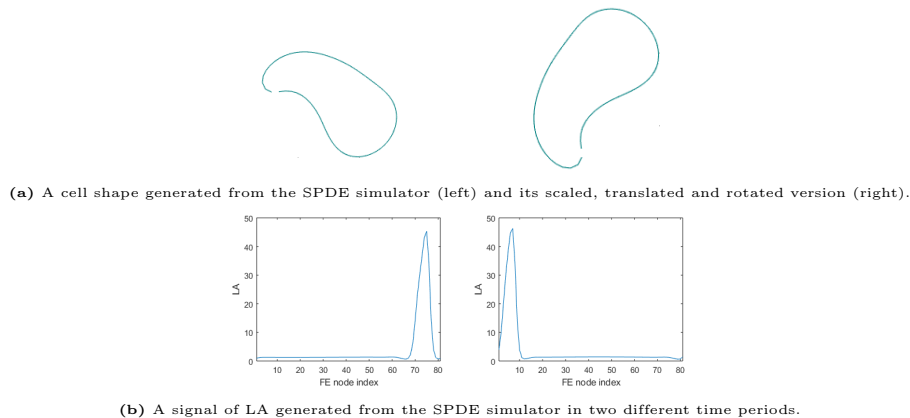


Figure 6: Shapes originating from a cell contour (top) and a chemical signal (bottom). For each pair the corresponding power spectra from Fourier analysis are the same.

Fourier-transformed chemical signals. In this subsection we focus on the Fourier analysis applied to chemical signals, leaving the discussion of the features obtained using Fourier descriptors applied to the cell contours to the next subsection. Consider Figure 7 which illustrates a particular type of multivariate chemical signal¹ $\alpha^{(j)}$ e.g. LA, LI or S (blue figures) for different parameter vectors $\theta^{(j)}$, $j = 1, \dots, M$. As mentioned in Section 2, all simulations are of time dimension $T = 1000$ but their spacial dimensions may vary (both across θ s and within a single simulation for a given θ). We denote the length of the spacial profile of the signal $\alpha_t^{(j)}$ at time t by $D_t^{(j)}$.

We start with computing $\mathbf{\Pi}_t^{(j)}$, the 2-sided power spectrum of $\alpha_t^{(j)}$, which is also of length $D_t^{(j)}$, as $\mathbf{\Pi}_t^{(j)} = |\text{fft}(\alpha_t^{(j)})|^2$ (where fft is the fast Fourier transform). Since we consider real signals only, we focus on 1-sided power spectra, in which we exclude the 0-period entry $\Pi_{t,1}^{(j)}$ (the first element of the $\mathbf{\Pi}_t^{(j)}$ vector) corresponding to the square of the sum of the signal. Thus, we take $\boldsymbol{\pi}_t^{(j)} = \mathbf{\Pi}_{t,2:\tau_t^{(j)}}^{(j)}$, where $\tau_t^{(j)}$ is the middle of the 2-sided power spectrum computed as $\tau_t^{(j)} = D_t^{(j)}/2 + 1$ for $D_t^{(j)}$ -even and $\tau_t^{(j)} = (D_t^{(j)} + 1)/2$ for $D_t^{(j)}$ -odd. We set $\tau^* = \min_{j,t} \{\tau_t^{(j)}\}$, the length of the shortest 1-sided power spectrum in the sample. For our calibration sample $\tau^* = 30$, which corresponds to the cells with the smallest spacial dimension of 58. Finding τ^* is important as it gives us the maximum common (over outputs for θ s in the calibration sample) dimension of 1-sided power spectra. This maximum common dimension is necessary as our ultimate goal is to obtain a matrix Ψ , to which we want to apply PCA,

¹We can also specify $\alpha_t^{(j)}$ differently e.g. as approximate time differences of the raw signals of LA, LI or S. This means computing for each time point and for each finite element node an approximate time derivative of a given signal. Approximation is due to using discrete series but most importantly due to the lack of alignment of finite element nodes over time.

with $\psi^{(j)}$, the i th row of Ψ , corresponding to $\theta^{(j)}$, $j = 1, \dots, M$. Hence, what we extract from different $\alpha^{(j)}$ s needs to be of the same size, even though $\alpha^{(j)}$ s themselves have a varying spacial dimension.

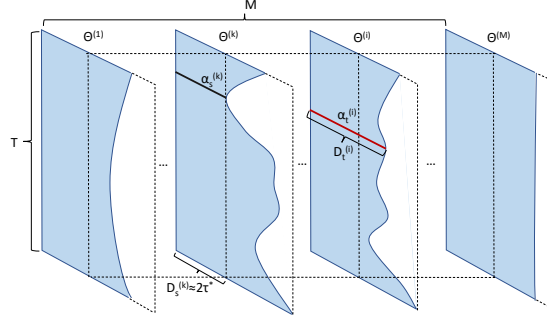


Figure 7: Finding τ^* , the maximum common (over parameter set $\{\theta^{(j)}\}$) dimension of 1-sided power spectra. Blue shaded shapes illustrate single multivariate outputs (e.g. S, LA or LI) for different $\theta^{(j)}$, $j = 1, \dots, M$. All outputs have the common time dimension of $T = 1000$ but varying space dimension. The signal under consideration $\alpha_t^{(i)}$ (red) is of dimension $D_t^{(i)}$, which might be higher than $D_s^{(k)}$, the one of $\alpha_s^{(k)}$ (black), the shortest signal in $\{\theta^{(j)}\}$ (projected to other signal with dashed lines). The 2-sided power spectrum of $\alpha_s^{(k)}$ is of length $D_s^{(k)} \approx 2\tau^*$. Since the signals are real, their 2-sided power spectra are symmetric, so it is sufficient to consider 1-sided power spectra. Hence, for all the signals of different length we extract *subsets* of their 1-sided power spectra of the maximum common length $\tau^* - 1$ (subtracting one as we exclude the 0-period entries).

Given a 1-sided power spectrum $\pi_t^{(j)}$, we want to extract its highest magnitudes with the corresponding frequencies. We include the frequencies in the analysis as they carry information about the shape of the spacial profile $\alpha_t^{(j)}$. To this end, we sort $\pi_t^{(j)}$ in descending order to obtain $\tilde{\pi}_t^{(j)}$ and sort the normalised frequencies $\varphi = \left[\frac{1}{\tau_t^{(j)}}, \dots, \frac{\tau_t^{(j)} - 1}{\tau_t^{(j)}} \right]$ using the same permutation to get $\tilde{\varphi}$. We consider only the first $(\tau^* - 1)$ values of both sorted sequences and express the magnitudes in $10 \log_{10}$ terms. The final part of the input to the matrix for PCA corresponding to $\alpha_t^{(j)}$ is given by

$$\tilde{\psi}_t^{(j)} = \left[10 \log_{10}(\tilde{\pi}_{t,1}^{(j)}), \tilde{\varphi}_1, 10 \log_{10}(\tilde{\pi}_{t,2}^{(j)}), \tilde{\varphi}_2, \dots, 10 \log_{10}(\tilde{\pi}_{t,\tau^*-1}^{(j)}), \tilde{\varphi}_{\tau^*-1} \right].$$

Concatenating $\tilde{\psi}_t^{(j)}$ over time

$$\psi^{(j)} = \left[\tilde{\psi}_1^{(j)}, \tilde{\psi}_2^{(j)}, \dots, \tilde{\psi}_T^{(j)} \right]$$

gives us the i th row of the Ψ matrix. Figure 8 illustrates the subsequent transformations of the original signal $\alpha_t^{(j)}$ and Algorithm 1 summarises the steps of our procedure.

A similar procedure can be used for univariate series such as GI or univariate series extracted from multivariate series (e.g. time series of means taken at each time point over spacial profiles). Since in this case there is no problem of

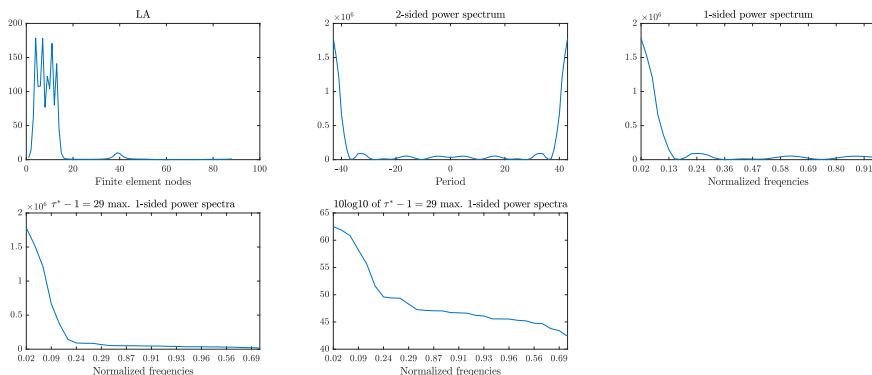


Figure 8: Construction of an input to PCA based on the Fourier transformation. Left-right, top-bottom: $\alpha_t^{(j)}$ – the original signal of LA at t ; $\Pi_t^{(j)}$ – the 2-sided power spectrum of $\alpha_t^{(j)}$; $\pi_t^{(j)}$ – the 1-sided power spectrum of $\alpha_t^{(j)}$; $\tilde{\pi}_{t,1;\tau^*-1}^{(j)}$ – the $\tau^* - 1 = 29$ highest magnitudes of 1-sided power spectrum; $10 \log_{10}(\tilde{\pi}_{t,1;\tau^*-1}^{(j)})$ – $10 \log_{10}$ of the highest magnitudes.

changing spacial dimensions (we deal with a univariate series of length $T = 1000$) the procedure becomes much simpler. We can consider the whole sorted 1-sided power spectrum (excluding the 0-period), without the need for extracting a part of the maximum common length. Formally, for a univariate signal $\alpha^{(j)}$ we compute $\Pi^{(j)} = \text{fft}(|\alpha^{(j)}|)^2$, set $\pi^{(j)} = \Pi_{2:501}^{(j)}$ and $\varphi = \frac{1}{501} [1, \dots, 500]$ and sort $\pi^{(j)}$ to get $\tilde{\pi}^{(j)}$ (sort φ using the same permutation to obtain $\tilde{\varphi}$). We then consider $\psi^{(j)} = \left[10 \log_{10}(\tilde{\pi}_1^{(j)}), \tilde{\varphi}_1, \dots, 10 \log_{10}(\tilde{\pi}_{500}^{(j)}), \tilde{\varphi}_{500} \right]$.

Fourier-transformed cell contours. To extract relevant information about the variation of cell shapes (XY coordinates) in time we also use the Fourier transform, however in a different way than for the chemical signals. Before presenting the details of our approach we note that we also attempted to apply generalised Procrustes analysis [24], a very popular method in landmark-based morphometrics scenarios [25], to summarise a cell membrane evolving in time. However, this approach was not successful as highly variable cell contours have no recurring characteristics which could be used as landmarks, i.e. a discrete set of points that are homologous (based on mathematical, physical or biological insights) between cells and different time points. We illustrate the problems encountered in applying generalized Procrustes analysis in Supplementary Material B, Figure B.1.

Elliptical Fourier analysis (EFA) provides a viable alternative to landmark-based methods for shape analysis. This method, together with eigenshape analysis, belongs to the class of outline analysis methods, which do not compare homologous points, but rather focus on estimating the coefficients of various functions fitted to points sampled around the outlines of objects. The basic idea of Fourier analysis is to decompose a periodic signal into a sum of sine and cosine functions, this can easily be applied to shape analysis since a closed outline can be regarded as periodic function [26]. EFA reconstructs the object's

Algorithm 1: Obtaining rows of the matrix Ψ for PCA for multivariate chemical signals using Fourier transform.

Input : index j corresponding to a parameter vector $\theta^{(j)}$,
 $j = 1, \dots, M$; multivariate chemical signal $\alpha^{(j)}$ (LA, LI or S);
time dimension T of α ; the length of the shortest 1-sided
power spectrum $\tau^* = \min_{j,t} \{\tau_t^{(j)}\}$ for the sample $\{\theta^{(j)}\}$.

- 1 **for** $t = 1$ **to** T **do**
- 2 Find $D_t^{(j)}$, the effective size of the signal $\alpha_t^{(j)}$ (by possibly
eliminating its last entries equal to zero).
- 3 Compute the 2-sided power spectrum: $\mathbf{\Pi}_t^{(j)} = |\text{fft}(\alpha_t^{(j)})|^2$ (where fft
is the fast Fourier transform).
- 4 Take the 1-sided part (since $\alpha_t^{(j)}$ is real): $\pi_t^{(j)} = \mathbf{\Pi}_{t,2:\tau_t^{(j)}}^{(j)}$, where
 $\tau_t^{(j)} = D_t^{(j)}/2 + 1$ for $D_t^{(j)}$ -even and $\tau_t^{(j)} = (D_t^{(j)} + 1)/2$ for
 $D_t^{(j)}$ -odd (discard the 0-period entry as it is just the square of the
sum of the signal).
- 5 Normalise frequencies: $\varphi = \left[\frac{1}{\tau_t^{(j)}}, \dots, \frac{\tau_t^{(j)} - 1}{\tau_t^{(j)}} \right]$.
- 6 To consider the highest magnitudes (with the corresponding
frequencies): sort $\pi_t^{(j)}$ in descending order to get $\tilde{\pi}_t^{(j)}$ (use the
same permutation to obtain $\tilde{\varphi}$).
- 7 Express the magnitudes in $10 \log_{10}$ terms.
- 8 Set $\tilde{\psi}_t^{(j)} =$
 $\left[10 \log_{10}(\tilde{\pi}_{t,1}^{(j)}), \tilde{\varphi}_1, 10 \log_{10}(\tilde{\pi}_{t,2}^{(j)}), \tilde{\varphi}_2, \dots, 10 \log_{10}(\tilde{\pi}_{t,\tau^*-1}^{(j)}), \tilde{\varphi}_{\tau^*-1} \right]$.
- 9 **end**
- 10 Set $\psi^{(j)} = \left[\tilde{\psi}_1^{(j)}, \tilde{\psi}_2^{(j)}, \dots, \tilde{\psi}_T^{(j)} \right]$ to obtain the i th row of Ψ .

outline as a finite sum of ellipses, which is the plane representation of a finite sum of a sine and cosine curve (also termed a “harmonic”).

In EFA, the outline of a shape is described through a set of points, not necessarily equally-spaced, sampled on the outline and their corresponding x and y coordinates. Since we are interested in the cell shape, the outline is described in terms of the finite element nodes and their corresponding x and y coordinates. Let P denote the perimeter of a closed outline (which is also the period of the signal) and define $\omega = 2\pi/P$. We can write the Fourier decompositions of the incremental changes of the x and y coordinates as a function of the cumulative length along the shape’s outline, p , which varies from 0 to P ,

$$x(p) = \frac{a_0}{2} + \sum_{n=1}^k [a_n \cos(n\omega p) + b_n \sin(n\omega p)],$$

$$y(p) = \frac{c_0}{2} + \sum_{n=1}^k [c_n \cos(n\omega p) + d_n \sin(n\omega p)],$$

where k denotes the number of finite number of points sampled on the shape’s outline and a_0 and c_0 are the coordinates of the centroid of the original outline. As such, each harmonic corresponds to four coefficients, with the two coefficients for the x projection in the n th harmonic given by the following formulae (c_n and d_n for the y projection have analogous expressions)

$$a_n = \frac{P}{2\pi^2 n^2} \sum_{j=1}^k \frac{\Delta x_j}{\Delta p_j} [\cos(n\omega p_j) - \cos(n\omega p_{j-1})],$$

$$b_n = \frac{P}{2\pi^2 n^2} \sum_{j=1}^k \frac{\Delta x_j}{\Delta p_j} [\sin(n\omega p_j) - \sin(n\omega p_{j-1})],$$

where p_j is the cumulative length along the shape’s outline until the j th finite element point has been reached. The number of harmonics included determines how good the approximation of the shape is, with the lower harmonics providing an approximation for the coarse-scale trends in the outline, while the high-frequency harmonics describe the finer-scale details (see Figure 9 for visualisation of the harmonic contribution to a cell shape).

3.1.2. Anchoring features

The Fourier descriptors discussed above are important to capture qualitative properties of signals, both chemical and cell-contour-based. However, some absolute “anchoring” measures are necessary to correctly infer parameters responsible for the size or speed of cells. In this sense they should provide complimentary information to the features based on the Fourier transform. An important insight obtained from the exploration of outputs from our calibration sample is that for some parameter configurations cells develop pseudopods, often very dynamically, while for other parameters cells remain in a simple, elongated shape. The initial cell shape for all the parameter was a circle, and depending on θ the

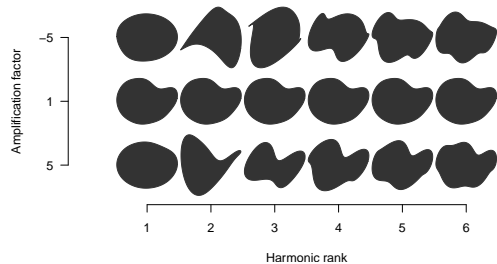


Figure 9: Harmonic contribution to a cell shape: the effect of every harmonic (x axis) on the cell shape reconstruction. Each corresponding harmonic’s coefficient is multiplied by an amplification factor (y -axis) to either illustrate the standard shape (amplification factor 1) or to exaggerate the coefficient’s effect (amplification factor 5 or -5). The level of detail in the reconstruction increases with the harmonic rank.

cell can diverge from this initial state, changing the proportions between its area and perimeter (area-perimeter ratio, APR). Moreover, the elongated cases are likely to cover long distances in a directed manner while the pseudopod-forming cases tend to either stay still or randomly wander around their initial location. These observations are in line with the findings of Neilson et al. [2].

Figure 10 shows various types of dynamics of cell areas, perimeters and APRs that can be observed for different cell types. For the elongated rigid type of cell (top row) all three measures increase smoothly over time, while for the circular rigid type (bottom row) they smoothly remain (roughly) constant over time. For the two flexible types (2nd and 3rd row) these three quantities are subject to sharp fluctuations over time, with the elongated type showing a downward trend in the later time periods and the circular type exhibiting a stable dynamics.

These observations lead us to specifying “anchoring” features as follows. First, we consider the mean value of APR for $t \geq 500$. Restricting the time domain to its second half should mitigate the impact of the convergence of the NSPDE system from its initial state to the stationary state implied by θ and capture the long-term value of the APR. Second, we take the label of the model best fitting the APR time series obtained as follows. We use linear/nonlinear least squares (implemented in the MATLAB `fit` function) to fit 15 different models (polynomials of order 2–9, exponentials of order 1–2, Fourier series of order 1–5) to the whole series of APR as this time we are interested in both the transition period and the steady state. Thus, we can group the cells depending on their APR series. Third, we compute the minimum, mean and maximum distance travelled by the cell to distinguish between stationary and moving cells. We also include the distance to the furthest point reached by the cell’s midpoint, which, in addition to the total distance travelled, should give an indication of whether the cell’s movement is directed or random. Lastly, we include a series of crude measures of shape description and evolution of shape in time based on the radius of the cell, calculated as the distance between the centroid of the cell and points on the cell’s outline. We include the minimum and maximum radius of the cell, taken across all the time points, as well as seven percentiles (1st,

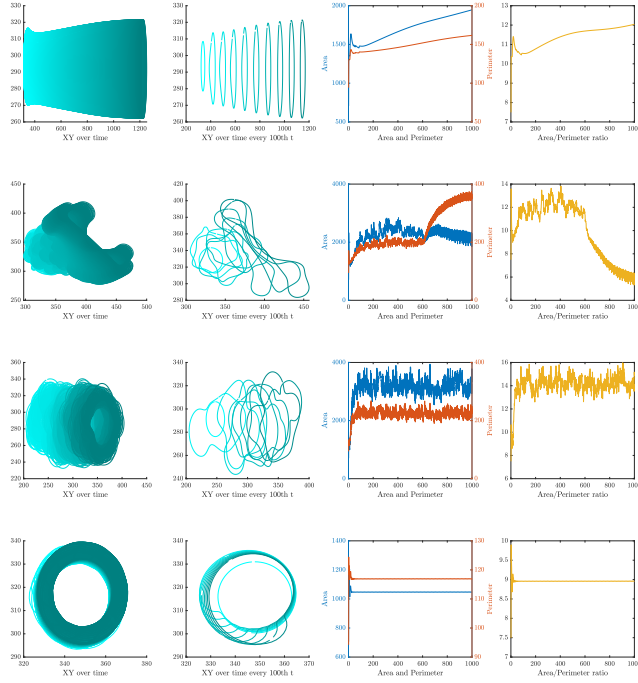


Figure 10: Different cell types and the dynamics of the corresponding areas, perimeters and area-perimeter ratios. Columns from left to right: contour evolution in time for $t = 1, 2, \dots, 1000$ (the shade indicates the time point, from early – light, to late – dark); contour evolution in time for $t = 1, 101, \dots, 901$; time series of GI for $t = 1, 2, \dots, 1000$ (left axis) and for $t \geq 100$ (right axis); time series of area (left axis) and perimeter (right axis); time series of area-perimeter ratios. Rows from top to bottom: elongated rigid type; elongated flexible type, circular flexible type; circular rigid type.

5th, 25th, 50th, 75th, 95th, 99th) of the distribution of the time series of ratios of minimum radius to maximum radius. The latter feature aims to encode some of the information relating to the cell elongation through time.

3.2. Gaussian process regression

To summarise multiple features and to make predictions about θ we employ Gaussian process (GP) regression, see Rasmussen and Williams [14] for an extensive treatment of GPs. The GP regression model is a nonparametric regression explaining the observed target values (here: each element of θ) in terms of the values of the latent function f at the corresponding locations (here: the extracted features, see Section 3.1 and Table 2). These unobserved function values are assumed to follow a GP, which is a stochastic process such that the joint distribution of any finite number of random variables from this process is Gaussian. A Gaussian process is completely specified by its mean and covariance function.

We model each parameter with a separate GP regression, which is similar to the approach of Fearnhead and Prangle [13] who use independent linear regressions for each model parameter separately. We note that it could be possible to model all the parameters jointly using the multivariate output GPs of Conti et al. [27] and Conti and O’Hagan [28], or the GP autoregressive (GPAR) regression model of Requeima et al. [29]. However, the former approach is designed particularly for sparse data for which the correction for correlations between the outputs can be useful. In our large dataset case such a correction is not expected to add much leverage as suggested in Conti and O’Hagan [28, Section 3.1]. Moreover, even though Conti et al. [27] allow for output correlations, they use the same input length scales in the GP kernel. As we point out later in this section (Section 3.2.2), allowing for different input length scales plays an important role in our methodology. The latter GPAR approach, although conceptually appealing, is not practical in our context due to the necessity of selecting an appropriate conditioning structure. More precisely, Requeima et al. [29] suggest to decompose the posterior for P outputs $\boldsymbol{\theta} = (\theta_1, \dots, \theta_P)^T$ as

$$p(\boldsymbol{\theta}(\mathbf{x})) = p(\theta_1(\mathbf{x}))p(\theta_2(\mathbf{x})|\theta_1(\mathbf{x})) \dots p(\theta_P(\mathbf{x})|\theta_1(\mathbf{x}), \dots, \theta_{P-1}(\mathbf{x})),$$

which requires selecting an ordering of the outputs. Since there is no natural ordering in our case, greedy optimisation with $P(P+1)/2$ configurations or exhaustive optimisation with $P!$ configurations would be necessary. Either way, applying GPAR to our case would require onerous computations which we believe should rather be carried out to extend the initial design (see Section 3.2.1 for a discussion how to spend a given computing budget). Finally, we note that GP regressions are only an intermediate step of our methodology, which involves an additional post-correction with ABC. Therefore, any inter-dependencies between the parameters potentially missed by the GP regression are expected to be recovered in the final step of our framework.

The GP regression is useful on its own merits, as it can flexibly capture the relationship between the feature space and parameter space, however its ultimate purpose in our framework is to provide summary statistics for ABC as presented in Section 3.3. For completeness, below we provide a brief summary of the GP regression framework.

Suppose we have M sets of outputs from the NSPDE simulator $\mathbf{y}^{(j)}$ obtained for different parameter vectors $\boldsymbol{\theta}^{(j)}$, $j = 1, \dots, M$. From each output $\mathbf{y}^{(j)}$ we extract features $\mathbf{x}^{(j)} \in \mathbb{R}^D$, which we collect in a matrix $\mathbf{X} = [\mathbf{x}^{(j)}]_{j=1, \dots, M}$. We let $\boldsymbol{\Theta}_i$ denote the vector of the i th NSPDE parameters collected over the calibration sample, i.e. $\boldsymbol{\Theta}_i = [\theta_i^{(1)}, \theta_i^{(2)}, \dots, \theta_i^{(M)}]^T$. We use \mathbf{X} as explanatory variables to explain each of the 10 NSPDE parameters θ_i by means of a GP regression model. For the i th NSPDE parameter the GP regression is given by

$$\theta_i^{(j)} = f_i(\mathbf{x}^{(j)}) + \varepsilon_i^{(j)}, \quad \varepsilon_i^{(j)} \stackrel{iid}{\sim} \mathcal{N}(0, \sigma_i^2), \quad j = 1, \dots, M, \quad (6)$$

where $\mathcal{N}(\mu, \sigma^2)$ denotes the Gaussian distribution with the mean μ and variance σ^2 (and *iid* stands for “independently and identically distributed”). The latent

values $f_i(\mathbf{x})$ follow a GP

$$f_i(\mathbf{x}) \sim \mathcal{GP}(m_i(\mathbf{x}), k_i(\mathbf{x}, \mathbf{x}')),$$

where

$$m_i(\mathbf{x}) = \mathbb{E}[f_i(\mathbf{x})], \quad k_i(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f_i(\mathbf{x}) - m_i(\mathbf{x}))(f_i(\mathbf{x}') - m_i(\mathbf{x}'))]$$

are the mean and the covariance function (kernel) of the process f_i , respectively. Hence, it is assumed that the responses θ_i are conditionally independent given the latent values $f_i(\mathbf{x})$. We stack the latent values for the i th regression in a vector $\mathbf{f}_i = [f_i(\mathbf{x}^{(1)}), f_i(\mathbf{x}^{(2)}), \dots, f_i(\mathbf{x}^{(M)})]^T$. Below we adopt a standard assumption that $m_i(\mathbf{x}) = 0$ so the latent process $f_i(\mathbf{x})$ is fully specified by the kernel function. The kernel can be specified in many different ways, including numerous standard functional forms, see Rasmussen and Williams [14, Ch. 4], and typically deciding which kernel is “the best” (in any sense) is not obvious without performing sensitivity analysis. We return to this issue in Section 3.2.2 and for now we only assume that the GP regression model in (6) corresponding to the chosen kernel is parametrised by a vector of hyperparameters ϕ_i consisting of the observation noise variance, σ_i^2 , and the kernel hyperparameters².

For the collected covariates \mathbf{X} we obtain the GP prior over function values $p(\mathbf{f}_i|\mathbf{X}, \phi_i) = \mathcal{N}(\mathbf{f}_i|\mathbf{0}, \mathbf{K}_i)$, with $\mathbf{K}_i = k_i(\mathbf{X}, \mathbf{X})$. The likelihood is given by $p(\Theta_i|\mathbf{f}_i) = \mathcal{N}(\Theta_i|\mathbf{f}_i, \sigma_i^2\mathbb{I})$, where \mathbb{I} is the identity matrix, and marginalising over the latent variables gives the formula for the marginal likelihood

$$p(\Theta_i) = \mathcal{N}(\Theta_i|\mathbf{0}, \mathbf{K}_i + \sigma_i^2\mathbb{I}).$$

Since the observation model is Gaussian, the conditional posterior distribution of the latent variables is also Gaussian and has the form

$$p(\mathbf{f}_i|\Theta_i, \mathbf{X}, \phi_i) = \mathcal{N}(\mathbf{K}_i(\mathbf{K}_i + \sigma_i^2\mathbb{I})^{-1}\Theta_i, \mathbf{K}_i - \mathbf{K}_i(\mathbf{K}_i + \sigma_i^2\mathbb{I})^{-1}\mathbf{K}_i).$$

3.2.1. Estimation

The GP regression model for the i th NSPDE parameter, $i = 1, \dots, 10$, is parametrised by the hyperparameter vector ϕ_i , which be estimated using maximum likelihood (ML) or in the Bayesian way. The latter framework is conceptually appealing as it allows for incorporating prior knowledge into the analysis and provides a full description of parameter uncertainty. However, these advantages come at the price of a time consuming estimation and more involved predictions from the fitted model. The increased computational burden of Bayesian hyperparameter estimation can be problematic for a thorough sensitivity analysis, aimed at selecting the best performing kernel. Since any estimation is done on a fixed calibration sample, any extra time can be spent on either a more robust estimation on this existing sample or on extending the existing design and

²In this we follow Rasmussen and Williams [14, Ch. 5] who consider σ_i^2 a hyperparameter.

thus improving the accuracy of the GP regression (forward simulations from the model (1)–(5) take around two minutes on a single CPU). Thus using quick-to-obtain ML hyperparameter estimates allows us to free up the computational resources which can be spend on extra forward simulation from the NSPDE model to obtain a larger calibration sample (denser initial design).

3.2.2. Kernel selection and relevance analysis

We consider five GP kernel types: squared exponential (se), Matérn 3/2 (m32), Matérn 5/2 (m52), rational quadratic (rq), neural network (nn), see Rasmussen and Williams [14, Ch. 4], which are standard in the literature. We first estimate all the kernels on the full set of variables (features) allowing for automatic relevance determination (ARD). ARD means that the kernel function has a separate length scale per predictor, e.g. the standard squared exponential kernel with ARD is given by

$$k(\mathbf{x}, \mathbf{x}') = \sigma_{se}^2 \exp\left(\sum_{d=1}^D \frac{(x_d - x'_d)^2}{2l_d}\right),$$

where $\boldsymbol{\phi} = (\sigma_{se}^2, l_1, \dots, l_D)^T$ is a vector of kernel hyperparameters to be estimated. For details of ARD see Neal [30] and Rasmussen and Williams [14, Ch. 5]. The inverse of the length scale parameters l_d , $d = 1, \dots, D$, can be seen as the weight of the corresponding explanatory variable x_d , determining how relevant it is. Hence, ARD kernels provide a built-in method of variable selection.

In the second step we estimated each kernel two more times, on restricted sets of variables. These were subsets of the original variable set indicated as “relevant” by ARD in the first estimation step. We consider two types of “relevance”: with the estimated length scales below 100 (denoted by the suffix ARD) and more restrictive, with length scales below 10 (denoted ARDrstr). We refer to the corresponding kernels as “restricted kernels”. In theory, ARD kernels should automatically eliminate irrelevant inputs by assigning them high length scales. In practice, however, the optimisation of the kernel hyperparameters might be easier numerically when carried out in a space of a lower dimension corresponding to a restricted feature subset.

3.3. Approximate Bayesian computations

The final step in our proposed inference framework is incorporating GP regression from Section 3.2 into ABC. The ABC methods are simulation based algorithms and provide a natural way to make inference in complex systems such as the NSPDE model (1)–(5). As all Bayesian methods, they aim at making statements about the posterior distribution

$$\pi(\boldsymbol{\theta}|\mathbf{y}_{obs}) \propto p(\mathbf{y}_{obs}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}),$$

where $\pi(\boldsymbol{\theta})$ is the prior distribution on the parameter vector $\boldsymbol{\theta}$ and $p(\mathbf{y}_{obs}|\boldsymbol{\theta})$ is the likelihood of the observed data \mathbf{y}_{obs} given $\boldsymbol{\theta}$. The ABC methods are particularly useful when evaluating $p(\mathbf{y}_{obs}|\boldsymbol{\theta})$ is too hard or impossible in practice

since these algorithms circumvent likelihood computations by comparing summary statistics extracted from the observed dataset \mathbf{y}_{obs} and artificial dataset \mathbf{y}^* generated from the model using parameter $\boldsymbol{\theta}^*$ sampled from some proposal distribution. The simplest rejection ABC algorithms, in its basic form, is given as follows, see Pritchard et al. [31]. First, a candidate parameter is sampled from the prior distribution $\boldsymbol{\theta}^* \sim \pi(\boldsymbol{\theta})$ and the model simulator is used to generate the corresponding dataset \mathbf{y}^* . Second, the simulated dataset \mathbf{y}^* is compared to the observed dataset \mathbf{y}_{obs} using a distance metric ρ and a tolerance level ϵ . Third, if the distance between \mathbf{y}_{obs} and \mathbf{y}^* is not higher than the chosen threshold ϵ , the parameter $\boldsymbol{\theta}^*$ that has led to \mathbf{y}^* is accepted.

3.3.1. GP-enhanced semi-automatic ABC

It turns out that in any nontrivial application comparing \mathbf{y}_{obs} and \mathbf{y}^* directly is infeasible. Therefore, the comparison of \mathbf{y}_{obs} and \mathbf{y}^* is based on the distance between summary statistics S extracted from the observed and simulated datasets. In the basic, summary-statistic-based rejection ABC, if $\rho(S(\mathbf{y}^*), S(\mathbf{y}_{obs})) \leq \epsilon$, then the proposed parameter $\boldsymbol{\theta}^*$ is accepted, otherwise it is rejected and a new proposal is made. Such a procedure is repeated N times and the retained $\boldsymbol{\theta}^*$ s constitute a sample from the approximate posterior distribution $\pi_{ABC}(\boldsymbol{\theta}|S(\mathbf{y}_{obs}); \epsilon)$, the accuracy of which is controlled by the tolerance level ϵ .

The choice of summary statistics is crucial for the performance of ABC. However, informative summary statistics are typically problem-specific and hard to devise in the general context. To address this issue, Fearnhead and Prangle [13] propose the semi-automatic approach to constructing summary statistics for ABC based on predictions from a linear regression model. The linear regression model is fitted on a calibration sample of parameter values and features extracted from the corresponding datasets and aims to provide estimates of summary statistics of parameters within ABC iterations. Fearnhead and Prangle [13] consider a separate linear regression model for each element of $\boldsymbol{\theta}$, θ_i , $i = 1, \dots, 10$, and they regress the vector of the i th parameters in the calibration sample $\boldsymbol{\Theta}_i = [\theta_i^{(1)}, \theta_i^{(2)}, \dots, \theta_i^{(M)}]^T$ on the $M \times D$ matrix $\mathbf{X} = [\mathbf{x}^{(j)}]_{j=1, \dots, M}$ constructed using the set of features extracted from the corresponding datasets $\mathbf{y}^{(j)}$, $j = 1, \dots, M$, in the calibration sample:

$$\boldsymbol{\Theta}_i = \beta_0^{(i)} + \mathbf{X}\boldsymbol{\beta}^{(i)} + \boldsymbol{\xi}_i, \quad i = 1, \dots, 10, \quad (7)$$

where $\boldsymbol{\xi}_i$ is a vector of zero-mean noise terms. The linear model prediction, $\hat{\theta}_i^* = \hat{\beta}_0^{(i)} + \mathbf{x}^*\hat{\boldsymbol{\beta}}^{(i)}$, where \mathbf{x}^* is the (row) vector of features extracted from the dataset \mathbf{y}^* , is used as the estimated summary statistic for the proposed θ_i^* and is compared with the corresponding summary statistic for the observed dataset \mathbf{y}_{obs} . The final decision regarding the proposed parameter vector $\boldsymbol{\theta}^*$ is made by comparing $\hat{\theta}_i^*$ and $\theta_i^{obs} = \hat{\beta}_0^{(i)} + \mathbf{x}_{obs}\hat{\boldsymbol{\beta}}^{(i)}$, where \mathbf{x}_{obs} is the (row) vector of features extracted from the observed data \mathbf{y}^{obs} , over different elements of $\boldsymbol{\theta}$ using e.g. the Euclidean distance.

Regression (7) plays a key role in the semi-automatic ABC approach as for the given metric ρ and threshold level ϵ , it is used to make the decision whether or not to accept the proposed $\boldsymbol{\theta}^*$. Therefore, a higher quality of the regression leads to more accurate estimation of the summary statistic and thus improved ABC inference. In highly complex, nonlinear problems, such as the biophysical cell migration model (1)–(5), linear regression is unlikely to provide a very accurate summary statistic. Hence, we extend the semi-automatic ABC approach of Fearnhead and Prangle [13] to allow for a more flexible regression type, for which we propose a nonparametric GP regression model. We refer to our approach as GP-enhanced semi-automatic ABC. In our application in Section 4 we use the final GP specification described in Section 4.3.2.

3.3.2. ABC-SMC

The basic rejection sampling ABC discussed above is known to be inefficient and various extensions have been proposed to result in improved algorithms, such as Markov chain Monte Carlo (MCMC) ABC considered by Marjoram et al. [32] or sequential Monte Carlo (SMC) ABC developed by Beaumont et al. [33]. These algorithms are likelihood-free equivalents of the likelihood-based Bayesian methods, see Sisson et al. [34]. As Sisson et al. [34] point out, ABC-SMC methods are particularly popular in the literature due to their increased efficiency compared to the ABC-MCMC algorithms.

ABC-SMC methods propagate a set of N parameter values called “particles” through a series of C intermediary distributions $\pi_{ABC}(\boldsymbol{\theta}|S(\mathbf{y}_{obs});\epsilon_c)$, $c = 1, \dots, C$, with decreasing tolerance levels: $\epsilon_1 > \dots > \epsilon_C \geq 0$. This ensures that the intermediary distributions gradually provide a closer and closer approximation to the target posterior distribution. The particles representing the c th intermediary distribution are denoted $\{\boldsymbol{\theta}_1^{(c)}, \dots, \boldsymbol{\theta}_N^{(c)}\}$, $c = 1, \dots, C$. If S is a sufficient statistic for $\boldsymbol{\theta}$ (which is usually only available for probabilistic models in the exponential family), then in the limit with respect to the number of intermediary distributions C the true posterior distribution is recovered, otherwise the limiting distribution is an approximation to the posterior distribution, the quality of which depends on the informativeness of S . We adopt the ABC-SMC

variant introduced by Beaumont et al. [9], presented in Algorithm 2.

Algorithm 2: ABC-SMC algorithm Beaumont et al. [9]

Input : model simulator $p(\mathbf{y}|\boldsymbol{\theta})$; prior distribution $\pi(\boldsymbol{\theta})$; distance function $\rho(\cdot, \cdot)$; sequence of decreasing tolerance levels $\epsilon_1, \dots, \epsilon_C$; summary statistics $S(\cdot)$.

```

1 for  $c = 1$  do
2   for  $n = 1$  to  $N$  do
3     while  $\rho(S(\mathbf{y}^*), S(\mathbf{y}_{obs})) \geq \epsilon_1$  do
4       Sample a parameter from the prior:  $\boldsymbol{\theta}^* \sim \pi(\boldsymbol{\theta})$ .
5       Generate data:  $\mathbf{y}^* \sim p(\mathbf{y}|\boldsymbol{\theta}^*)$ .
6       Calculate summary statistics:  $S(\mathbf{y}^*)$ .
7     end
8     Save the proposed parameter:  $\boldsymbol{\theta}_n^{(1)} = \boldsymbol{\theta}^*$ .
9     Calculate particle weight:  $w_n^{(1)} = \frac{1}{N}$ .
10  end
11  Normalise weights:  $W_n^{(1)} = \frac{w_n^{(1)}}{\sum_{m=1}^N w_m^{(1)}}$ , for  $n = 1, \dots, N$ .
12  Set  $\tau_2^2$  as twice the empirical variance of the  $\boldsymbol{\theta}_n^{(1)}$ 's.
13 end
14 for  $c = 2$  to  $C$  do
15   for  $n = 1$  to  $N$  do
16     while  $\rho(S(\mathbf{y}^*), S(\mathbf{y}_{obs})) \geq \epsilon_c$  do
17       Pick  $\boldsymbol{\theta}^{**}$  from  $\boldsymbol{\theta}_n^{(c-1)}$ 's with probabilities  $W_n^{(c-1)}$ .
18       Sample a particle from a perturbed kernel:
19        $\boldsymbol{\theta}^* \sim K(\boldsymbol{\theta}|\boldsymbol{\theta}^{**}; \tau_c^2)$ .
20       Generate data from the model:  $\mathbf{y}^* \sim p(\mathbf{y}|\boldsymbol{\theta}^*)$ .
21       Calculate summary statistics:  $S(\mathbf{y}^*)$ .
22     end
23     Save the proposed parameter:  $\boldsymbol{\theta}_n^{(c)} = \boldsymbol{\theta}^*$ .
24     Calculate particle weight:  $w_n^{(c)} = \frac{\pi(\boldsymbol{\theta}_n^{(c)})}{\sum_{m=1}^N W_m^{(c-1)} K(\boldsymbol{\theta}_n^{(c)}|\boldsymbol{\theta}_m^{(c-1)}; \tau_c^2)}$ .
25   end
26   Normalise weights:  $W_n^{(c)} = \frac{w_n^{(c)}}{\sum_{m=1}^N w_m^{(c)}}$ , for  $n = 1, \dots, N$ .
27   Set  $\tau_{c+1}^2$  as twice the weighted empirical variance of the  $\boldsymbol{\theta}_n^{(c)}$ 's.
28 end

```

Particle degeneracy is a well-known phenomenon in the SMC literature, see Doucet et al. [35]. It consists in the empirical distribution of particles becoming highly skewed after only a few iterations leading to one particle eventually carrying the whole mass. To overcome this problem in the context of ABC-SMC one can fit a density kernel K around each of the N particles from iteration c and use the resulting N -component mixture distribution as the candidate distribution for sampling particles at $c + 1$. In our application we follow Beaumont

et al. [9] and use a Gaussian kernel to perturb the sampled particle.

The idea how to weight the retained particles shares similarities with the standard importance sampling weights and we refer to Toni et al. [36] for the derivation of the weights formula. In brief, the numerator of $w_n^{(c)}$ represents the prior distribution, while the denominator represents a Monte Carlo approximation to the proposal distribution. The latter is the perturbed intermediary distribution at the previous time point.

4. Results

In this section we report our results, in which we continue to make a distinction between the pre-data and post-data phases, see Figure 5. The former phase consists of generating the calibration sample (Section 4.1), selecting the appropriate set of features (Section 4.2) as well as fitting the GP regression models (Section 4.3). The post-date phase includes estimating the parameters of the NSPDE model on the given dataset using the fitted GP regression models for an approximate Bayesian uncertainty quantification as well as running the ABC-SMC algorithm for an additional post-correction (Section 4.4). Moreover, as pointed out in Section 2.3, we focus on two cases of data observability, *Fully Observed Data* (with all the outputs being observable) and *Partially Observed Data* (with only the evolving cell membrane being possible to record). The data we analyse were provided by the organisers of the Cside [37] competition³, in which our results ranked first in the category *Stochastic Differential Equations*.

4.1. Calibration samples

As pointed out in Section 2, neither Neilson et al. [2] nor Tweedy et al. [3] explicitly specify biophysically plausible ranges for the NSPDE parameters. Thus in our application we needed to come up with a working assumption on the plausible parameter region to construct the calibration design [38] used in the pre-data phase of our method. To this end, we experimented with the NSPDE simulator and investigated its outputs for different parameter values (obtained by varying the default parameter vector $\tilde{\theta}$ from [17], specified in Table 1). We then assessed the generated outputs using our prior knowledge in the observation space. These experiments led us to confine the parameter space to a compact domain given by $[0.5\tilde{\theta}, 2\tilde{\theta}]$. This truncation is symmetric on a log scale, and we demonstrate that this domain is broad enough to generate a substantial output variability, see Figure 11 for versatile patterns of cell membrane evolution (see also Figures A.1–A.4 in Supplement A for the variability in the patterns of chemical signal dynamics). Our consultations with cell biophysicists

³Cside stands for “*competitive statistical inference for differential equations*” and was an event hosted at the University of Glasgow, sponsored by EPSRC’s SoftMech and funded by the Biometrika Trust in which participants were invited to infer parameters of differential equation models for the provided datasets.

and biophysicists confirmed that the generated patterns cover typical mechanisms observed in laboratory experiments (personal communication with the last author of [2] and the first author of [3]).

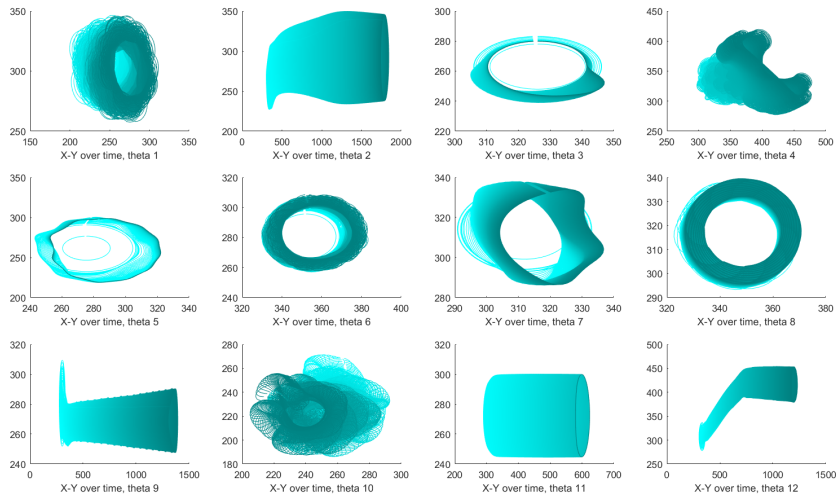


Figure 11: Examples of cell membrane evolution for different parameter vectors for $t = 1, 2, \dots, 1000$. The shade indicates the time point, from early – light, to late – dark.

We note that in a more general context, outside our particular application, truncating the parameter space to a compact domain is meaningful for several reasons. First, extreme parameter values far away from the default ones typically lead to a violation of the physical model assumptions and to singularities in the solutions (demonstrating as crashes of the finite element discretisation programs needed for the numerical solution). Our experiments with the NSPDE simulator of the model (1)–(5) revealed that it is very sensitive to the parameter choice. Second, the first (pre-data) phase of our inference procedure assumes parameter identifiability in the sense that the map from the data to the parameter space is one-to-one, meaning that different features of the data are associated with different parameter values. This assumption may only hold locally, i.e. for a certain neighbourhood around $\tilde{\theta}$. Third and most importantly, the truncation of the parameter space may often reflect actual research practice. When studying the biophysical properties of a system, researchers typically apply prior knowledge from the literature or previous studies carried out in their field. For the particular example of cell movement, domain-experts studying the properties of a given cellular organism can be guided in their inference by parameter ranges available for related cellular organisms, and this prior knowledge can be naturally incorporated into the inference scheme.

We emphasize that the truncation is only needed for the first (pre-data) phase of our method. In fact, it turned out in the Cside [37] competition that some of the ground-truth parameter values were outside of our initial compact domain and we demonstrate later in this section how the second (post-data)

phase of our method based on ABC-SMC can deal with that. We also note that instead of a truncation a squashing function from the real numbers to a compact interval could be used, and the methodological modifications required are straightforward to implement.

We use the compact domain $[0.5\tilde{\theta}, 2\tilde{\theta}]$ to specify two calibration designs (used to generate calibration samples), one for training and the other for validation. The training calibration sample is constructed using 1000 points generated with a Sobol sequence [39] for GP kernels selection (see Section 3.2.2), and further 1000 points (continuing the previous 1000 points) for the refinement of the final kernel specification. The validation calibration sample consists of 100 points generated from a Latin hypercube design [40]. Thus in total we use 2100 calibration samples.

4.2. Feature analysis

The 2100 calibration samples generated using the space filling designs are used to determine relevant features. Since the final feature selection is mostly based on the exploration of the outputs from the NSPDE simulator, below we briefly comment on some of the insights gained from these experiments.

EFA coefficients. As discussed in Section 3.1, we employ the EFA coefficients to model cell shape variation. We use the EFA coefficients as input variables to PCA, which allows us to establish a low-dimensional space to assess similarities and differences across shapes. We apply EFA to all the cell contours, generated over 1000 time points and with 2100 different parameter values in our calibration sample. Here we treat the time series of cell contours generated for the same θ as independent shapes, so PCA is applied to 1000×2100 contours⁴. We use the R package *Momocs* [26] to perform EFA and PCA applied to the resulting harmonic coefficients. The first two principal components are found to explain 74.5% of the total variation.

Since the cell contour at each time period (for each parameter θ) is associated with its own principal components, for each θ we consider a time series of length 1000 of the first and the second principal component (as the first two principal components are already highly informative). We utilise these series in two ways. First, for each series we consider its seven percentiles (1st, 5th, 25th, 50th, 75th, 95th and 99th) which we take as features. Second, for each series we count the number of its zero crossings (separately x and y axis crossings). These zero crossings refer to the cell's path projected in the PCA space. Figure 12 illustrates that the number of zero crossings in 1000 time points provides a useful tool to differentiate between distinct cell types at the same time allowing us to group similar types together. Figure C.1 in Supplement C.1 shows the underlying original cell movements.

⁴More precisely speaking, we carry out PCA twice (independently): once for the training calibration sample of 2000 points and once for the validation calibration sample of 100 points.

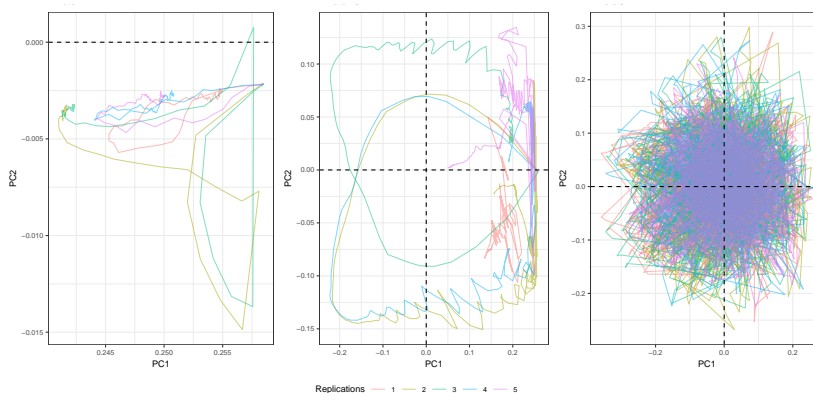


Figure 12: Cell path projections into the PCA space based on Fourier shape descriptors for three different parameter vectors (in separate panels). Each panel shows five replications (depicted with different colours) generated using the same value of θ . Horizontal axis – the 1st principal component; vertical axis – 2nd principal component. Notice that the number of crossing of the x and y axis is very low for the cell on the left panel, medium for the cell in the middle panel, and very large for the cell on the right panel.

Fourier analysis for chemical signals. We carry out PCA on the matrices obtained using the Fourier transform method in Algorithm 1, where the Fourier transform is applied to signals obtained from chemical outputs of GI, LA, LI and S. Figure 13 shows projections into the first two principal component spaces corresponding to the multivariate outputs (LA, LI, S) from the training sample of 1000 Sobol points. The left column corresponds to signals $\alpha_t^{(i)}$ in Algorithm 1 (i.e. a spacial profile at time t of a multivariate chemical signal obtained for the i th parameter in the training calibration sample) set equal to the crude (directly recorded) outputs of LA, LI and S; in the middle column $\alpha_t^{(i)}$ was computed as the centred difference between chemical signals in two subsequent time periods (computed on possibly many finite element nodes); the right column pertains to signals being time series of means (over space) of LA, LI and S. We observe several clusters emerged in these nine PCA spaces, suggesting that the chosen chemical signals based features should provide useful information for predicting the elements of θ . Some signals seem more relevant than others, as they lead to more clear-cut clusters. However, we abstain from a manual selection of relevant features at this point, leaving the task of feature selection to the ARD GP kernels.

Final feature selection and feature importance. Table 2 presents the final selection of the variables used to estimate the GP regression models. We fit a separate GP regression model for each element of θ , separately for *Fully Observed Data* and *Partially Observed Data*. In total 56 and 31 variables are considered for *Fully Observed Data* and *Partially Observed Data*, respectively. Table 3 reports the percentages explained by the first two principal components for the features based on PCA.

Figure 14 illustrates feature importance for predicting 4 selected parameters

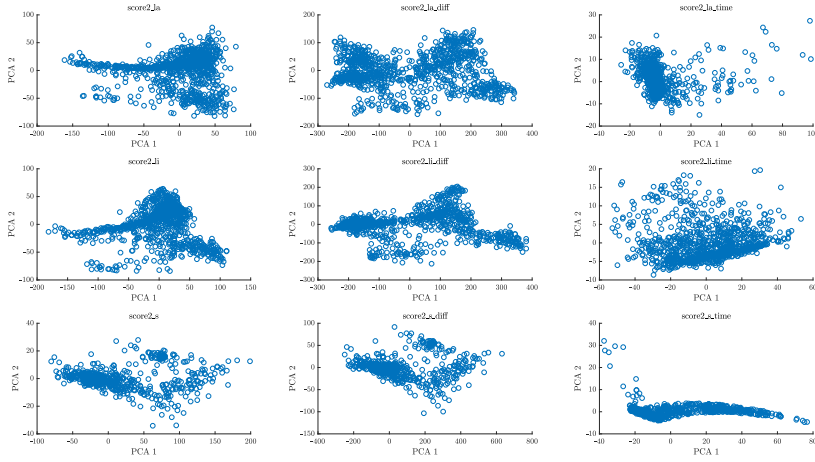


Figure 13: Projections into the spaces spanned by the first two principal components for the training calibration sample of 1000 Sobol points. Each circle corresponds to the projection of a signal generated from the parameter vector corresponding to one of the 1000 training calibration samples. PCA spaces obtained from the Fourier transform method described in Algorithm 1 applied to different signals. Signals (row-wise, from top-left to bottom-right): LA , time differences of LA, time series of means of LA; LI , time differences of LI, time series of means of LI; S , time differences of S, time series of means of S.

for *Fully Observed Data* (f_a, r_c, k_b and D_b). The measure of importance is the inverse length scale from the final GP ARD kernel specification (to be discussed later in Section 4.3.2), with a higher value indicating that the corresponding output is more sensitive to the given feature. We can see that the set of most relevant features can considerably differ between the GP regressions corresponding to the different NSPDE model parameters, with some NSPDE parameters being explained by only a few features (e.g. k_b), while for others multiple features are informative (e.g. r_c). In addition, we find out that almost every feature is relevant for at least one NSPDE parameter. This indicates that the proposed feature selection is sensible and broad enough to capture intricacies of different NSPDE parameters.

4.3. GP results

For both *Fully Observed Data* and *Partially Observed Data* we fit 10 GP regression models, one for each parameter in θ . To select the best performing kernel for each regression we fit 15 kernels on a subsample of 1000 points of the training calibration sample. Such a size of the subsample is large enough to allow for a reliable model selection and feature relevance analysis but results in much shorter estimation times compared to fitting on the full training calibration sample of 2000 points. The full training calibration sample is then used to retrain the final kernel selection. As the regression outputs (NSPDE parameters) come from the initial design and hence have scales in an order that avoids numerical instabilities, we only standardize the inputs to the GP regressions. This is necessary for numerical stability due to different scales of the inputs. We

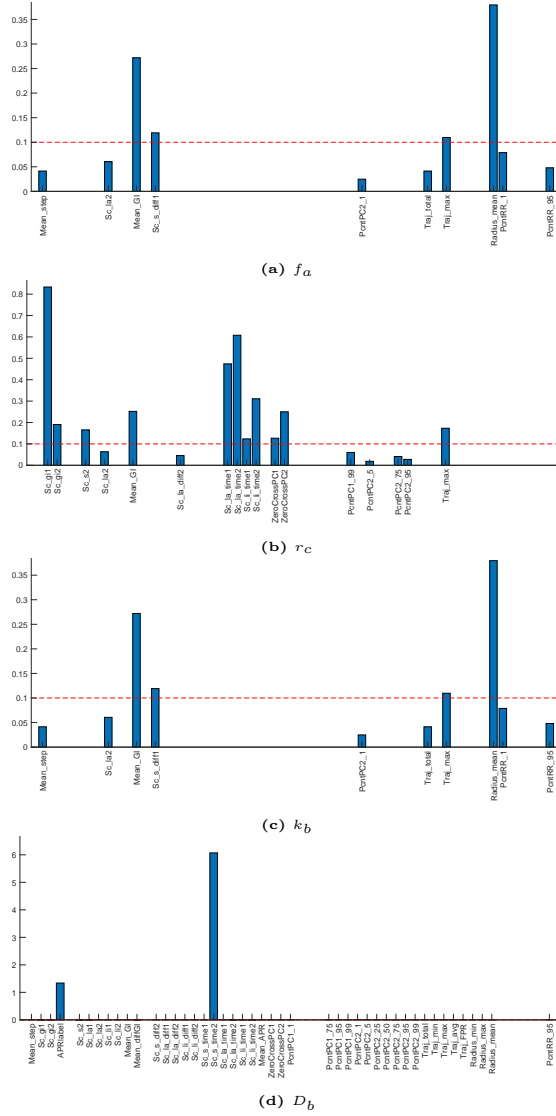


Figure 14: *Fully Observed Data*: estimated inverse length scales in the final GP kernels for 4 selected parameters. For GP kernels with automatic relevance determination (considered in the present paper), the inverse length scales provide a measure of feature relevance, see Section 3.2.2. Horizontal axis – 56 different features (see Table 2). For inverse length scales smaller than 0.01 *bars* are not shown, hence features with *labels* provided have inverse length scales higher than 0.01. In the two top panels the red horizontal line is a benchmark of 0.1 for comparison.

Table 2: Features used in the GP regression. For *Fully Observed Data* all the features were used (56), for *Partially Observed Data* only those from the bottom part (31). * refers to the features constructed as score values corresponding to the \dagger th principal component (PC, $\dagger \in \{1, 2\}$) from PCA applied to the reported value. QPS – quadratic power spectra. Percentiles denoted by \ddagger , with $\ddagger \in \{1, 5, 25, 50, 75, 95, 99\}$.

Feature	Explanation
For <i>Fully Observed Data</i> only (25 in total)	
Mean_step	Mean distance moved by the cell in one unit of time
mean_APR	Mean value of the cell contour area–perimeter ratio for $t \geq 500$
APRlabel	Label of the model best fitting area–perimeter ratio (polynomial order 2–9, exponential order 1–2, Fourier series order 1–5)
mean_GI	Mean value of GI
mean_diffGI	Mean value of the first difference of GI for $t \geq 500$
score2_gi \dagger *	QPS of the time series of GI
Sc.s \dagger *	Concatenated (over time) QPS (amplitudes and normalised frequencies) of S at every 100th t
Sc.la \dagger *	Concatenated (over time) QPS (amplitudes and normalised frequencies) of LA at every 100th t
Sc.li \dagger *	Concatenated (over time) QPS (amplitudes and normalised frequencies) of LI at every 100th t
Sc.s_diff \dagger *	Concatenated (over time) QPS (amplitudes and normalised frequencies) of centred differences (over time) of S
Sc.la_diff \dagger *	Concatenated (over time) QPS (amplitudes and normalised frequencies) of centred differences (over time) of LA
Sc.li_diff \dagger *	Concatenated (over time) QPS (amplitudes and normalised frequencies) of centred differences (over time) of LI
Sc.s_time \dagger *	QPS of the time series of mean (over space) S
Sc.la_time \dagger *	QPS of the time series of mean (over space) LA
Sc.li_time \dagger *	QPS of the time series of mean (over space) LI
For <i>Fully Observed Data</i> and <i>Partially Observed Data</i> (31 in total)	
ZeroCrossPC \dagger	Number of zero crossings for the \dagger th PC from PCA applied to Fourier shape descriptors of cell contour
PcntPC \dagger \ddagger	\ddagger th percentile for the \dagger th PC from PCA applied to Fourier shape descriptors of cell contour
Traj_total	Length of the total trajectory travelled by a cell
Traj_min	Minimum trajectory covered by a cell in one unit of time
Traj_max	Maximum trajectory covered by a cell in one unit of time
Traj_avg	Average trajectory covered by a cell in one unit of time
Traj_FPR	The distance to the furthest point reached by a cell
Radius_max	Maximum radius of a cell across all time points
Radius_min	Minimum radius of a cell across all time points
Radius_mean	Mean radius of a cell across all time points
PcntRR \ddagger	\ddagger th percentile of the distribution of the time series of ratios of minimum to maximum cell radius

use `fitrgp` and `predict` functions from MATLAB Statistics and Machine Learning Toolbox for fitting and predictions.

Table 3: Percentages explained (% Expl.) by the first two principal components ($\dagger \in \{1, 2\}$) for the PCA-based features. See Table 2 for feature explanation.

Feature	% Expl.	Feature	% Expl.	Feature	% Expl.
Sc_s \dagger	44.41	Sc_s_time \dagger	93.40	Sc_s_diff \dagger	43.40
Sc_la \dagger	53.50	Sc_la_time \dagger	41.13	Sc_la_diff \dagger	53.48
Sc_li \dagger	58.70	Sc_li_time \dagger	80.55	Sc_li_diff \dagger	57.27
Sc_gi \dagger	72.46	ZeroCrossPC \dagger	74.77		

4.3.1. Kernel selection: out-of-sample performance and comparison with linear regression

Tables 4 and 5 compare the performance of the 15 kernels considered as well as that of linear regression and the lasso [41] for *Fully Observed Data* and *Partially Observed Data*, respectively, on the validation calibration sample. The comparison is in terms of root mean square relative errors (V_i , $i = 1, \dots, 10$, defined per NSPDE parameter) and mean root total square relative errors (V_{tot}), respectively, computed as

$$V_i^{[s]} = \left(\frac{1}{M} \sum_{j=1}^M \left(\frac{\hat{\theta}_i^{(j)[s]} - \theta_i^{(j)}}{\theta_i^{(j)}} \right)^2 \right)^{1/2}, \quad i = 1, \dots, 10,$$

$$V_{\text{tot}}^{[s]} = \frac{1}{M} \sum_{j=1}^M \left(\sum_{i=1}^{10} \left(\frac{\hat{\theta}_i^{(j)[s]} - \theta_i^{(j)}}{\theta_i^{(j)}} \right)^2 \right)^{1/2},$$

where s refers to the kernel specification (or linear regression or the lasso), $\hat{\theta}_i^{(j)[s]}$ denotes the value of the i th parameter predicted by the GP regression using the s th kernel for the j th validation calibration sample and $\theta_i^{(j)}$ is the true value of the i th parameter used to generate the j th calibration sample.

For completeness, we also define the absolute relative errors ($V_{\text{abs},i}$) and root mean squared errors (V_{avg}) as

$$V_{\text{abs},i} = \frac{|\hat{\theta}_i - \theta_i|}{\theta_i}, \quad i = 1, \dots, 10,$$

$$V_{\text{avg}} = \left(\frac{1}{10} \sum_{i=1}^{10} \left(\frac{\hat{\theta}_i - \theta_i}{\theta_i} \right)^2 \right)^{1/2},$$

where $\hat{\theta}_i$ denotes the point estimate⁵ of the i th NSPDE parameter and θ_i the ground-truth value of the i th parameter. We will use these two measures to

⁵Here, we use the GP-predicted mean as the point estimate, while in the next section we use the estimated modes of the ABC-predicted posterior distribution.

compare the performance of the GP regression approach (based on the final GP specification discussed in Section 4.3.2) with the one of the ABC-SMC approach in Section 4.4 on the observed Cside [37] data, hence we do not need to index these errors by the kernel type (not averaged over M calibration samples).

We first note that in the majority of cases, the GP regressions outperform the linear regression and the lasso, both for individual parameters (V_i) as well as in an overall performance (V_{tot}). Only for 2 parameters in *Partially Observed Data* (r_c and s_a) none of the GP kernels provides an improvement over the linear regression/lasso, however the differences in performance are small in these cases (for r_c the V_i 's are 0.331/0.338 vs 0.339 and for s_a – 0.350/0.350 vs 0.376, for the linear/lasso model and the best GP kernel, respectively)⁶. This is an important result, as Fearnhead and Prangle [13] base their semi-automatic ABC scheme on linear regression and mention that in their application the lasso was not beneficial (even though they note that for regressions with large numbers of explanatory variables the lasso may be appropriate). Since the performance of any summary-statistic-based ABC is determined by the goodness of the underlying summary statistic we have grounds to expect that GP-enhanced semi-automatic ABC works better than the linear-regression-based semi-automatic ABC of Fearnhead and Prangle [13].

Regarding the performance of individual GP kernels, there are four main observations. First, there is no single “strictly dominating” kernel for all the NSPDE parameters so it is worth adopting case-specific kernels. Second, the best performing kernels for the same parameters may differ between *Fully Observed Data* and *Partially Observed Data*, which comes as no surprise given different feature sets in both cases. Third, in several instances restricted kernels improve upon their unrestricted counterparts, which confirms our initial belief regarding an enhanced numerical stability of restricted kernels. Fourth, the out-of-sample performance considerably varies among parameters: e.g. f_a or d_a seem to be relatively easy to infer from the data while D_b , s_a or b_a are much harder to pin down accurately.

4.3.2. Final kernel configuration

Table 6 presents the final kernel for each parameter of the NSPDE system. This selection was made based on the out-of-sample performance in terms of the V_i 's of 15 different kernels estimated on 1000 training calibration samples⁷. To improve this final configuration we re-estimated it on a bigger calibration sample consisting of 2000 Sobol points starting the optimisation from the hyperparameter values delivered by the first estimation. As can be seen in Table 7 this refinement slightly improves the out-of-sample performance of the GP regression in terms of yielding somewhat lower mean V_i (denoted $\bar{V}^{[s]}$) compared

⁶For s_a in *Fully Observed Data* all the GP kernels outperform linear regression but are inferior to the lasso.

⁷If a restricted kernel (ARD or ARDrstr) performed equally well as a general one, the latter one was chosen to allow for extra flexibility.

Table 4: Out-of-sample root mean square relative errors $V_i^{[s]}$, and mean root total square relative errors $V_{\text{tot}}^{[s]}$: *Fully Observed Data*. Regression types: linear, lasso [41], GP. GP regressions with kernels: squared exponential (se), Matérn 3/2 (m32), Matérn 5/2 (m52), rational quadratic (rq), neural network (nn); suffixes ARD and ARDrstr for the kernels estimated in the 2nd step (see Section 3.2.2) on restricted sets of features which were indicated as “relevant” by ARD in the 1st estimation step, with two types of “relevance”: with the estimated inverse length scales above 0.01 (denoted by the suffix ARD) and more restrictive, with inverse length scales above 0.1 (denoted ARDrstr). For the lasso: the regularisation parameter for each regression chosen based on 10-fold cross-validation on the training set. The best performing kernel for each parameter and overall in bold. # GP better linear/lasso – the number GP kernels performing better than the linear regression/lasso.

Kernel	f_a	r_c	k_b	d_b	D_b	k_M	s_a	b_a	D_a	d_a	$V_{\text{tot}}^{[s]}$
se	0.372	0.156	0.292	0.331	0.435	0.209	0.376	0.337	0.455	0.056	1.364
seARD	0.372	0.156	0.292	0.332	0.421	0.209	0.376	0.337	0.456	0.056	1.356
seARDrstr	0.372	0.160	0.176	0.252	0.235	0.206	0.376	0.336	0.435	0.058	1.138
m32	0.372	0.160	0.159	0.367	0.601	0.259	0.376	0.331	0.508	0.056	1.451
m32ARD	0.372	0.153	0.175	0.359	0.629	0.247	0.376	0.331	0.519	0.041	1.472
m32ARDrstr	0.372	0.155	1.000	0.263	1.000	0.349	0.376	0.331	0.355	1.000	2.060
m52	0.372	0.188	0.233	0.325	1.033	0.238	0.376	0.318	0.633	0.058	1.750
m52ARD	0.372	0.153	0.221	0.360	1.014	0.241	0.376	0.318	0.601	0.062	1.726
m52ARDrstr	0.372	0.155	0.380	0.184	1.291	0.294	0.376	0.318	0.411	0.067	1.921
rq	0.108	0.149	0.205	0.348	0.688	0.222	0.376	0.313	0.561	0.060	1.422
rqARD	0.108	0.155	0.204	0.325	0.672	0.222	0.376	0.313	0.556	0.064	1.405
rqARDrstr	0.141	0.158	0.165	0.331	0.291	0.197	0.376	0.313	0.770	0.069	1.307
nn	0.330	0.446	0.371	0.350	0.350	0.489	0.370	0.376	0.357	0.279	1.513
nnARD	0.205	0.307	0.165	0.241	0.233	0.334	0.348	0.360	0.2052	0.147	1.088
nnARDrstr	0.193	0.304	0.178	0.190	0.304	0.309	0.346	0.359	0.402	0.222	1.185
linear	0.651	0.388	0.298	0.887	0.259	0.491	0.713	0.560	0.876	0.513	2.203
lasso	0.201	0.318	0.380	0.361	0.410	0.230	0.325	0.359	0.573	0.116	1.393
# GP better linear	15	14	12	15	2	15	15	15	15	14	15
# GP better lasso	4	14	13	14	5	6	0	12	12	11	6

Table 5: Out-of-sample root mean square relative errors $V_i^{[s]}$, and mean root total square relative errors $V_{\text{tot}}^{[s]}$: *Partially Observed Data*. Regression types: linear, lasso [41], GP. GP regressions with kernels: squared exponential (se), Matérn 3/2 (m32), Matérn 5/2 (m52), rational quadratic (rq), neural network (nn); suffixes ARD and ARDrstr for the kernels estimated in the 2nd step (see Section 3.2.2) on restricted sets of features which were indicated as “relevant” by ARD in the 1st estimation step, with two types of “relevance”: with the estimated inverse length scales above 0.01 (denoted by the suffix ARD) and more restrictive, with inverse length scales above 0.1 (denoted ARDrstr). For the lasso: the regularisation parameter for each regression chosen based on 10-fold cross-validation on the training set. The best performing kernel for each parameter and overall in bold. # GP better linear/lasso – the number GP kernels performing better than the linear regression/lasso.

Kernel	f_a	r_c	k_b	d_b	D_b	k_M	s_a	b_a	D_a	d_a	$V_{\text{tot}}^{[s]}$
se	0.203	0.347	0.291	0.229	0.323	0.306	0.376	0.340	0.321	0.125	1.191
seARD	0.203	0.347	0.291	0.229	0.327	0.306	0.376	0.340	0.321	0.125	1.190
seARDrstr	0.204	0.347	0.286	0.248	0.327	0.306	0.376	0.340	0.321	0.126	1.200
m32	0.211	0.339	0.279	0.223	0.328	0.306	0.376	0.337	0.319	0.123	1.181
m32 ARD	0.211	0.340	0.283	0.222	0.328	0.306	0.376	0.337	0.319	0.123	1.184
m32ARDrstr	0.219	0.342	1.000	0.366	0.328	0.306	0.376	0.337	0.319	0.129	1.578
m52	0.211	0.346	0.283	0.225	0.328	0.299	0.376	0.338	0.318	0.123	1.186
m52ARD	0.211	0.346	0.283	0.226	0.328	0.299	0.376	0.338	0.318	0.123	1.187
m52ARDrstr	0.210	0.346	0.286	0.330	0.328	0.299	0.376	0.338	0.318	0.123	1.234
rq	0.214	0.353	0.291	0.228	0.326	0.298	0.376	0.336	0.303	0.121	1.183
rqARD	0.214	0.353	0.290	0.228	0.326	0.298	0.376	0.336	0.303	0.125	1.183
rqARDrstr	0.214	0.353	0.282	0.308	0.326	0.298	0.376	0.336	0.329	0.126	1.224
nn	0.350	0.381	0.341	0.261	0.336	0.337	0.375	0.371	0.337	0.226	1.351
nnARD	0.276	0.353	0.412	0.331	0.337	0.314	0.397	0.369	0.343	0.180	1.372
nnARDrstr	0.276	0.365	0.412	0.331	0.337	0.300	0.397	0.367	0.343	0.180	1.371
linear	0.239	0.331	0.288	0.259	0.339	0.304	0.350	0.381	0.312	0.175	1.235
lasso	0.236	0.338	0.283	0.252	0.339	0.307	0.350	0.373	0.336	0.178	1.238
# GP better linear	12	0	7	9	15	7	0	15	2	12	11
# GP better lasso	12	0	2	9	15	13	0	15	12	12	11

to the configuration estimated on 1000 Sobol points (the mean V_i is decreased by 2% and 3% for *Fully Observed Data* and *Partially Observed Data*, respectively).

Table 6: Final kernels used for *Fully Observed Data* and *Partially Observed Data*. R/NR refers to ‘retrained’/‘not retrained’ i.e. estimated or not on 2000 calibration samples, starting from the estimates from 1000 calibration samples. Kernel types: squared exponential (se), Matérn 3/2 (m32), Matérn 5/2 (m52), rational quadratic (rq), neural network (nn); suffixes ARD and ARDrstr for the kernels estimated in the 2nd step (see Section 3.2.2) on restricted sets of features which were indicated as “relevant” by ARD in the 1st estimation step, with two types of “relevance”: with the estimated inverse length scales above 0.01 (denoted by the suffix ARD) and more restrictive, with inverse length scales above 0.1 (denoted ARDrstr).

Parameter	<i>Fully Observed Data</i>		<i>Partially Observed Data</i>	
f_a	seARD	NR	m32ARDrstr	R
r_c	m32	NR	m32ARD	R
k_b	m32	NR	qrARDrstr	R
d_b	m32ARD	R	nn	R
D_b	rqARD	R	m52ARDrstr	R
k_M	rq	R	nn	R
s_a	nn	NR	nn	R
b_a	rq	R	m52ARDrstr	R
D_a	rqARD	R	nnARD	R
d_a	rq	R	qrARD	R

Table 7: Out-of-sample fit: the single best performing kernel (lowest $V_{\text{tot}}^{[s]}$) from Tables 4 and 5 (nnARD and m32, respectively) and the final configuration before refinement (bf. refin.) and after refinement (aft. refin.) on 2000 Sobol points. Selecting the best kernel for each parameter results in decreased prediction errors, which are further diminished by retraining on a larger dataset. $\bar{V}^{[s]} = 0.1 \sum_{i=1}^{10} V_i^{[s]}$, i.e. denotes the mean over 10 parameters of per-parameter V_i ’s.

Measure	Single best kernel	Combined bf. refin.	Combined aft. refin.
<i>Fully Observed Data</i>			
$\bar{V}^{[s]}$	0.255	0.194	0.190
$V_{\text{tot}}^{[s]}$	1.088	0.868	0.858
<i>Partially Observed Data</i>			
$\bar{V}^{[s]}$	0.284	0.280	0.272
$V_{\text{tot}}^{[s]}$	1.181	1.171	1.145

4.3.3. Results on the Cside data

Table 8 presents the results obtained on the datasets from the Cside [37] competition. Shaded fields indicate the cases for which the ground-truth values turned out to be outside the prior range. As expected, the relative absolute errors are typically 4–10 times lower for *Fully Observed Data* than for *Partially Observed Data*, indicating the importance of the information provided by chemical signals. Importantly, with the sole exception of D_a , the true values of the non-shaded parameters always lie in the estimated 2-standard deviation width CI.

Table 8: GP results on the Cside [37] data: means $\hat{\theta}_i$ and standard deviations (in parentheses) for *Fully Observed Data* and *Partially Observed Data*, together with the ground-truth values θ_i and the absolute relative error $V_{\text{abs},i} = |\hat{\theta}_i - \theta_i|/\theta_i$. Shaded fields indicate the cases for which the ground-truth values were outside the calibration domain used in phase 1 of our inference method.

	f_a	r_c	k_b	d_b	D_b	k_M	s_a	b_a	D_a	d_a
<i>Fully Observed Data</i>										
$\hat{\theta}_i$	0.0030	0.0541	0.0041	0.0142	0.0617	0.2474	8.066e-5	0.1395	0.0299	0.0240
std($\hat{\theta}_i$)	(0.0010)	(0.0175)	(0.0010)	(0.0033)	(0.0132)	(0.0433)	(0.0011)	(0.0417)	(0.0068)	(0.0029)
θ_i	0.0032	0.0500	0.0035	0.0130	0.0600	0.3000	8.000e-5	0.3000	0.0250	0.0250
$V_{\text{abs},i}$	0.0730	0.0827	0.1840	0.0933	0.0290	0.1754	0.0083	0.5351	0.1976	0.0395
<i>Partially Observed Data</i>										
$\hat{\theta}_i$	0.0024	0.0801	0.0037	0.0167	0.0541	0.2103	7.650e-5	0.1220	0.0366	0.0176
std($\hat{\theta}_i$)	(0.0010)	(0.0326)	(0.0011)	(0.0045)	(0.0204)	(0.0635)	(0.0010)	(0.0458)	(0.0101)	(0.0105)
θ_i	0.0040	0.0900	0.0020	0.0100	0.0500	0.1000	7.000e-5	0.0750	0.0150	0.0200
$V_{\text{abs},i}$	0.3921	0.1098	0.8286	0.6719	0.0816	1.1034	0.0929	0.6267	1.4398	0.1201

Figures C.3 and C.5 in Supplementary Material C present datasets generated using the predicted mean parameter values for *Fully Observed Data* and for *Partially Observed Data*, respectively. Comparing these with their ground-truth counterparts in Figures C.3b and C.5b reveals that the developed GP regression framework succeeds in capturing numerous visual properties of the observed datasets. Again, the resemblance is particularly high for *Fully Observed Data*, with the predicted chemical signals being very similar to the ground-truth ones and with the predicted membrane trajectory characterised by long, rather directed movement. Also cell shapes are very alike for the predicted and real datasets in this case. For *Partially Observed Data* the visual similarity is somewhat smaller, however it is worth noting that the GP regression framework is still able to reproduce elongated cell shapes and directed movement. This is especially noticeable when one takes into consideration a dataset generated using the default parameter value, depicted in Figure 1, where the cell wanders randomly.

4.4. ABC results

We first briefly report the results from the initial small-scale ABC phase (see Figure 5), submitted to the Cside [37] competition, and move to a thorough analysis of the large-scale ABC study thereafter. We split the latter into two subparts: the single parameter analysis, in which only one parameter is updated with the remaining nine parameters fixed to their GP estimates from Table 8, and multiple parameter analysis, in which all ten parameters are updated simultaneously. For all the studies we use the ABC-SMC algorithm of Beaumont et al. [9] implemented in the R package `EasyABC` created by Jabot et al. [42].

In the initial small-scale ABC-SMC study we used $N = 40$ particles and a four-step tolerance schedule (set to $\{40, 20, 10, 5\}$ and $\{20, 10, 5, 2.5\}$ for *Fully Observed Data* and *Partially Observed Data*, respectively). For the smallest

tolerance levels we recorded the rejection rates of 3.68% for *Fully Observed Data* and 21.1% for *Partially Observed Data*. The prior distributions were set to $\mathcal{U}(0.3\tilde{\theta}_i, 0.5\tilde{\theta}_i)$, $i = 1, \dots, 10$, where $\tilde{\theta}_i$ is the default parameter value from Tweedy [17] (see Table 1). Table 9 presents the results. Out of the ten NSPDE model parameter, two parameters for *Fully Observed Data* (k_b and D_a) and six parameters for *Partially Observed Data* show an improvement compared to the GP regression estimates in terms of the $V_{\text{abs},i}$'s. Even though these improvements are only minor, this preliminary ABC analysis constitutes an important step in our inference framework, as the obtained results allow us to decide upon the tuning parameters of the ABC-SMC algorithms for the full large-scale study. Moreover, the results delivered by the initial ABC run provide a promising starting point for further investigation and are particularly encouraging for *Partially Observed Data*.

Table 9: ABC results on the Cside [37] data, submitted to the Cside [37] competition: MAP estimates $\hat{\theta}_i$ and standard deviations (in parentheses) for *Fully Observed Data* and *Partially Observed Data*, together with the ground-truth values θ_i and the absolute relative error $V_{\text{abs},i} = |\hat{\theta}_i - \theta_i|/\theta_i$. Shaded columns indicate the parameters for which the ground-truth values were outside the calibration domain used in Phase 1 of our inference method. Bold values indicate the ABC parameter estimates that are better (in terms of absolute relative error) compared to the GP estimates.

	f_a	r_c	k_b	d_b	D_b	k_M	s_a	b_a	D_a	d_a
<i>Fully Observed Data</i>										
$\hat{\theta}_i$	0.0027	0.0315	0.0040	0.0081	0.0752	0.1826	5.927e-5	0.1007	0.0296	0.0352
std($\hat{\theta}_i$)	(0.0002)	(0.0226)	(0.0009)	(0.0058)	(0.0180)	(0.0690)	(3.306e-5)	(0.0387)	(0.0096)	(0.0072)
θ_i	0.0032	0.0500	0.0035	0.0130	0.0600	0.3000	8.000e-5	0.3000	0.0250	0.0250
$V_{\text{abs},i}$	0.1718	0.3698	0.1562	0.3773	0.2540	0.3913	0.2591	0.6644	0.1850	0.4085
<i>Partially Observed Data</i>										
$\hat{\theta}_i$	0.0023	0.1187	0.0031	0.0115	0.0668	0.2072	4.705e-5	0.0932	0.0234	0.0188
std($\hat{\theta}_i$)	(0.0006)	(0.0346)	(0.0012)	(0.0045)	(0.0198)	(0.0750)	(3.301e-5)	(0.0392)	(0.0102)	(0.0078)
θ_i	0.0040	0.0900	0.0020	0.0100	0.0500	0.1000	7.000e-5	0.0750	0.0150	0.0200
$V_{\text{abs},i}$	0.4349	0.3192	0.5731	0.1522	0.3353	1.0725	0.3278	0.2430	0.5580	0.0575

In the remaining part of this section we report the results carried out after the Cside [37] competition, corresponding to the ultimate step of our inference framework, i.e. the full-scale ABC analysis. The aim of this investigation is to learn in which cases a more considerable improvement over the GP regression estimates can be expected from ABC when run with more particles and more algorithm iterations. We fix the number of particles to $N = 100$ and decrease the tolerance level linearly, with ten intermediary distributions. We set the prior distribution for each θ_i as $\mathcal{N}(1.25\tilde{\theta}_i, (0.375\tilde{\theta}_i)^2)$, where $\tilde{\theta}_i$ is the default parameter value from Tweedy [17] (see Table 1). Recall that the compact calibration domain (used for GP regression fitting) is $[0.5\tilde{\theta}, 2\tilde{\theta}]$, so 95% of the mass of the chosen prior distribution for ABC is in this interval. Keeping 5% of the prior distribution mass outside the calibration domain allows us to explore those regions in the parameter space which might be missed by our calibration

design. Specifically, after the submission to the Cside [37] competition it turned out that for 3 out of 20 parameters (for both *Fully Observed Data* and *Partially Observed Data*) the ground-truth values were outside our calibration domain, therefore in our final ABC phase we aim to look into whether the ABC-SMC algorithm is able to correct for this mismatch.

4.4.1. Single parameter analysis

We first investigate whether the ABC-SMC algorithm can improve upon the GP estimates when we infer each parameter separately, with the other nine parameters being fixed at their corresponding GP estimates from Table 8. For each parameter, we run two replications to assess the convergence. The results of the single parameter analysis are presented in Figures 16 and 17 in the form of ABC posterior density estimates for each of the ten intermediary distributions, for *Fully Observed Data* and *Partially Observed Data*, respectively. We report the maximum *a posteriori* (MAP) estimates, standard deviations and AREs in Table 10.

Table 10: ABC results on the Cside [37] data, single parameter analysis: MAP estimates $\hat{\theta}_i$ and standard deviations (in parentheses) for *Fully Observed Data* and *Partially Observed Data*, together with the ground-truth values θ_i and the absolute relative error $V_{\text{abs},i} = |\hat{\theta}_i - \theta_i|/\theta_i$. Shaded columns indicate the parameters for which the ground-truth values were outside the calibration domain used in phase 1 of our inference method. Bold values indicate the ABC parameter estimates that are better (in terms of absolute relative error) compared to the GP estimates.

	f_a	r_c	k_b	d_b	D_b	k_M	s_a	b_a	D_a	d_a
<i>Fully Observed Data</i>										
$\hat{\theta}_i$	0.0033	0.0536	0.0042	0.0145	0.0782	0.2458	6.46e-05	0.1788	0.0276	0.0247
std($\hat{\theta}_i$)	(0.0001)	(0.0136)	(0.0001)	(0.0012)	(0.0067)	(0.0141)	(1.66e-05)	(0.0345)	(0.0012)	(0.0009)
θ_i	0.0032	0.0500	0.0035	0.0130	0.0600	0.3000	8.00e-5	0.3000	0.0250	0.0250
$V_{\text{abs},i}$	0.0229	0.0714	0.2027	0.0067	0.3027	0.1806	0.1921	0.4039	0.1039	0.0111
<i>Partially Observed Data</i>										
$\hat{\theta}_i$	0.0034	0.0077	0.0043	0.0139	0.0503	0.1734	8.23e-06	0.0688	0.0363	0.0184
std($\hat{\theta}_i$)	(0.0002)	(0.0049)	(0.0004)	(0.0003)	(0.0012)	(0.0163)	(8.26e-07)	(0.0355)	(0.0028)	(0.0017)
θ_i	0.0040	0.0900	0.0020	0.0100	0.0500	0.1000	7.00e-5	0.0750	0.0150	0.0200
$V_{\text{abs},i}$	0.1406	0.9139	1.1401	0.3903	0.0051	0.7336	0.8823	0.0826	1.4200	0.0784

For *Fully Observed Data*, ABC provides better estimates (modes of the ABC posterior densities) for six out of the ten parameters than the GP regression alone. These can be seen as the points below the zero-line in the top panel of Figure 15, which presents the differences in $V_{\text{abs},i}$'s by means of the Bland-Altman plot (the errors of the GP estimates are subtracted from the errors of the ABC MAP estimates, hence the points below the zero-line correspond to the parameters for which ABC-SMC outperforms the GP regressions). We note that this and the subsequent Bland-Altman plots are included for visualization only and do not constitute formal statistical hypothesis tests. Regarding the overall performance, since in this case the data contain chemical signals, the GP

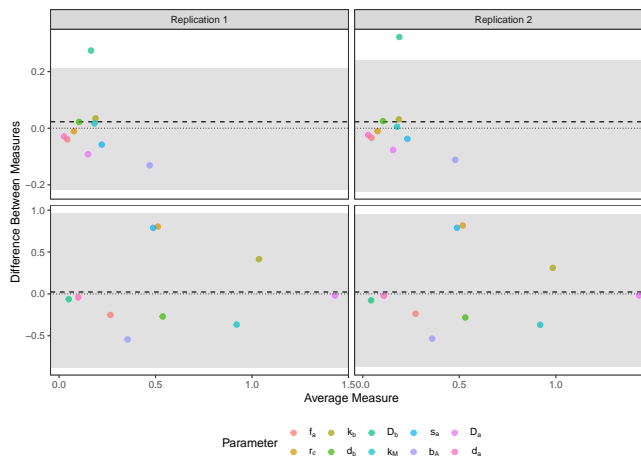


Figure 15: Bland-Altman plot for the single parameter analysis, for two independent replications, top: *Fully Observed Data*, bottom: *Partially Observed Data*. ABC and GP method comparison based on the absolute relative error $V_{\text{abs},i} = |\hat{\theta}_i - \theta_i|/\theta_i$, with lower values indicating a better fit. x axis: the mean of the ABC and GP $V_{\text{abs},i}$'s; y axis – the difference between the ABC and GP $V_{\text{abs},i}$'s; dashed horizontal line – the mean difference between the ABC and GP $V_{\text{abs},i}$'s; grey area – mean $\pm 1.96 \times$ standard deviations. The points under the 0 line (dotted) correspond to the ABC estimates with lower $V_{\text{abs},i}$'s than the GP estimates.

regressions fitted to the full set of features (i.e. including the chemical-signal-based features) are already pretty accurate and the ABC algorithm is able to only marginally improve upon them: V_{avg} is equal to 0.199 for ABC, compared to 0.202 for GP regression. Moreover, the fact that the remaining nine parameters are fixed to essentially incorrect values (i.e. GP estimates) explains why the ABC posterior densities for the currently inferred parameter may concentrate quite far from both the corresponding GP estimates and ground-truth values – as it is the case for D_b and b_a , see Figure 16.

For *Partially Observed Data*, for seven out of the ten parameters the single-parameter algorithm delivers ABC posterior density estimates with modes closer to the ground-truth values than the GP estimates alone. This is also illustrated in the Bland-Altman plot in Figure 15, bottom row. However, V_{avg} is higher for ABC (0.752) than for GP regressions (0.711), which signals a slightly lower overall performance of ABC. Figure 17 shows that the ABC estimates for some parameters, such as D_b , b_a and d_a , are close to the ground-truth values, and for other parameters, such as D_a , k_b and k_M , both the GP and the ABC estimates are far from the ground-truth values. This indicates that the features based only on cell contours (and not using chemical signals) might not be informative enough about some of the NSPDE model parameters. This comes as no surprise and confirms that the set-up of *Partially Observed Data* was indeed more challenging than the one of *Fully Observed Data*. Finally, for some parameters, such as r_c and s_a , the ABC sampler seems to converge very far away from both the GP estimates and the ground-truth values. This issue is consistently observed over two independent replications, which again suggests that this is

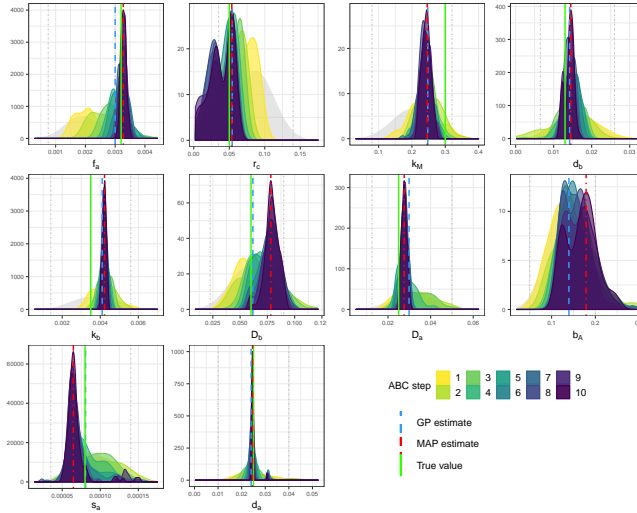


Figure 16: ABC-SMC posterior density estimates for the single parameter analysis (with other nine parameters fixed to their GP estimates) for *Fully Observed Data*. Approximate posterior distributions for NSPDE model parameters in each panel, with the prior distribution (grey) set to $\mathcal{N}(1.25\tilde{\theta}_i, (0.375\tilde{\theta}_i)^2)$, where $\tilde{\theta}_i$ – the default parameter value from Tweedy [17]. The number of accepted particles in each intermediary distribution is $N = 100$.

the consequence of fixing other nine parameters at incorrect values (i.e. GP estimates).

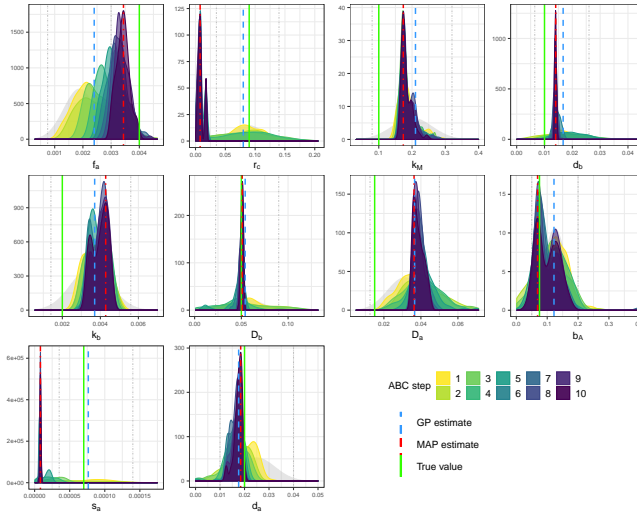


Figure 17: ABC-SMC posterior density estimates for the single parameter analysis (with other nine parameters fixed to their GP estimates) for *Partially Observed Data*. Approximate posterior distributions for NSPDE model parameters in each panel, with the prior distribution (grey) set to $\mathcal{N}(1.25\tilde{\theta}_i, (0.375\tilde{\theta}_i)^2)$, where $\tilde{\theta}_i$ – the default parameter value from Tweedy [17]. The number of accepted particles in each intermediary distribution is $N = 100$.

Overall, the results from the single parameter analysis indicate that in several cases ABC-SMC can infer the NSPDE model parameter more accurately compared to the GP alone. We note that constraining the remaining parameters to be fixed at the GP estimates may cause a bias in the conditional posterior distribution of the focal parameter estimated with ABC. A natural extension of the single parameter analysis is thus to allow all the parameters to be inferred simultaneously, which we discuss in the next section.

4.4.2. Multiple parameter analysis

Table 11 presents the results, for *Fully Observed Data* and *Partially Observed Data*, from the main part of our ABC study based on updating all the NSPDE parameters together. The results for *Fully Observed Data* reveal that seven out of the ten parameters are inferred more accurately with ABC-SMC (using the modes of the ABC posterior density estimates) compared to the GP regression alone. This is illustrated in the Bland-Altman plot in Figure 18 (top row), in which the differences between the errors ($V_{\text{abs},i}$) from ABC and GP regressions are mostly below the zero-line. Unlike in the single parameter study, this time the ABC-SMC algorithm also leads to an overall inference improvement: V_{avg} amounts to 0.166 and 0.202 for ABC and GP regressions, respectively. Figure 19 shows that the ABC posterior density estimates concentrate around the ground-truth values, with the only exception of parameter b_A . While the (Gaussian) prior distributions allow the ABC algorithm to sample outside the calibration domain, the ground-truth value for b_a , equal to 0.3, is located too far in the tail of the prior distribution for this parameter, which hinders the convergence of the ABC algorithm and results in a finite-simulation ABC posterior concentrating far from the ground-truth value.

Table 11: Multiple parameter analysis results: MAP estimates $\hat{\theta}_i$ and standard deviations (in parentheses) for *Fully Observed Data* and *Partially Observed Data*, together with the ground-truth values θ_i and the absolute relative error $V_{\text{abs},i} = |\hat{\theta}_i - \theta_i|/\theta_i$. Shaded columns indicate the parameters for which the ground-truth values were outside the sampling design interval. Bold values indicate the ABC parameter estimates that were better (in terms of absolute relative error) compared to the GP estimates.

	f_a	r_c	k_b	d_b	D_b	k_M	s_a	b_a	D_a	d_a
<i>Fully Observed Data</i>										
$\hat{\theta}_i$	0.0029	0.0558	0.0037	0.0142	0.0606	0.2962	8.07e-05	0.1532	0.0234	0.0252
std($\hat{\theta}_i$)	(0.0007)	(0.0325)	(0.0007)	(0.0033)	(0.0353)	(0.1190)	(3.5e-05)	(0.0335)	(0.0223)	(0.0050)
θ_i	0.0032	0.0500	0.0035	0.0130	0.0600	0.3000	8.00e-5	0.3000	0.0250	0.0250
$V_{\text{abs},i}$	0.0791	0.1151	0.0704	0.0897	0.0106	0.0128	0.0093	0.4895	0.0622	0.0097
<i>Partially Observed Data</i>										
$\hat{\theta}_i$	0.0017	0.0924	0.0038	0.0152	0.0664	0.2032	8.44e-05	0.1068	0.0307	0.0244
std($\hat{\theta}_i$)	(0.0010)	(0.0379)	(0.0016)	(0.0067)	(0.0279)	(0.0938)	(4.15e-05)	(0.0581)	(0.0124)	(0.0109)
θ_i	0.0040	0.0900	0.0020	0.0100	0.0500	0.1000	7.00e-5	0.0750	0.0150	0.0200
$V_{\text{abs},i}$	0.5770	0.0264	0.8847	0.5210	0.3288	1.0321	0.2058	0.4233	1.0466	0.2192

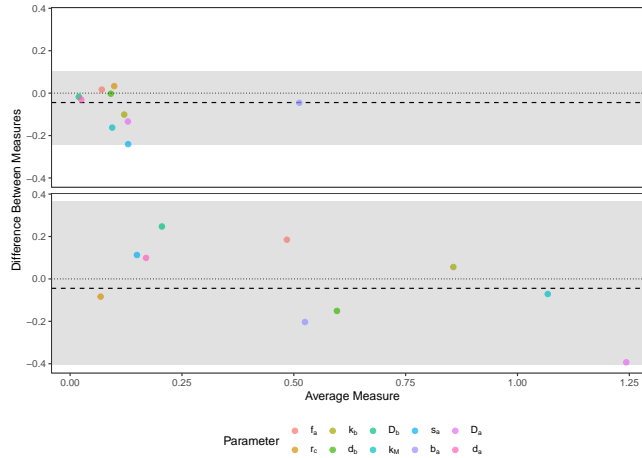


Figure 18: Bland Altman plot for multiple parameter analysis, top: *Fully Observed Data*, bottom: *Partially Observed Data*. ABC and GP method comparison based on the absolute relative error $V_{\text{abs},i} = |\hat{\theta}_i - \theta_i|/\theta_i$, with lower values indicating a better fit. x axis: the mean of the ABC and GP $V_{\text{abs},i}$'s; y axis – the difference between the ABC and GP $V_{\text{abs},i}$'s; dashed horizontal line – the mean difference between the ABC and GP $V_{\text{abs},i}$'s; grey area – mean $\pm 1.96 \times$ standard deviations. The points under the 0 line (dotted) correspond to the ABC estimates with lower $V_{\text{abs},i}$'s than the GP estimates.

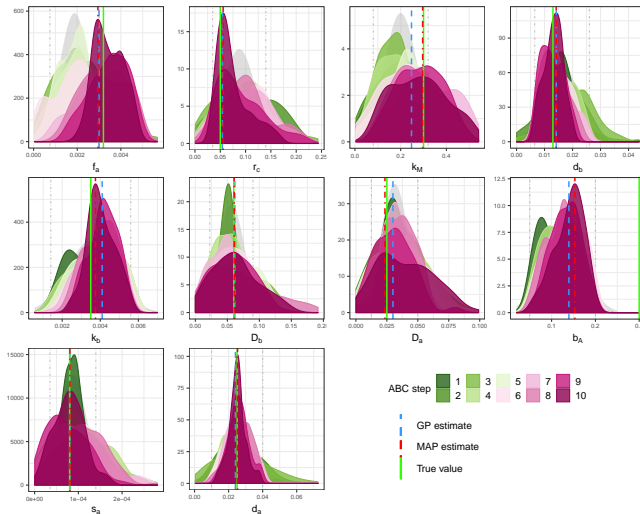


Figure 19: ABC-SMC posterior density estimates for multiple parameter analysis (all parameters inferred simultaneously) for *Fully Observed Data*. Marginal approximate posterior distributions for NSPDE model parameters in each panel, with the prior distribution (grey) set to $\mathcal{N}(1.25\tilde{\theta}_i, (0.375\tilde{\theta}_i)^2)$, where $\tilde{\theta}_i$ – the default parameter value from Tweedy [17]. The number of accepted particles in each intermediary distribution is $N = 100$.

For *Partially Observed Data* the results from Table 11 indicate that the ABC-SMC point estimates (modes of the ABC posterior density estimates) are closer to the ground-truth values than the GP estimates for five out of the

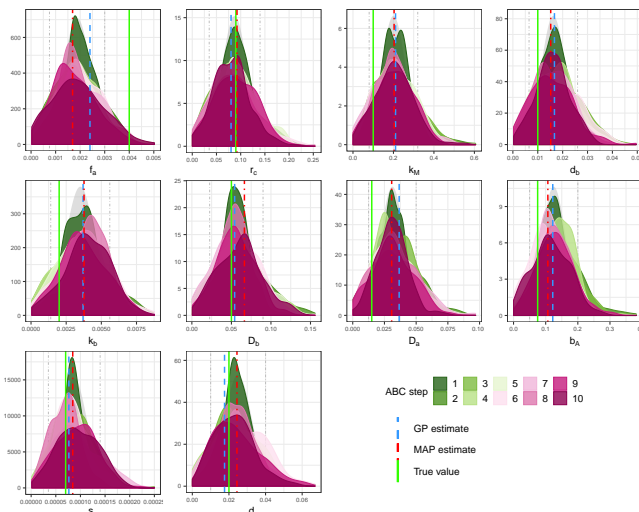


Figure 20: ABC-SMC posterior density estimates for multiple parameter analysis (all parameters inferred simultaneously) for *Partially Observed Data*. Marginal approximate posterior distributions for NSPDE model parameters in each panel, with the prior distribution (grey) set to $\mathcal{N}(1.25\hat{\theta}_i, (0.375\hat{\theta}_i)^2)$, where $\hat{\theta}_i$ – the default parameter value from Tweedy [17]. The number of accepted particles in each intermediary distribution is $N = 100$.

ten parameters. This can be also seen in the Bland-Altman plot in Figure 18, bottom row. However, the overall performance of the ABC-SMC algorithm is much better than that of the GP regression model, with V_{avg} equal to 0.626 and 0.711 for ABC and GP regressions, respectively.

The ABC posterior density estimates from the multiple parameter analysis shown in Figure 20 are wider compared to their single parameter counterparts in Figure 17. This is natural given an increased complexity of inferring all ten parameters simultaneously. However, this increased uncertainty is compensated by a more accurate inference which results from inferring all the parameter simultaneously and not fixing some of them to the GP estimates. In particular, the ABC posterior densities for s_a and r_c are now concentrated around the true values of these parameters. For the multiple parameter analysis the highest $V_{\text{abs},i}$'s are obtained for parameters k_b , k_M and D_a , which suggests that these parameters are intrinsically difficult to estimate – presumably because the features extracted from the data, used to fit the GP regressions, do not provide enough information to infer these parameters more accurately. Comparing the errors ($V_{\text{abs},i}$) for *Fully Observed Data* and *Partially Observed Data*, see Table 11 and Figure 18, we can see that the errors for the former are considerably lower. This is a consequence of extracting the extended set of features, including chemical-signals-based features, and hence improved GP predictions.

4.4.3. The benefits from using ABC-SMC correction

Our analysis shows the advantage of the final phase in our inference scheme (see Figure 5): the GP-enhanced ABC-SMC method for multiple parameters

leads to more accurate predictions, compared to using only the predictions from the GP regression model, for most of the NSPDE model parameters for *Fully Observed Data* and for half of the parameters for *Partially Observed Data*. Our proposed inference framework also improves the overall inference as measured by V_{avg} . Moreover, the single parameter analysis reveals that the estimation of those parameters that happen to be outside the compact calibration domain used in Phase 1 can particularly benefit from our developed GP-enhanced ABC scheme. In this updating scheme, for the three parameters (out of twenty in total) for which the ground-truth values were outside our initial calibration domain, ABC is more successful in extrapolating outside the initial design than the GP regressions alone and leads to estimates closer to the ground-truth values.

Finally, we note that an important advantage of the ABC step is related to the associated flexible uncertainty quantification mechanism. While the GP regressions alone are able to provide fairly accurate point estimates of the NSPDE parameters, the implied posterior distribution is necessarily Gaussian as a consequence of the assumed residual distribution in the model (6). As Figures 19 and 20 show, the ABC-posterior marginal densities are non-elliptical (often highly skewed), which suggests that the Gaussian assumption might be unrealistic in this case. By relaxing the Gaussianity constraint, the ABC-SMC step provides a crucial correction to the *shape* of the posterior distribution and hence overall uncertainty quantification.

5. Discussion

Parameter estimation in complex biophysical systems described by nonlinear stochastic partial differential equations (NSPDEs) is a challenging problem, due to the intractability of the associated likelihood and the high computational costs of the numerical integration of the NSPDEs. In the present article, we have explored an approach based on approximate Bayesian computation (ABC), which is a popular method in the kit of likelihood-free “plug and play” tools. ABC effectively replaces the unknown likelihood by a metric in some feature space. The challenge in practical applications is to extract informative features, or summary statistics, from the data, and appropriately weight and combine them. Our work has been motivated by the semi-automatic ABC paradigm proposed by Fearnhead and Prangle [13]. Here, the authors apply a linear regression approach, based on preliminary features (obtained in a pilot run of forward simulations) as explanatory variables and each model parameter as a response. The outputs of these linear regression models are then used as refined features for the subsequent ABC simulations. In our work, we have replaced linear regression by more flexible Gaussian process (GP) regression, making use of automatic relevance determination (ARD) kernels for automatic feature selection and weighting.

Our findings indicate that the GP regression approach to extracting summary statistics achieves a significant improvement over the linear regression approach of Fearnhead and Prangle [13]. This improvement can also be seen when subjecting linear regression to L1-regularisation for Lasso-based variable

selection, suggesting that it is the nonlinear flexibility of the GP that is the source of the improvement. In fact, our GP predictions achieved point parameter estimates that, in most cases, were so accurate that they left little room for improvement (in terms of location) with ABC. If the inference could only be based on GP regression (without ABC), this would achieve a substantial reduction in the computational costs.

However, there are two practical problems for the GP regression approach. First, it is subject to a prior restriction to a compact domain in parameter space. This is required for the GP training phase, where the compact set is covered with a space-filling design, and forward simulations from the NSPDE model are run from the parameters defined by the design points. In principle this compact domain could be enforced via a nonlinear squashing function defined over the set of real numbers. However, this approach would lead to a low effective density of design points in the relevant parameter domain. A better approach is a truncation based either on prior knowledge or a preliminary set of exploratory simulations. Parameters whose true values fall outside this prior truncation domain may then be recovered with ABC. The second problem is that the assumed form of the regression model (6), with the assumed normality of the residual distribution, restricts the posterior distribution to be Gaussian. For that reason, our GP regression should be seen as a (powerful!) step in the ABC-SMC pipeline rather than a replacement of it.

We have followed Fearnhead and Prangle [13] and made the NSPDE parameters the response variables of our GP regression, with the explanatory variables given by the features. ABC-SMC is then effectively used to relax the domain and distributional restrictions, as just discussed. As an alternative approach one could turn the features into the response variables of the GP regression, and use the NSPDE parameters as explanatory variables. This will effectively lead to a GP emulator Conti et al. [27], from which an objective function could be constructed, allowing for the stochasticity of the system in the vein of Andrieu and Roberts [5]. Parameter inference and uncertainty quantification could then be based on marrying this approach with a Hamiltonian Monte Carlo sampling approach, as proposed in Rasmussen [43]. Due to the need for repeated forward simulations from the NSPDEs for bias correction, this approach is unlikely to be computationally more efficient than the method proposed in the present paper. However, an implementation of this idea and a comparison of both methods in terms of accuracy and computational efficiency would be an interesting topic for future research.

We have applied the proposed method to a competition on parameter estimation in differential equations systems [37]. The data were generated from a NSPDE system describing chemotaxis, in which cell movement is as an emergent property based on advection-diffusion-reaction processes of various chemical compounds in the cell membrane, see Neilson et al. [2], Tweedy et al. [3]. Two datasets were made available as two separate challenges of the competition: *Fully Observed Data*, which include spatio-temporally varying compound concentrations, membrane configurations and cell locations, and *Partially Observed Data*, which only include cell locations and membrane configurations. Despite

the fact that we did not manage to get our large-scale ABC simulation study to converge by the submission deadline, our method came first in the competition. In subsequent post-competition work, we have continued the ABC simulations until convergence, to obtain a closer insight into the performance of the proposed methodology. Our simulations demonstrate the improvement obtained with GP regression over the linear regression method employed by Fearnhead and Prangle [13] (for obtaining summary statistics) and a quantification of the further improvement that can be achieved with ABC. They also allow an assessment of the parameter estimation performance that can be obtained for complex NSPDE systems with different levels of observability.

A future research challenge is to find a more systematic way of obtaining the features (as discussed in Section 3.1) to which the GP regression model is applied, and to find the optimal trade-off for the compact truncation of the parameter set that is used for training the GP regression model. Too large a domain will degrade the performance of the GP due to sparse coverage of the relevant (unknown!) subdomain; too small a domain will leave many true parameter values fall outside the domain covered by the GP, putting more strain on the ABC sampler for a subsequent correction.

The ultimate objective is to apply the proposed modelling and inference framework to real cell migration data, as e.g. obtained by high-resolution microscopy. Such applications would be of particular interest to cancer research, so as to shed more light on the mechanisms underlying metastasis. This will involve additional methodological challenges, though. First, microscopy studies give us time series of images in video format. Obtaining cell coordinates from these videos requires image segmentation and tracking techniques, either manual or automatic. Second, the NSPDE model of Tweedy [17], which was used in the competition, is expressed in dimensionless quantities. Linking the equations to real observations is not immediately feasible and requires appropriate variable transformations. Third, and most importantly, we lack a ground truth for real cell movement videos, rendering it difficult to assess the accuracy of our inference procedure. For that reason, we have restricted our analysis to simulated data, leaving the extension to real data to future work.

We are convinced that any such analysis will benefit from the study presented in our paper, which has explored the three essential steps for inference in complex systems with high-dimensional, stochastic outputs: feature extraction for dimension reduction, GP regression for feature weighting and fast parameter estimation, and an ABC-SMC scheme for refined inference and approximate Bayesian uncertainty quantification.

Nomenclature – most important symbols

Outputs from the NSPDE model (1)–(5):

- X, Y (correspond to $[\Gamma(t)]_{t=1}^T$) – time series of space coordinates of the cell membrane;
- LA (corresponds to $[a(\gamma, t)]_{t=1, \gamma=0}^{T, |\Gamma(t)|}$) – time series of local activator values;

LI (corresponds to $[b(\gamma, t)]_{t=1, \gamma=0}^{T, |\Gamma(t)|}$) – time series of local inhibitor values;

S (corresponds to $[s(\gamma, t)]_{t=1, \gamma=0}^{T, |\Gamma(t)|}$) – time series of the strength of the chemosensation stimulus.

Indexing convention:

$i = 1, \dots, 10$ for the 10 NSPDE parameters and the related concepts;

$j = 1, \dots, M$ for the elements of the calibration sample and the corresponding outputs;

$d = 1, \dots, D$ for the elements of the extracted feature vectors (hence D is the GP kernel dimension);

$n = 1, \dots, N$ for the ABC-SMC particles;

$c = 1, \dots, C$ for the intermediary ABC-SMC distributions;

$t = 1, \dots, T = 1000$ for the time dimension of the NSPDE simulator output;

$s = 1, \dots, S$ for the GP kernel specification.

Selected variables:

$\boldsymbol{\theta} = [\theta_i]_{i=1}^{10} = (f_a, r_c, k_b, d_b, D_b, k_M, s_a, b_a, D_a, d_a)^T$ – the vector of parameters of the NSPDE model (1)–(5);

$\tilde{\boldsymbol{\theta}}$ – the default value of $\boldsymbol{\theta}$ from Tweedy [17];

\mathbf{y} – observed output from the NSPDE simulator (e.g. X, Y, LA, LI and S for *Fully Observed Data*);

\mathbf{x} – (low dimensional) vector of features extracted from \mathbf{y} ;

$\boldsymbol{\alpha}$ – a chemical signal (LA, LI or S, or a transformation of one of them).

GP regression (6):

$\mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}^T))$ – Gaussian process with the mean function m and the variance function k (called kernel);

$\mathbf{X} = [\mathbf{x}^{(j)}]_{j=1}^M$ – feature matrix (collecting \mathbf{x} s from the calibration sample);

$\mathbf{K} = k(\mathbf{X}, \mathbf{X})$ – covariance of the training calibration sample;

$\boldsymbol{\Theta}_i = [\theta_i^{(j)}]_{j=1}^M$ – vector of the i th NSPDE parameter collected from the calibration sample;

$f(\mathbf{x})$ – the values of the latent process at \mathbf{x} ;

$\mathbf{f} = f(\mathbf{X}) = [f(\mathbf{x})^{(j)}]_{j=1}^M$ – latent values for the observed sample \mathbf{X} ;

$\boldsymbol{\phi}$ – vector of the GP regression model (6) hyperparameters.

ABC:

π – intractable posterior distribution;

π_{ABC} – ABC approximation to π ;
 ρ, ϵ – distance metric, tolerance level;
 S – summary statistic (summarising \mathbf{x} into a scalar);
 $\boldsymbol{\theta}^*, \mathbf{y}^*, \mathbf{x}^*$ – proposed $\boldsymbol{\theta}$ and corresponding \mathbf{y} and \mathbf{x} ;
 $w_n^{(c)}$ ($W_n^{(c)}$) – the unnormalised (normalised) weight of the n th particle in the c th intermediary distribution.

Acknowledgments

This work was funded by the UK Engineering and Physical Sciences Research Council (EPSRC), grant numbers EP/N014642/1 and EP/R018634/1. Dirk Husmeier is supported by a grant from the Royal Society of Edinburgh, award number 62335.

References

- [1] T. Jin, D. Hereld, *Chemotaxis: Methods and Protocols*, Springer, 2016.
- [2] M. P. Neilson, D. M. Veltman, P. J. M. van Haastert, S. D. Webb, J. A. Mackenzie, R. H. Insall, Chemotaxis: a Feedback-Based Computational Model Robustly Predicts Multiple Aspects of Real Cell Behaviour, *PLoS Biology* 9 (2011) 1–11.
- [3] L. Tweedy, B. Meier, J. Stephan, D. Heinrich, R. G. Endres, Distinct cell shapes determine accurate chemotaxis, *Scientific Reports* 3 (2013) 2606.
- [4] G. MacDonald, J. A. Mackenzie, M. Nolan, R. H. Insall, A Computational Method for the Coupled Solution of Reaction–Diffusion Equations on Evolving Domains and Manifolds: Application to a Model of Cell Migration and Chemotaxis, *Journal of Computational Physics* 309 (2016) 207–226.
- [5] C. Andrieu, G. Roberts, The Pseudo-Marginal Approach for Efficient Monte Carlo Computations, *Annals of Statistics* 37 (2009) 697–725.
- [6] M. Beaumont, Estimation of Population Growth or Decline in Genetically Monitored Populations, *Genetics* 164 (2003) 1139–1160.
- [7] C. Andrieu, A. Doucet, R. Holenstein, Particle Markov Chain Monte Carlo Methods, *Journal of the Royal Statistical Society Series B* 72 (2010) 269–342.
- [8] G. Deligiannidis, A. A. Doucet, M. K. Pitt, The Correlated Pseudomarginal Method, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 80 (2018) 839–870.

- [9] M. A. Beaumont, J. M. Cornuet, J. M. Marin, C. P. Robert, Adaptive Approximate Bayesian Computation, *Biometrika* 96 (2009) 983–990.
- [10] S. N. Wood, Statistical Inference for Noisy Nonlinear Ecological Dynamic Systems, *Nature* 466 (2010) 1102.
- [11] L. F. Price, C. C. Drovandi, A. Lee, D. J. Nott, Bayesian Synthetic Likelihood, *Journal of Computational and Graphical Statistics* 27 (2018) 1–11.
- [12] J. Liepe, H. Taylor, C. P. Barnes, M. Huvet, L. Bugeon, T. Thorne, J. R. Lamb, M. J. Dallman, M. P. H. Stumpf, Calibrating Spatio-temporal Models of Leukocyte Dynamics Against in vivo Live-imaging Data Using Approximate Bayesian Computation, *Integrative Biology* 4 (2012) 335–345.
- [13] P. Fearnhead, D. Prangle, Constructing Summary Statistics for Approximate Bayesian Computation: Semi-Automatic Approximate Bayesian Computation, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 74 (2012) 419–474.
- [14] C. E. Rasmussen, C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.
- [15] R. Wilkinson, Accelerating ABC Methods using Gaussian Processes, in: *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, PMLR, 2014, pp. 1015–1023.
- [16] P. S. Craig, M. Goldstein, A. H. Seheult, J. A. Smith, Pressure Matching for Hydrocarbon Reservoirs: a Case Study in the use of Bayes Linear Strategies for Large Computer Experiments, in: *Case Studies in Bayesian Statistics*, Springer, 1997, pp. 37–93.
- [17] L. Tweedy, Meinhardt Simulations on an Evolving Line, Technical Report, Cside 2018 Supplementary Notes, 2018.
- [18] H. Meinhardt, Orientation of Chemotactic Cells and Growth Cones: Models and Mechanisms, *Journal of Cell Science* 112 (1999) 2867–2874.
- [19] T. Jin, N. Zhang, Y. Long, C. A. Parent, P. N. Devreotes, Localization of the g protein $\beta\gamma$ complex in living cells during chemotaxis, *Science* 287 (2000) 1034–1036.
- [20] G. Dziuk, C. M. Elliott, Finite elements on evolving surfaces, *IMA journal of numerical analysis* 27 (2007) 262–292.
- [21] P. J. Van Haastert, L. Bosgraaf, The local cell curvature guides pseudopodia towards chemoattractants, *HFSP journal* 3 (2009) 282–286.
- [22] M. P. Neilson, J. A. Mackenzie, S. D. Webb, R. H. Insall, Modeling cell movement and chemotaxis using pseudopod-based feedback, *SIAM Journal on Scientific Computing* 33 (2011) 1035–1057.

- [23] J. C. Gower, G. B. Dijksterhuis, *Procrustes Problems*, volume 30, Oxford University Press, 2004.
- [24] I. L. Dryden, K. V. Mardia, *Statistical Shape Analysis, with Applications in R. Second Edition*, John Wiley and Sons, Chichester, 2016.
- [25] I. Mariñas del Collado, *Statistical Models for the Evolution of Facial Curves*, Ph.D. thesis, University of Glasgow, 2017.
- [26] V. Bonhomme, S. Picq, C. Gaucherel, J. Claude, Momocs: Outline Analysis Using R, *Journal of Statistical Software* 56 (2014) 1–24.
- [27] S. Conti, J. P. Gosling, J. E. Oakley, A. O’Hagan, Gaussian Process Emulation of Dynamic Computer Codes, *Biometrika* 96 (2009) 663–676.
- [28] S. Conti, A. O’Hagan, Bayesian Emulation of Complex Multi-output and Dynamic Computer Models, *Journal of Statistical Planning and Inference* 140 (2010) 640–651.
- [29] J. Requeima, W. Tebbutt, W. Bruinsma, R. E. Turner, The gaussian process autoregressive regression model (gpar), in: *The 22nd International Conference on Artificial Intelligence and Statistics*, 2019, pp. 1860–1869.
- [30] R. M. Neal, *Bayesian Learning for Neural Networks*, volume 118, lecture notes in statistics 118 ed., Springer, 2012.
- [31] J. Pritchard, M. T. Seielstad, A. Perez-Lezaun, M. W. Feldman, Population Growth of Human Y Chromosomes: a Study of Y Chromosome Microsatellites, *Molecular Biology and Evolution* 16 (1999) 1791–1798.
- [32] P. Marjoram, J. Molitor, V. Plagnol, S. Tavaré, Markov Chain Monte Carlo Without Likelihoods, *Proceedings of the National Academy of Sciences of the United States of America* 100 (2003) 15324–15328.
- [33] M. A. Beaumont, W. Zhang, D. J. Balding, Approximate Bayesian Computation in Population Genetics, *Genetics* 162 (2002) 2025–2035.
- [34] S. A. Sisson, Y. Fan, M. Beaumont, *Handbook of Approximate Bayesian Computation*, Chapman and Hall/CRC, 2018.
- [35] A. Doucet, N. D. Freitas, N. Gordon, An Introduction to Sequential Monte Carlo Methods, in: *Sequential Monte Carlo Methods in Practice*, Springer, 2001, pp. 3–14.
- [36] T. Toni, D. Welch, N. Strelkova, A. Ipsen, M. P. H. Stumpf, Approximate Bayesian Computation Scheme for Parameter Inference and Model Selection in Dynamical Systems, *Journal of the Royal Society Interface* 6 (2009) 187–202.

- [37] Cside, Competitive Statistical Inference for Differential Equations, 2018. <https://www.gla.ac.uk/schools/mathematicsstatistics/events/conferences/cside2018/>, Last accessed on 2018-12-17.
- [38] D. R. Cox, N. Reid, The Theory of the Design of Experiments, Chapman and Hall/CRC, 2000.
- [39] I. M. Sobol', On the distribution of points in a cube and the approximate evaluation of integrals, Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki 7 (1967) 784–802.
- [40] M. D. McKay, R. J. Beckman, W. J. Conover, Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code, Technometrics 21 (1979) 239–245.
- [41] Tibshirani, Robert, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society: Series B (Methodological) 58 (1996) 267–288.
- [42] F. Jabot, T. Faure, N. Dumoulin, EasyABC: Performing Efficient Approximate Bayesian Computation Sampling Schemes using R, Methods in Ecology and Evolution 4 (2013) 684–687.
- [43] C. E. Rasmussen, Gaussian Processes to Speed up Hybrid Monte Carlo for Expensive Bayesian Integrals, Bayesian Statistics (2003) 651–659.