

Competitive statistical inference for differential equations: a report on Cside 2018

Benn Macdonald, Muhamad Hifzhudin Noor Aziz,
Radostin Simitev, Edward Acheampong, Chon Lok Lei,
Michael Clerx, Gary R. Mirams, Mitchel J. Colebank,
M. Umar Qureshi, Mette S. Olufsen, James Ritchie, Iain Murray,
Luke Tweedy, Mihaela Paun, Alan Lazarus, Umberto Noè,
Dirk Husmeier, Agnieszka Borowska and Diana Giurghita

Differential equations provide a powerful way of modelling an observed system by providing it with a mathematical description. Examples of the systems differential equations model range from predator-prey interactions in ecosystems (see Lotka (1932) [26]), fluid mechanics where systems can be chaotic (i.e. slightly differing initial states can evolve into considerably different states, for example Lorenz (1963) [25] and Strebel (2013) [40]), diffusion - which can be used to describe the mobility of molecules (e.g. Fick (1995) [13]), activation/deactivation of spiking neurons (see FitzHugh (1961) [15]), financial market prices in econometrics (for example Alexander et al. (2012) [1]) and particle physics (for example Tomé and Oliveira (2015) [42]). However, the act of inferring the parameters that govern the equations from noisy data, especially in a timely fashion, is challenging.

A typical approach to assessing how well these methods perform is to generate synthetic data by solving the equations once and adding noise to reflect observational error. In this way, since the parameters that generated the data are known, one can quantify how well the method performed, subject to some assessment criteria. However, by using simulated data we usually retain knowledge of the system that wouldn't be available from a real experiment, which may distort our assessment to how a particular method performs when applied in practice.

Cside (Competitive statistical inference for differential equations) is a competitive framework for comparing methods that infer the parameters of differential equations. It challenges participants to infer the parameters from systems described by differential equations in a blinded fashion and thereby more fairly judge the performance of a method. Since participants were competing against one another, this acted as a method of filtering out approaches that would perform more poorly. In this way, we are able to gain an understanding as to how well the current state-of-the-art performs when dealing with these types of systems, providing guidance as to how to tackle these problems in practice.

Participants were first given access to three biological models described by differential equa-

tions, along with code and instructions for generating test datasets. They had two months to familiarise themselves with the systems, test/configure their methods, before competition data was released. Each participant then received a dataset generated from their chosen model/s, where the parameter generating values were known only to the event organiser. Some information about the systems were made available (see Section 2 for details) in order to mimic experimental and prior information that might be available from experts in the field. Participants had one and a half weeks to submit their parameter estimates. This amount of time was chosen to ensure there was sufficient challenge and similarity with respect to the time constraints present when applying these methods in practice, whilst also taking into account entrants’ different schedules in order to maximise engagement with the event and see/analyse as many different methods as possible.

The structure of this paper is as follows. In Section 1, a description of the biological models that were used during the challenge will be provided. Participants had access to a set of Ordinary Differential Equations (ODEs) that describe cardiac electrical excitation, a Partial Differential Equation (PDE) system describing blood pressure and flow in the pulmonary arteries and a set of Stochastic Differential Equations (SDEs) describing the behaviour of cells migrating as a response to chemotaxis, based on a pseudopod-centred mechanism. By including an ODE, PDE and SDE system, we hope to obtain results that can inform how to approach analysis in practice in the wider research area where systems are described by differential equations. Section 2 describes the simulation set-up of the problems posed to participants. Parameter value choices, levels of observational noise, as well as which variables are observable, can all be found in this section. Sections 3 - 4 contain the methodological descriptions and results, respectively, for the qualifying methods (first three participants for models 1 and 2 and first two participants for model 3 and the additional challenge).

1 Models

1.1 Model 1: A model of cardiac electrical excitation

Model 1 is a simplified model of the electrical potential of cardiac cell membranes [5]. Cardiac cell membranes are composed of a biphospholipid layer protruded by voltage-gated ion channels, e.g. see [8]. The layer is impermeable to charged particles thus supporting a non-zero equilibrium voltage potential across the membrane. The ion channels are large proteins that open and close depending on the instantaneous value of the voltage and allow in/outflow of ion currents. When “excited”, i.e. driven away from equilibrium, these structures cause the formation of a large transmembrane voltage transient known as an action potential. The action potential serves as a signal for cardiac cells to contract and thus controls the heartbeat – the main function of a living heart. To a first approximation the cell membrane is modelled as an electrical circuit of a capacitor C_m (the biphospholipid layer) and an active resistor supporting ionic currents I_{ion} (the ionic channels) connected in parallel, giving the ordinary differential equation $C_m \dot{E} = I_{\text{ion}}$ for the voltage $E(t)$ [24]. The electrophysiological complexity of the cardiac membrane is encapsulated by the models of the ionic current I_{ion} . The current is due to a large number of ionic currents each flowing through their own dedicated population of channels with distinct dynamics. Along with ion channels, pumps and exchangers also transport ions across the membrane using active processes rather than passive diffusion. Over the last 60 years, models have been developed for

the action potential of various types of cardiac cells, various animal species, and for healthy and pathological situations, with many contemporary models being very large systems of ordinary differential equations including hundreds of parameters to be determined experimentally [37].

Despite their differences and complexity most models of cardiac action potentials have similar underlying mathematical structure [6]. Model 1 is a simplified model in the sense that it preserves the typical asymptotics of cardiac equations and thereby captures essential aspects of cell AP models. The model is obtained from the Noble equations for Purkinje fibers [30] using a set of verifiable simplification steps [5]. It has been used to derive asymptotic expressions for the conduction velocity restitution in cardiac tissues [39], to elucidate the conditions for propagation and block of excitation in the atria [38], to understand the formation of excitation waves [4], and to study cellular alternans. The model contains 3 state variables, the transmembrane voltage potential $E(t)$, and the gating variables $h(t)$ and $n(t)$ describing the inactivation of the fast inward and slow outward currents, respectively, coupled by the non-linear ordinary differential equations,

$$\frac{dE}{dt} = \frac{1}{\epsilon_1 \epsilon_2} G_{Na}(E_{Na} - E)H(E - E_*)h + \frac{1}{\epsilon_2} (g_2(E)n^4 + G(E)), \quad (1a)$$

$$\frac{dh}{dt} = \frac{1}{\epsilon_1 \epsilon_2} F_h(H(E_{\dagger} - E) - h), \quad (1b)$$

$$\frac{dn}{dt} = F_n(E)(H(E - E_{\dagger}) - n). \quad (1c)$$

Here $H(\cdot)$ denotes the Heaviside step function and the rest of the functions prescribing the dynamics are given by

$$\begin{aligned} g_2(E) &= g_{21}H(E_{\dagger} - E) + g_{22}H(E - E_{\dagger}), \\ F_n(E) &= f_n(rH(E_{\dagger} - E) + H(E - E_{\dagger})), \\ E_2 &= (k_1/k_2 + 1)E_{\dagger} - E_1 k_1/k_2, \\ E_3 &= (k_2/k_3 + 1)E_* - E_2 k_2/k_3, \end{aligned}$$

$$G(E) = \begin{cases} k_1(E_1 - E), & E \in (-\infty, E_{\dagger}), \\ k_2(E - E_2), & E \in [E_{\dagger}, E_*), \\ k_3(E_3 - E), & E \in [E_*, +\infty). \end{cases}$$

The model contains 12 parameters,

$$\boldsymbol{\theta} = [k_1, k_2, k_3, E_1, E_{Na}, E_{\dagger}, E_*, F_h, f_n, G_{Na}, g_{21}, g_{22}]^T.$$

At the default parameter values used in [39], the following initial conditions lead to a fully-developed action potential

$$E(0) = E_{\text{stim}} \geq E_*, \quad h(0) = 1, \quad n(0) = 0. \quad (2)$$

To understand the behaviour of the solution and the natural parameter ranges it is useful to consider the asymptotic phase portrait of the model shown in Figure 1 (also see [5]). Equations (1) contain two small parameters formally ordered as $0 \ll \epsilon_1 \ll \epsilon_2 \ll 1$. In the limit of $\epsilon_1 \rightarrow 0^+$

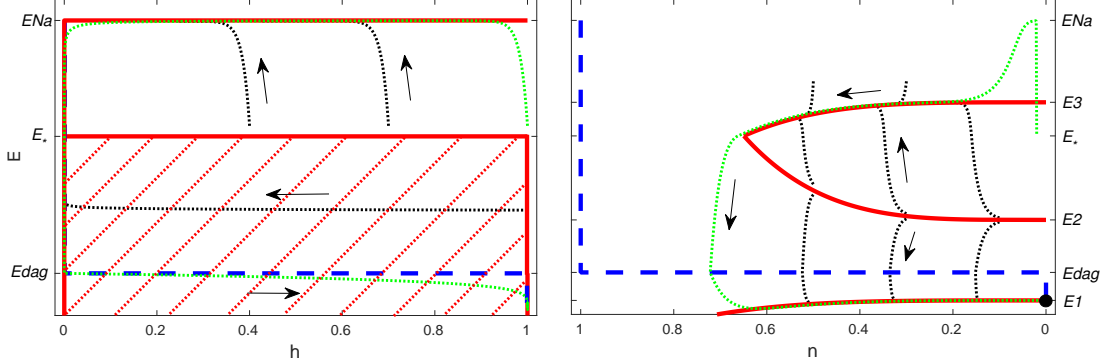


Figure 1: The phase portrait of the fast system (left panel) and the slow system (right panel) of caricature model. The blue dashed line is the nullcline of $h(t)$ in the fast system and $n(t)$ in the slow system. The solid red lines and the cross-hatched region in the left and right panels are the horizontal nullclines of $E(t)$. The dotted black lines with attached arrows are the trajectories. The dotted green line is the corresponding action potential, $E_{dag} = E_{\dagger}$.

a super-fast system is obtained with a phase portrait shown in the left panel of Figure 1. If $E_{stim} \geq E_*$, then the super-fast subsystem is activated and generates a fast upstroke. If the E_{stim} is smaller than this threshold value, then the dynamics is governed only by the slow subsystem. In the limit $\epsilon_2 \rightarrow 0^+$ the slow system splits further into a fast and a slow part with phase portraits shown in the right panel of Figure 1. It features a slow manifold divided into three parts, a ‘systolic’ branch, ‘diastolic’ branch, and a threshold branch. When E_{stim} is above the threshold branch the trajectory will be attracted to the upper systolic branch and will slowly follow it until it ends, it will then make a fast jump to the diastolic branch and follow this towards global equilibrium to complete one action potential transient. For the purposes of the event, ϵ_1 and ϵ_2 were held fixed at the value 1, recovering the original system of Biktashev et al. [5].

Numerical codes in various languages (Matlab, C++, Fortran and Python) for this model are available on GitHub¹.

Model 1 poses a numerically stiff problem. We recommend setting the relative tolerance to 10^{-5} when using an adaptive time step solver to solve Equations (1). Setting larger tolerances leads to undesired effects on the likelihood landscape, as demonstrated in Figure 2.

1.2 Model 2: A 1D fluid dynamics model of the pulmonary circulation

Model 2 is a one-dimensional (1D) fluid dynamics model of pulse wave propagation in a branching network of compliant blood vessels [32]. For this study, the model is tailored to predict time varying blood pressure and flow at any location in a 21-vessel pulmonary arterial tree for a male C57BL6/J control mouse (see Fig. 3) [11, 35]. Each vessel within the network is modelled

¹<https://github.com/hifzhudin/CaricNobleModel>

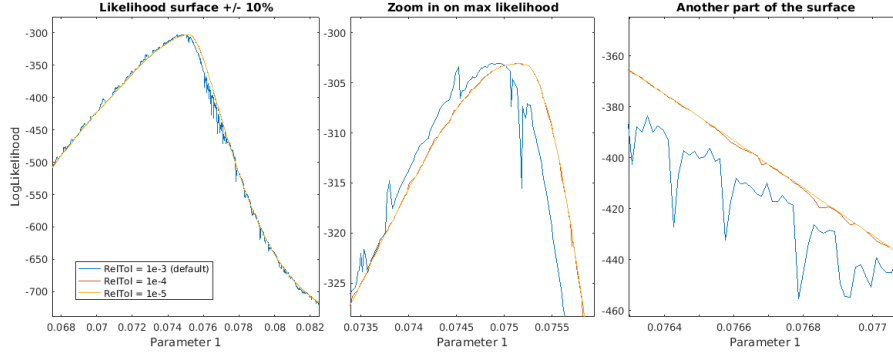


Figure 2: Effect of ODE solver tolerance setting in the likelihood surface. Note that a coarse tolerance level (relative tolerance 10^{-3}) can introduce bumps, spikes and many local maxima in the likelihood surface and shift the peak of the global maximum likelihood. A more refined solution (relative tolerance 10^{-5}) is required to give an accurate likelihood surface.

as a thin-walled, linearly elastic, axisymmetric surface of revolution having constant length L and dynamic radius R . Geometric properties including vessel dimensions (i.e. reference radius r_0 (cm) and length L (cm)) and network connectivity are extracted from a micro-computed tomography (micro-CT) image of an excised lung [35], whereas dynamic flow and pressure are measured in-vivo. The model is governed by a hyperbolic system of non-linear partial differential equations (PDEs) formulated in cylindrical polar coordinates (r, θ, x) , obtained by averaging the continuity and Navier-Stokes equations over each vessel's cross-sectional area, giving

$$\frac{\partial A}{\partial t} + \frac{\partial q}{\partial x} = 0, \quad \frac{\partial q}{\partial t} + \frac{\partial}{\partial x} \left(\frac{q^2}{A} \right) + \frac{A}{\rho} \frac{\partial p}{\partial x} = -\frac{2\pi\nu r}{\delta} \frac{q}{A}, \quad (3)$$

where $0 \leq x \leq L$ and $0 \leq t \leq T$ denote the axial and temporal coordinates with T being the length of the cardiac cycle (assumed constant, $T = 0.11$ s), $p(x, t)$ denotes the transmural blood pressure (mmHg), $q(x, t)$ is the volumetric flow rate (ml/s), $A(x, t) = \pi R(x, t)^2$ (cm²) is the vessel cross-sectional area, and $R(x, t)$ (cm) is the radius. The blood density $\rho = 1.057$ (g/ml), the kinematic viscosity $\nu = 0.0462$ (cm²/s), and the boundary layer thickness $\delta = 0.03$ (cm) are assumed constant. To close the system of equations (3), we prescribe a so-called tube-law, a mathematical relation between the cross-sectional area A and the transmural pressure p . By assuming a linear stress-strain relation between the vessel wall and the transmural pressure, we can model the spatial pressure in a given vessel as

$$p - p_0 = Eh/r_0 \left(\sqrt{A/A_0} - 1 \right), \quad (4)$$

where $p_0 = 0$ is the constant external pressure acting on the vessel wall, $A_0 = \pi r_0^2$ is the vessel cross-sectional area at the reference pressure ($p = p_0$), E is the Young's modulus (mmHg) and h is the wall thickness (cm) such that $h \ll r_0$. Assuming a constant E and h/r_0 throughout the network, the ratio Eh/r_0 is considered one parameter that describes the vessel wall stiffness.

Numerical integration of (3), coupled with (4), requires boundary conditions at the network inlet, outlets, and each vessel junction (i.e. each vessel bifurcation). We impose a measured

inflow $q_{in}(t)$ at the inlet of the first vessel and assume conservation of flow and continuity of pressure at each bifurcation, i.e.

$$q_p(L, t) = q_{d_1}(0, t) + q_{d_2}(0, t), \quad p_p(L, t) = p_{d_1}(0, t) = p_{d_2}(0, t), \quad (5)$$

where the subscript p denotes the parent vessel and the subscripts d_1, d_2 represent the two daughter vessels. The outflow boundary conditions at the outlet of each terminal vessel are imposed via a three-element Windkessel (3-WK) [35], analogous to an electrical circuit with two resistors in series and a capacitor in parallel. The 3-WK boundary condition is modelled by the first order ODE

$$\frac{dp}{dt} - R_1 \frac{dq}{dt} = q \left(\frac{R_1 + R_2}{R_2 C_T} \right) - \frac{p}{R_2 C_T}, \quad (6)$$

where R_1 and R_2 denote proximal and distal resistances (mmHg s/ml) of the vascular region beyond the point of truncation. The sum $R_1 + R_2$ gives the total vascular resistance R_T , and C_T denotes the total peripheral compliance (ml/mmHg) of the same region. Finally, we assume that predictions are periodic, that is the pressure and flow are repeated at the onset of each cardiac cycle. Details of how the model parameters are initialised and optimised can be found in [11] and [35].

In this study, we wish to infer the vessel stiffness (Eh/r_0). To make the problem computationally tractable, we assume that stiffness is constant throughout all 21 vessels. In addition, we consider the use of the scaling factors r_1, r_2 and c , which scale the nominal values of the 3-WK parameters (i.e. $\hat{R}_1 = r_1 R_1^{nom}$, $\hat{R}_2 = r_2 R_2^{nom}$, and $\hat{C}_T = c C_T^{nom}$). For more details on this method, see [11]. For this problem, the scaling factors are set to $r_1 = 2.0193 \times 10^{-1}$, $r_2 = 8.8890 \times 10^{-1}$, and $c = 1.4665 \times 10^0$. More information about the calculation of nominal values for this model and the use of scaling factors can be found in [11, 35].

Variable	Meaning	Value	Units
p_0	reference pressure	0	mmHg
r_0	reference radius	vessel dependent	cm
ν	kinematic viscosity	0.0462	cm ² /s
δ	boundary layer thickness	0.03	cm
ρ	blood density	1.057	g/ml
T	heart rate	0.11	s
E	Young's modulus	estimated	mmHg
h	wall thickness	estimated	cm
R_1^i	proximal resistance for terminal vessel i	vessel dependent	mmHg s/ml
R_2^i	distal resistance for terminal vessel i	vessel dependent	mmHg s/ml
C_T^i	compliance for terminal vessel i	vessel dependent	ml/mmHg
r_1	proximal resistance scaling factor	2.0193×10^{-1}	dimensionless
r_2	distal resistance scaling factor	8.8890×10^{-1}	dimensionless
c_1	compliance scaling factor	1.4465	dimensionless

Table 1: Variables of System (3)–(4) for Model 2 [43].

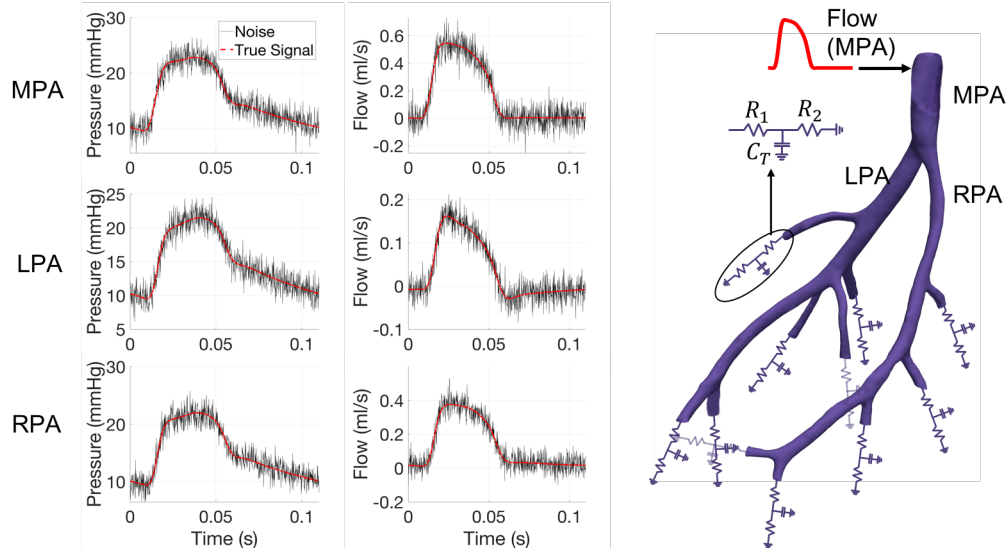


Figure 3: (Left) True model predictions (red curves) along with noise induced signal (black curves) in the main, left, and right pulmonary arteries (MPA, LPA, and RPA, respectively); (Right) 21 vessel network from a control mouse with 3 element Windkessel models attached to each terminal vessel. Flow data in the MPA is used as an input to the model.

1.3 Model 3: A pseudopod-centred model of eukaryotic chemotaxis

Model 3 is representative of the pseudopod-centred view of single cell chemotaxis [2], in which directional decision-making is an emergent property of competition between multiple membrane patches demonstrating cytoskeletal activity. The central idea is that some auto-activating molecule (or system of molecules) referred to as the local activator (LA) drives the membrane in the outward normal direction, and that its accumulation is biased by external chemoattractant signals. In addition to the local activator, a local inhibitor (LI) destabilises dominant patches of activator to encourage dynamic reorientation to changing signals, and a fast-diffusing, global inhibitor (GI) allows distal regions of activator to compete. Finally, tension retracts the boundary normally to maintain a preferred cell size, allowing the cell to move rather than simply expand (see Fig. 4 for example images of this model as it evolves). The system evolves according to the following equations:

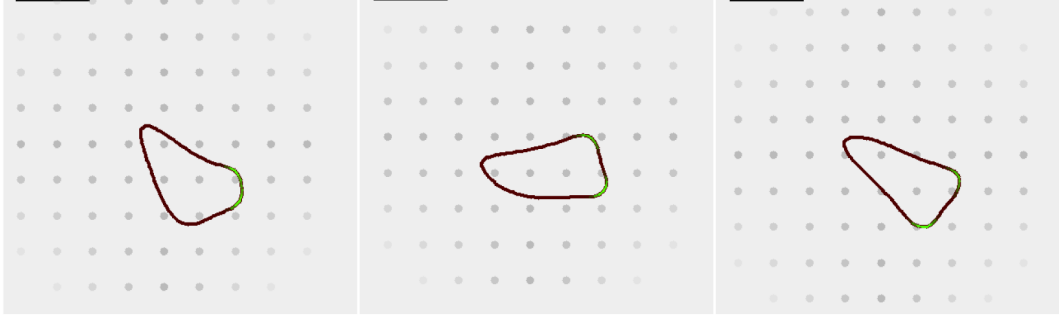


Figure 4: **Example frames from model 3 simulation (A)** Three stills of a running simulation are shown. The strength of the local activator signal is displayed around the perimeter in green, and the value of the global inhibitor is displayed in red. The membrane changes shape in response to these signals, with protusions clearly visible in areas with high concentrations of the local activator.

$$\partial_t a = D_a \Delta_\Gamma a + \frac{s(a^2/c + b_a)}{(k_M + b)(1 + s_a^2)} - d_a a, \quad (7)$$

$$\partial_t b = D_b \Delta_\Gamma b + k_b a - d_b b, \quad (8)$$

$$\partial_t c = \frac{r_c}{|\Gamma|} \oint_\Gamma a d\mathbf{x} - r_c a, \quad (9)$$

$$s(\mathbf{x}, t) = 1 + d_r \text{RND} + \left(\frac{C(\mathbf{x}, t)}{C(\mathbf{x}, t) + k_d} \right) (1 + d_r \text{RND}), \quad (10)$$

$$\partial_t \Gamma = (f_a a(\mathbf{x}, t) - \lambda(A - A_0)) \hat{\Gamma}(\mathbf{x}, t), \quad (11)$$

where (7), (8) and (9) describe the evolution of the local activator (LA), local inhibitor (LI) and global inhibitor (GI), respectively, (10) represents the stimulus strength in terms of external concentration C at point \mathbf{x} and time t , while (11) defines the evolution of the boundary line $\Gamma(\mathbf{x}, t)$ of the cell, with $\hat{\Gamma}(\mathbf{x}, t)$ denoting the outward normal of the boundary and $|\Gamma|$ being the total length of the boundary. RND in (10) is a noise term introduced to break the symmetry of the system to allow for pseudopod formation, uniformly drawn from $(0, 1)$, with d_r controlling its magnitude. For a more complete discussion of this model, see [29].

Simulations implementing this model output time series for LA, LI, GI and membrane shape (a series of x, y coordinates for sample points on the membrane). The aim is to infer from these time series the values of 10 parameters:

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

The remaining parameters of the system are treated as fixed. For a full table of parameters see Table 2.

The key challenge of model 3 is that it is an SDE and so evolves stochastically. Repeat simulations with identical parameters can therefore lead to quite different results. In practical

experiments, there are limitations on observing the biochemistry of the cell. In contrast, its boundary is readily observable using light microscopy. As such, the additional challenge for model 3 is to infer the parameters from the boundary coordinates alone, reflecting the initial role of this simulation in explaining cell shape as a response to external chemotactic factors [44].

Parameter	Meaning	Default value $\tilde{\theta}$
f_a	rate of the outward force from LA	0.0015
r_c	response speed of GI	0.07
k_b	birth rate of LI	0.0028
d_b	death rate of LI	0.013
D_b	diffusivity of LI	0.045
k_M	Michaelis-Menten-like constant for LI	0.16
s_a	variable controlling saturation of LA	7.0E-5
b_a	basal production level of LA	0.1
D_a	diffusivity of LA	0.025
d_a	the death rate of LA	0.02

Table 2: Parameters of System (7)–(11) to be inferred for Model 3 and Additional Challenge as specified by [43].

2 Simulation

2.1 Model 1 (cardiac electrical excitation)

There are three observed variables; E (the transmembrane voltage), h (a gating variable) and n (another gating variable), and 12 parameters to infer; $k_1, k_2, k_3, E_1, E_{Na}, E_{\dagger}, E_*, F_h, F_n, G_{Na}, g_{21}$ and g_{22} . Recall that ϵ_1 and ϵ_2 were kept fixed at the value 1 for this event to recover the system as outlined in Biktashev et al. [5].

When solving the system of equations using the parameter values defined in Simatev and Biktashev [39] and ode15s in Matlab, 205 points are chosen by the solver (in order to ensure there is enough resolution over the course of the signal). Changing the parameters would cause the solver to choose a new spacing for the time points, which is unsuitable for scoring in this event (since participants would have different time courses/number of time points associated with their results). Hence, 205 points were simulated over the time domain [0,900], with 150 points linearly spaced between [0,300] to ensure there was enough coverage in the region where the signals exhibited high variance and the remaining 55 points linearly space between [301,900] where the signals plateaued. This time course was kept fixed for the event.

The parameter values were chosen to between 0.5 and 4 times literature values taken from Simatev and Biktashev [39]. This ensured that the simulated data consisted of signals that were biologically plausible, but different enough to the those produced by Simatev and Biktashev [39] (which was highlighted to the participants as a resource for the model).

Note that the default tolerance level of $1e^{-3}$ in Matlab for solving the ODEs (using ode15s) was too large for this model, causing numerical error in the solution to be large. The subsequent likelihood landscape would cause optimisers/samplers to perform poorly. To avoid this, a tolerance level of $1e^{-5}$ was used to generate the event data. Once data were generated for E, h and

Parameter	Value	Parameter	Value
k_1	0.12	k_2	0.1
k_3	0.3	E_1	-65
E_{Na}	60	E_{\dagger}	-60
E_{\star}	-20	F_h	0.25
F_n	0.015	G_{Na}	50
g_{21}	-7	g_{22}	-5

Table 3: Data generating parameters for Model 1 (coupled ODEs describing cardiac electrical excitation).

n, a signal to noise ratio (SNR = variance(signal)/variance(noise)) of 10 was used to corrupt the output with Gaussian white noise, so as to imitate experimental error.

2.2 Model 2 (vessel stiffness in pulmonary circulation)

There are two observed variables, p (blood pressure) and q (blood flow), and one parameter to infer ($\frac{Eh}{r_0}$, the vessel stiffness). Although there are many branches in the pulmonary arterial tree, the model predicts quantities of interest in the first 21 arteries. In order to ensure that inference for this system was feasible in the time frame of the event, observations were constrained to the first three vessels. The system is observed over time (t) and space (x) (recall that Model 2 is a PDE system). The time points were uniformly spaced between 0 and T , where T is the length of the cardiac cycle (assumed constant at $T = 0.11$ s), in increments of 1.34×10^{-5} s (to 2dp). The spatial points were uniformly spaced in increments of 2.5×10^{-2} cm between 0 and L , where L is the vessel length and changes depending on which vessel is considered (0.41cm, 0.445cm and 0.372cm for vessels 1-3 respectively). From these spatial points, one spatial location is considered for each vessel (the midpoint of the discretised values of x for each vessel respectively). This resulted in 8192 datapoints for each vessel which was subsampled by taking every 8th point to match the dimensionality (1024 points) of real data, see Vanderpool et al. [45], Tabima et al. [41] and Colebank et al. [11] for details. This approach ensured convergence of the numerical methods used for solving the PDEs, whilst mimicking real-world data experiments.

The range of values for the parameter $\frac{Eh}{r_0}$ that result in a valid PDE solution is $[10^4, 10^5]$ and hence the data generating parameter was chosen within this space.

Parameter	Value
Eh/r_0	75000

Table 4: Data generating parameters for Model 2 (PDEs describing vessel stiffness in pulmonary circulation).

Once data were generated for p and q, an SNR of 10 was used to corrupt the output with Gaussian white noise, so as to imitate experimental error.

2.3 Model 3 (eukaryotic chemotaxis)

2.3.1 Fully observable model

Since model 3 is a Stochastic Differential Equation (modelling cells migrating as a response to chemotaxis, based on a pseudopod-centred mechanism), inferring the parameters of this system is very challenging. In order to ensure that the parameters could be inferred during the time constraints imposed by the event, an assumption was made that all output from this model is measurable. Therefore the observed variables are the global inhibitor (GI), local inhibitor (LI), local activator (LA), stimulus strength, x co-ordinate and y co-ordinate of each finite element node of the boundary line. As it is unlikely to observe the GI, LI, LA and stimulus strength in practice, an additional challenge for this model was posed, where only the x and y co-ordinates are observed. Please see Section 2.3.2 for more details on this additional challenge.

There are 10 parameters to be inferred; f_a , r_c , k_b , d_b , D_b , k_M , s_a , b_a , D_a and d_a . The units (u) for time are arbitrary and time increments with a step size of 0.1u between [0.1u,100000u]. The code samples every 100u (evenly spaced), resulting in 1000 timepoints in total. The spacing is therefore 0.1u, 100.1u, 200.1u, 300.1u, ... , 99900.1u and was not changed by participants.

The true parameters chosen for this model were between 0.5 and 3 times the default values in Table 2 (the participants were aware of these default values). This posed a difficult task to the participants, whilst still reflecting observations that could be seen in practice.

Parameter	Value	Parameter	Value
f_a	0.0032	r_c	0.05
k_b	0.0035	d_b	0.013
D_b	0.06	k_M	0.3
s_a	$8e^{-5}$	b_a	0.3
D_a	0.025	d_a	0.025

Table 5: Data generating parameters for Model 3 (SDEs describing eukaryotic chemotaxis, with all variables fully observed).

The SNR used to corrupt the output with Gaussian white noise was 100,000 (representing the precision of the equipment used to take measurements for this problem in practice) for each model output (GI, LI, LA, stimulus strength, x co-ordinate and y co-ordinate).

2.3.2 Additional challenge

As previously mentioned, the assumption that all the variables in model 3 are measurable is unrealistic in practice and is such, data was additionally made available from model 3, where only the x and y co-ordinates were observed. The data were simulated exactly as described in Section 2.3.1, but using different true parameters (as shown in Table 6) and with participants only obtaining the observations of the x and y co-ordinates. The true parameters chosen for this additional challenge were between 0.5 and 3 times the default values in Table 2 (the participants were aware of these default values). This posed a difficult task to the participants, whilst still reflecting observations that could be seen in practice.

Parameter	Value	Parameter	Value
f_a	0.004	r_c	0.09
k_b	0.002	d_b	0.01
D_b	0.05	k_M	0.1
s_a	$7e^{-5}$	b_a	0.075
D_a	0.015	d_a	0.02

Table 6: Data generating parameters for Model 3 (SDEs describing eukaryotic chemotaxis, with only the x and y co-ordinates observed).

3 Methodology

3.1 Heuristic identification of practical parameter spaces for ODE estimation (Model 1)

The novelty of the approach considered here rely greatly in the process of identifying the ‘right’ practical parameter spaces from which initial parameter values can be chosen when utilising the classical methods of parameter estimation.

Let the mathematical model of the cardiac excitation described in Section 1.1 be written in the general form as

$$\frac{d\mathbf{X}_t}{dt} = \mathbf{f}(\mathbf{X}_t, t, \Theta), \quad (12)$$

where t denotes time, $\mathbf{X}_t = [E, h, n]_t^T$ denotes the vector state variables with initial state value $\mathbf{X}_0 = [10, -1, 0]^T$, Θ the rate parameter and $\mathbf{f}(\cdot)$ describing the nonlinear relationship between \mathbf{X}_t and θ . Estimation of the unknown model parameters, Θ can only be possible if there is a minimum number of measured data available, $\mathbf{Y}_n = [Y_1, Y_2, Y_3]_n^T$ at a discrete time, $t_n (n = 1, 2, \dots, N)$. In the study, these measurements, \mathbf{Y}_n are the sum of the observable $\mathbf{X}_n(\Theta)$ and a measurement error, e_n . Thus,

$$\mathbf{Y}_n = \mathbf{X}_n(\Theta) + e_n, \quad (13)$$

We first perform what we have term as system exploratory analysis (SEA), a type of analysis where we explore the model (12) with respect to the given measured data (13). In dynamical systems modelling, no attention is usually given to this type of analysis as compared with statistical modelling. It is the process of identifying ranges of the parameter spaces where the trajectories of the model (12) starts to deviate from totally from the dynamics present in the time series plots of the measured data (13). The impetus for this method is that it helps to restrict the ranges of the parameters and as such, reduces the issues of parameters values being trapped at a local optima during optimisation. The SEA algorithm is given below.

Note that, the input parameter ranges are chosen with respect to the reference parameter values given. We select values below and above the reference value which changes by a factor relative to the reference value. The simulated trajectories for all the possible parameter combinations are then compared with the time series plots of the measured dataset to identify the

Algorithm 1 System Exploratory Analysis (SEA).

```

1: Inputs:
   Model:  $\frac{d\mathbf{X}_t}{dt} = \mathbf{f}(\mathbf{X}_t, t, \Theta)$ 
   Time span: [0, 300]
   Initial state value:  $\mathbf{X}_0 = [10, -1, 0]^T$ 
   Parameter ranges:  $\Theta = \begin{bmatrix} k_1 \text{ range} \\ \vdots \\ g_{22} \text{ range} \end{bmatrix}$ 

2: for  $k_1$  in  $k_1$  range do
3:   for  $k_2$  in  $k_2$  range do
4:      $\vdots$ 
5:     for  $g_{22}$  in  $g_{22}$  range do
6:       try
7:         Solve Model
8:       Catch
9:         Error
10:      end
11:    end for
12:     $\vdots$ 
13:  end for
14: end for

15: Return:
    Simulated Trajectories:  $E, h, n$ 

```

‘right’ practical parameter space for which dynamics present in the given timeseries data are not lost.

Secondly, statistical inference is carried out via a theoretically well established method called maximum likelihood estimation [14]. Based on the assumption that the measurements errors, \mathbf{e}_n in equation (13) are independent across all observations and at all discrete time points and that the measurement errors are independent and identically normally distributed with mean zero and known covariance Σ_n where,

$$\Sigma_n = \text{diag}(s_1^2, s_2^2, s_3^2), \quad (14)$$

with s_1, s_2 and s_3 respectively denoting the estimated standard deviations of the measured $E(Y_1)$, $h(Y_2)$ and $n(Y_3)$, least squares estimation which is an important special case of the maximum likelihood principles emerges [33]. Thus, maximising the likelihood function is equivalent to minimising the objective function

$$\mathcal{C}(\Theta|\mathbf{Y}) = \sum_{n=1}^N \|\mathbf{W}_n(\mathbf{Y}_n - \mathbf{X}_n(\Theta))\|^2. \quad (15)$$

It is worth noting that in practical terms, objective function (15) are often flat, with many local extrema or non-smooth, which can cause the function itself, or its derivatives, to be discontinuous at some points in the parameter space. Thus, to ensure we have attained global minima during numerical optimisation, we generate initial starting parameter values from the ‘right’ practical parameter space using the latin hypercube sampling scheme since it ensures that the entire range of each parameter is explored [27] and then minimisation is carried out as follows to obtain $\hat{\Theta}$:

- Fminsearchbnd optimiser [22] is executed for 150 iterations for different initial guesses and then the initial guess with the minimum functional value is further executed using Fminsearchbnd optimiser [22] till convergence is achieved.
- A second MATLAB package called Isqnonlin which is gradient-based algorithm is then used to evaluate the gradient at the estimated parameter values found by the Fminsearchbnd Optimiser [22].

Finally, the covariance matrix of the parameter estimators $\text{Cov}(\hat{\Theta})$ is the basis for assessing uncertainty of model parameter estimators and it is computed via the linear approximation technique. Thus,

$$\text{Cov}(\hat{\Theta}) = \frac{\mathcal{C}(\Theta|\mathbf{Y})}{(3 * N - 12)} (\hat{\mathbf{F}}^T \mathbf{W} \hat{\mathbf{F}})^{-1}, \quad (16)$$

where $\hat{\mathbf{F}} = \frac{\partial \mathbf{X}(\Theta)}{\partial \Theta} |_{\Theta=\hat{\Theta}}$. The standard deviation of the parameter estimators, $s(\hat{\Theta})$ is given by

$$s(\hat{\Theta}) = \sqrt{\text{diag}(\text{Cov}(\hat{\Theta}))}, \quad (17)$$

3.2 Parameter transforms, global optimisation and MCMC (Model 1)

3.2.1 Likelihood and posterior

From the way the problem was set, we know that

$$\mathbf{Y} = \mathcal{M}(t; \boldsymbol{\theta}) + \boldsymbol{\epsilon}, \quad (18)$$

where \mathbf{Y} is data, \mathcal{M} is the model, $\boldsymbol{\theta}$ is the parameters, and $\boldsymbol{\epsilon}$ noise. We also know that

$$\epsilon_i \sim \mathcal{N}(0, \sigma_i^2) \quad (19)$$

with some constant noise level σ_i for each output. Using this, we can define our likelihood function to be

$$p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\sigma}) = \prod_{i \text{ output}} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\sum_t \frac{(\mathcal{M}_i(t; \boldsymbol{\theta}) - Y_{i|t})^2}{2\sigma_i^2}\right). \quad (20)$$

Finally we define a Bayesian posterior distribution as

$$p(\boldsymbol{\theta}, \boldsymbol{\sigma}|\mathbf{Y}) = \frac{p(\boldsymbol{\theta}, \boldsymbol{\sigma})p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\sigma})}{p(\mathbf{Y})} \quad (21)$$

$$\propto p(\boldsymbol{\theta})p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\sigma}), \quad (22)$$

where $p(\boldsymbol{\theta})$ is our prior. Since we did not have certain prior knowledge for the parameters, we defined our prior as

$$p(\boldsymbol{\theta}) \sim \mathcal{U}(\boldsymbol{\theta}_{min}, \boldsymbol{\theta}_{max}), \quad (23)$$

where $\boldsymbol{\theta}_{min}$ and $\boldsymbol{\theta}_{max}$ indicate minimum and maximum values of $\boldsymbol{\theta}$ and act as boundaries on the parameter space. The choice of boundaries is important for the performance of optimisers and samplers, and is discussed in the following section.

3.2.2 Parameter transformations

Optimiser and sampler performance can be improved by applying parameter transformations that create a more convex search space. To find suitable transforms, we plotted 1D cuts or slices through the high-dimensional likelihood surface for the individual parameters. For each θ_i in $\boldsymbol{\theta}$, we looked at $h(\theta_i)$ defined as

$$h(\theta_i) := p(\mathbf{Y}|\theta_i, \boldsymbol{\theta}_{not\ i} = \boldsymbol{\theta}_{default}). \quad (24)$$

Based on these 1D likelihood slices and the electrophysiological meaning of the parameters, we divided the parameters into two groups, placing conductance values and rate constants in the first $\boldsymbol{\theta}_1 = (k_1, k_2, k_3, F_h, f_n, G_{Na}, g_{21}, g_{22})^T$, and reversal potential values in the second group $\boldsymbol{\theta}_2 = (E_1, E_{Na}, E_{\dagger}, E_*)^T$.

To preserve both model dynamics and the physiological interpretation of the parameters, the sign of $\boldsymbol{\theta}_1$ must remain unchanged. Furthermore, as all $\boldsymbol{\theta}_1$ appear as multiplication terms in the model, we concluded a transformation $g(\theta_i) = \ln(\theta_i)$ was most suitable for these parameters, and chose their boundaries $\boldsymbol{\theta}_{min}, \boldsymbol{\theta}_{max}$ to be $[0.1, 10] \times \boldsymbol{\theta}_{default}$.

In the model equations, the parameters in $\boldsymbol{\theta}_2$ all appear as subtractions from E . From this we concluded that $\boldsymbol{\theta}_2$ should be varied linearly by the optimisers and samplers, hence no transformation need be applied. As boundaries $[E_{min}, E_{max}]$ we chose -150 mV to 100 mV, which corresponds roughly to the minimum and maximum values of E seen in experiments.

The resulting 1D likelihood slices using our boundaries and transformations are shown for all parameters in Figure 5.

3.2.3 Noise estimate

We explored two options for finding the value of $\boldsymbol{\sigma}$: First, it can be included in the list of parameters and inferred at the same time. However, as it was given that the signal-to-noise ratio (SNR) is 10, by definition we can write

$$\text{SNR} = \frac{\sigma_{signal}^2}{\sigma^2} \implies \sigma^2 = \frac{\sigma_{signal}^2}{\text{SNR}}, \quad (25)$$

and hence

$$p(\boldsymbol{\theta}, \boldsymbol{\sigma}|\mathbf{Y}) \rightarrow p(\boldsymbol{\theta}|\mathbf{Y}, \boldsymbol{\sigma}). \quad (26)$$

We chose to use this value in the remainder of this work, thereby reducing the number of parameters to be inferred by 1, at the possible expense of a slight loss of precision.

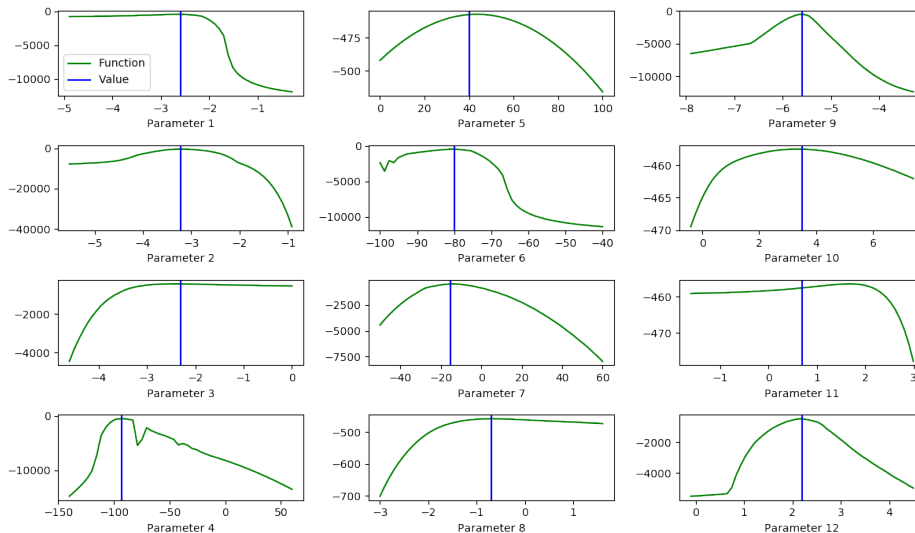


Figure 5: Cuts through the transformed likelihood surface along each parameter axis, keeping the remaining parameters at their default values. Each parameter is varied from the minimum to the maximum value we decided to explore. Note the values on the x-axis are for the transformed parameters.

3.2.4 Global optimisation and MCMC

We used a two-tier strategy, in which a global optimisation algorithm was used to find a candidate solution anywhere within the wide boundaries, and then a sampling algorithm was used to explore the posterior distribution around this point and to quantify the uncertainty in our parameter estimates.

For global optimisation we used the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [20] algorithm. As starting points for the optimisation we used either a uniformly sampled value from anywhere within the boundaries, or a value sampled from a distribution centered on $\theta_{default}$. For each case, we ran optimisations from 10 sampled starting points. Roughly half of the randomly picked initial guesses converged to a similar final log-likelihood value, while the other are several order of magnitude off which clearly indicates it stuck at some non-optimal parameter space. For initial guess starting from the default parameter values, all 10 runs, with different random seeds, converges to the same final log-likelihood value running from randomly picked initial guesses.

For the sampling stage, we used an adaptive covariance Markov Chain Monte-Carlo (MCMC) method, initialised at the best found found in the optimisation stage. The resulting Markov chain was then used to obtain a Maximum A-Posteriori probability (MAP) estimate.

All optimisation and MCMC were run using Pints, an open source Python package [10]. To run simulations in Python, we first re-implemented the model in Myokit [9] and verified our results matched the output of the competition Matlab code. Details of our method implementation and code is available in <https://github.com/CardiacModelling/cside-2018>.

We tested our approach by running repeated trials on simulated data sets intended to mimic the competition data, but for which the true parameters were known. In these tests, we found we could reliably recover the input parameter sets.

3.2.5 Precision matrix estimation

Although our methods return MCMC chains which represent our posterior estimation, the competition required a precision matrix as our final submitted result. We computed the precision matrix by inverting a covariance matrix calculated from the (detransformed) MCMC chain samples.

3.3 Bayesian Inference with emcee (Model 1)

We choose to use a Bayesian approach when inferring the parameters of Model 1, using a Markov Chain Monte Carlo (MCMC) method to draw samples from the posterior. In this section we will describe the model choices we made for prior and likelihood, the tools we used to produce these samples, and how we used them to generate our competition estimates.

3.3.1 Parameterisation and Priors

Most optimisation and sampling routines work best when we can remove constraints by choosing an appropriate parameterisation. From our reading of the original paper describing Model 1, we were able to determine three classes of original parameters:

1. Unconstrained parameters, which were sampled directly.
2. Constrained positive/negative parameters, such as k_1 . We parameterised these by using a log-scale, i.e. sampling $\log k_1$.
3. Ordered parameters $E_1 < E_{\dagger} < E_{\star}$. We parameterised these by taking logs of the differences between them, i.e. sampling $E_1, \log(E_{\dagger} - E_1), \log(E_{\star} - E_{\dagger})$.

Given these parameterisations, we then selected normal or log-normal priors $p(\theta_i)$ for each reparameterised variable as appropriate. As extreme values of parameters would cause the solver to crash or produce degenerate solutions, we reasoned that the competition parameters would not deviate too far from the example parameters provided. We selected parameters for the priors that ensured that values roughly around the order of magnitude of the example parameters would be considered plausible. Table 7 summarises our parameterisation and prior choices.

3.3.2 Likelihood

We choose to evaluate the likelihood directly, by running the provided solver for a given set of parameters. As we know the data was generated by adding independent Gaussian noise to the solver output, it makes sense to choose a Gaussian log-likelihood,

$$\log p(D \mid \theta, \sigma) = \sum_{j=1}^3 \sum_{t=1}^T \log \mathcal{N}(y_j^{(t)}; f_j(\theta)^{(t)}, \sigma_j), \quad (27)$$

Parameters	Prior
k_1, k_2, k_3	$\log \mathcal{N}(-2, 1)$
E_1	$\mathcal{N}(0, 100)$
$E_{\dagger} - E_1$	$\log \mathcal{N}(2, 1)$
$E_{\star} - E_{\dagger}$	$\log \mathcal{N}(3, 1)$
E_{Na}	$\mathcal{N}(0, 100)$
F_h	$\log \mathcal{N}(0, 1)$
F_n	$\log \mathcal{N}(-2, 1)$
G_{Na}	$\log \mathcal{N}(3, 30)$
$-g_{21}, -g_{22}$	$\log \mathcal{N}(1, 1)$

Table 7: Parameterisation and priors used for Method 3

where σ_j are additional parameters we can infer along with θ . If a particular sample of the parameters caused the solver to crash or otherwise take too long to produce a solution, we returned a log-likelihood of negative-infinity, guaranteeing that the sample will be rejected.

3.3.3 Sampling

We generated samples from the posterior using *emcee*[16], a Python implementation of the affine-invariant ensemble sampler[19], which only requires a function to evaluate the log-posterior up to a constant. The sampler runs a large number of chains in parallel, referred to as walkers, and proposes new positions for a walker by considering the current positions of all other walkers. This allows the sampler to account for the approximate shape of the posterior when generating proposals, allowing it to work well even in the presence of strong correlations. Beyond selecting the number of walkers, the sampler does not in general require the user to specify tuning settings.

The sampler requires careful selection of the starting point of the walkers in order to ensure convergence to the posterior within a reasonable number of samples. The *emcee* authors suggest using a tight cluster of points around the Maximum A Posteriori (MAP) estimate of the parameters, θ_{MAP} . We used the implementation of Powell’s method[34] available in the SciPy optimisation package[23] to find θ_{MAP} , as it does not require derivatives of the log-posterior to be taken, and is reasonable robust to invalid parameter settings that result in negative-infinity log-posterior values.

We ran the sampler on our log-posterior function with 100 walkers for 10,000 steps each, producing 1,000,000 samples in total. Each walker was initialised at the MAP solution, with a small amount of independent zero-mean Gaussian noise added. We opted for a conservative approach when considering burn-in, discarding the first half of each chain to leave 500,000 samples. This process was repeated 4 times in total to assess convergence.

3.3.4 Evaluation

The first step when assessing most Bayesian analyses is to check that the samples drawn from the posterior have actually converged. We used the potential scale reduction factor \hat{R} statistic, which measures how well multiple parallel MCMC chains agree[17]. If the \hat{R} value for each parameter is less than 1.1, the chains are usually considered adequately converged. Note that \hat{R}

cannot be computed directly on the parallel walkers used by *emcee*, as it requires the chains to be independent.

Having checked that our MCMC chains have converged, we then check that our procedure can adequately recover known parameters from the solver output. We initially used the example parameters provided with the solver, before using random perturbations of those example parameters as well. Finally, we used our samples to compute the expected ODE output $E[f(\theta)]$, and plotted the scaled residuals between this and the data to check that the data was actually normally distributed around the output and not skewed at any particular point in the timeseries.

3.4 Nearly exact Bayesian Inference (Model 2)

For model 2, a simple approach can compute the results of ideal Bayesian inference to within a small numerical tolerance.

We assumed we know the observation noise process, which was given in the competition description. Then model 2 only had a single unknown parameter, the stiffness k_3 . The likelihood function of k_3 given observations \mathbf{Y} is a 1D function that we can evaluate at any point. We simply plotted this likelihood for a grid of k_3 values. As the function was well-behaved, we could then select a region of high likelihood and replot the likelihood function on a fine grid.

By placing a uniform prior over the grid of k_3 values considered, we can compute the posterior distribution $P(k_3 | \mathbf{Y})$ over this grid of values. In the limit of a fine grid, expectations under this distribution will tend to those of the ‘true’ posterior distribution given by a continuous uniform prior over the range considered. To use another prior, we’d simply evaluate the prior on the grid and normalize it.

The mean and standard deviation of the posterior give an estimate of the unknown parameter k_3 . If the simulator output for known stiffness k_3 is $\hat{\mathbf{Y}}(k_3)$, the time series underlying the data can be estimated as a combination of these simulations, weighted by how plausible the parameters are: $\sum_i \hat{\mathbf{Y}}(k_3) P(k_3^{(i)} | \mathbf{Y})$.

3.5 Emulation with nonlinear optimisation, MCMC and bootstrapping (Model 2)

We start the analysis by specifying the statistical model:

$$\begin{aligned} q_t &= f^q(x_t; \boldsymbol{\theta}) + \epsilon_{1t}, \\ p_t &= f^p(x_t; \boldsymbol{\theta}) + \epsilon_{2t}, \end{aligned} \tag{28}$$

where $q_t \in \mathbf{q}$ are the noisy measured flows, $p_t \in \mathbf{p}$ are the noisy measured pressures from the first 3 vessels, $f(\cdot)$ describes the system behaviour that comes from numerically solving the fluids model equations (3) - (6), i.e. $f_q(\cdot)$ is the output (simulated) flow and $f_p(\cdot)$ is the output (simulated) pressure, $\boldsymbol{\theta}$ is the parameter that we wish to infer from the observed flow and pressure (i.e. stiffness in (4)), $x_t \in \mathbf{x}$ denote other input variables (e.g. inflow into the Main Pulmonary Artery) and ϵ are the errors, which we know are i.i.d following a Gaussian distribution and are different among the flow and pressure profiles.

We choose the residual sum of squares (RSS) as the objective function to be minimised; under the Gaussian assumption of the errors, it is proportional to the negative log likelihood of a Normal distribution. In this study, RSS is calculated for the 6 profiles: 3 flows and 3 pressures

and we aim to find the stiffness parameter value that minimises the RSS scaled by each signal variance in (29).

$$RSS = \sum_{j=1}^3 \left(\frac{(\mathbf{p}_j - f_j^p(\mathbf{x}; \boldsymbol{\theta}))^2}{\sigma_{pj}^2} + \frac{(\mathbf{q}_j - f_j^q(\mathbf{x}; \boldsymbol{\theta}))^2}{\sigma_{qj}^2} \right) \quad (29)$$

To find each signal’s noise variance, we use the fact that we know the signal-to-noise ratio (SNR):

$$SNR = \frac{\sigma_{\text{signal}}^2}{\sigma_{\text{noise}}^2}. \quad (30)$$

Prior to the data becoming available, we proceed by running a large number of forward simulations of the PDE model for a fine grid of parameter values within the biologically meaningful range provided by the organiser. This provides an emulator for the flow and pressure profiles, which can be used to extract the parameter value which minimises the RSS value (ordinary least squares) between the simulated data provided by the organiser and the ‘optimum’ data. Additionally, the emulator allowed quickly inspecting the landscape of the objective function, which was unimodal. We then insert the variance of this optimum data in (30) to find the noise variance of each signal, and subsequently obtain the reweighted RSS in (29). At the next step we updated our parameter estimate by applying nonlinear constraint optimisation to the rescaled RSS (reweighted least squares) with the gradient-based SQP algorithm [7], part of the *fmincon* function in *Matlab*. For this purpose and to ensure high accuracy, we used the simulator (i.e. solved the PDEs).

Having obtained the point estimate for the parameter in the model, we wish to quantify uncertainty in our knowledge of the parameter. This can proceed under either of the frequentist and Bayesian frameworks. Adopting a bayesian stance, we consider the parameter as a random variable, adopting a prior distribution which represents our level of knowledge before observing the data. In this case, a uniform prior was used, constraining the inference to a particular range without favouring any particular parameter values within this range. Uncertainty can be quantified using an MCMC sampler, where the noise variance is fixed at its estimated value (see eq. 28) Consideration from the frequentist viewpoint necessitates a different quantification technique as the randomness shifts from the parameters to the data; this time, we assume that there exists one true fixed parameter value that we hope to recover, with uncertainty resulting from different instantiations of the data. This randomness can be exploited in a bootstrapping framework, where we repeatedly conceptualise new data sets, introducing parameter uncertainty as a result of the uncertainty in the data. In cases where noise variance is unknown, one can adopt a nonparameteric bootstrapping approach, resampling from the current dataset to produce our conceptualised dataset and reoptimise the parameters. For the challenge considered here, the SNR was given beforehand and so having optimized the parameter, we could estimate the noise variance and generate new data by sampling new noise instantiaions from a Gaussian distribution. This process was repeated and the parameter reoptimized 100 times to provide an estimate of the uncertainty.

3.6 The Scaled Expected Improvement Method (Model 2)

Let $Y = \mathcal{M}(x)$ denote the computational model of the pulmonary circulation described in Section 1.2. The input, x , corresponds to the parameter Eh/r_0 in (4), while the output is the $T \times 6$ matrix

$$Y = [p_1, p_2, p_3, q_1, q_2, q_3]$$

comprising pressure ($p_i, i = 1, 2, 3$) and flow ($q_i, i = 1, 2, 3$) curves for timepoints $t = (t_1, \dots, t_T)$ at the midpoints of the first three vessels.

Inference of the model parameter x is performed by minimization of the following objective function:

$$f(x) = \sum_{i=1}^T \sum_{j=1}^6 \frac{(y_{ij} - y_{ij}^{\text{obs}})^2}{\hat{\sigma}_j^2}, \quad (31)$$

where $Y^{\text{obs}} = [y_{ij}^{\text{obs}}]$ is the observed data matrix provided by the conference organizers and $Y = [y_{ij}] = \mathcal{M}(x)$ is the simulation at x .

In (31), $\hat{\sigma}_j$ is an estimate of the noise standard deviation and is obtained from the definition of the signal-to-noise ratio (SNR):

$$SNR = \frac{\sigma_{\text{signal}}^2}{\sigma^2} \implies \sigma^2 = \frac{\sigma_{\text{signal}}^2}{SNR}. \quad (32)$$

Let Y_j^{obs} denote the j th column of the data matrix Y^{obs} . Each signal standard deviation σ_{signal} ($j = 1, \dots, 6$) is estimated by fitting a Gaussian process (GP) to the data (t, Y_j^{obs}) , with a constant mean function and a Matérn 5/2 kernel:

$$k(x_i, x_j) = \sigma_{\text{signal}}^2 \left(1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2} \right) \exp \left(-\frac{\sqrt{5}r}{l} \right), \quad r = \|x_i - x_j\|, \quad (33)$$

see [36] for more details. The value σ_{signal} is taken to be the corresponding estimated GP kernel hyperparameter, while the signal-to-noise ratio was given by the conference organizers to be equal 10.

Minimization of the objective function $f(x)$ is performed via Bayesian optimization (BO) using the Scaled Expected Improvement acquisition function [31]. Bayesian optimization is a *global* optimization algorithm that can tackle problems where the objective function is expensive-to-evaluate, possibly returning noisy values, and where gradient information might not be available. Optimization algorithms design a sequence of points $\{x_n\}$ that aims to find the global minimizer x_{global} . BO casts the optimization problem into a decision-theoretic framework, where the data are the function evaluations $\{x_i, f(x_i)\}$, the quantity to estimate is the minimizer x_{global} and the action or decision to take is the next query point x_{n+1} . The algorithm involves two steps: (i) iteratively maintaining a GP model of the objective function f , which is updated as new information on the function arrives, and (ii) evaluating f at the point which maximizes an acquisition function $a(x)$ derived from the GP in (i). In order to obtain a first set of data, the function is evaluated at $n_{\text{init}} = 10 \times d$ points sampled from a Latin hypercube design in the input space, with d denoting the dimensionality of the input space (in this case 1).

Let $f \sim GP(m, k)$ denote the GP prior on the objective function f , with $m(x) = c$ and $k(x_i, x_j)$ being the ARD Squared Exponential kernel:

$$k(x_i, x_j) = \sigma_{\text{signal}}^2 \exp\left(-\frac{1}{2} \sum_{k=1}^d \frac{(x_{ik} - x_{jk})^2}{l_k^2}\right). \quad (34)$$

The acronym ARD stands for automatic relevance determination, which means allowing for a different lengthscale l_k per input dimension. If the lengthscale along one dimension is large, the GP is flat along that direction, effectively meaning that the corresponding input variable is not relevant. The GP model hyperparameters, $\theta = (l_1, \dots, l_d, \sigma_{\text{signal}}, \sigma)$, are estimated at every iteration of the BO algorithm by maximizing the log marginal likelihood, see Section 5.4.1 in [36]. After n function evaluations $\mathcal{D}_n = \{(x_1, f_1), \dots, (x_n, f_n)\}$, with $f_i = f(x_i)$, the posterior GP is denoted:

$$f(\cdot) \mid \mathcal{D}_n \sim GP(\hat{f}(\cdot), s(\cdot, \cdot)) \quad (35)$$

$$\hat{f}(x) = m(x) + k(x)'(K + \sigma^2 I)^{-1}(y - m(X)) \quad (36)$$

$$s(x_i, x_j) = k(x_i, x_j) - k(x_i)'(K + \sigma^2 I)^{-1}k(x_j), \quad (37)$$

where $m(X) = (m(x_1), \dots, m(x_n))'$ is the prior mean at the training inputs, $k(x) = (k(x_1, x), \dots, k(x_n, x))'$ is the $n \times 1$ vector of covariances between the output at the training inputs and the test input, and $K = [k(x_i, x_j)]$ for $i, j = 1, \dots, n$. The predictive variance is obtained as $s^2(x) = s(x, x) = \text{Cov}[f(x), f(x)]$.

The incumbent minimum is denoted $f_{\min} = \min(f_1, \dots, f_n)$. From the marginalization property of GPs, at a given index x the random variable $f(x) \sim N(\hat{f}(x), s^2(x))$. Define the *Improve-ment* random variable as:

$$I(x) = \begin{cases} f_{\min} - f(x) & f(x) < f_{\min} \\ 0 & \text{otherwise.} \end{cases} \quad (38)$$

Let $u = (f_{\min} - \hat{f}(x))/s(x)$ and $\Phi(\cdot)$, $\phi(\cdot)$ denote the cumulative distribution function and probability density function of a $N(0, 1)$ random variable respectively. The expectation of $I(x)$ is known as the Expected Improvement (EI) acquisition function [28]:

$$EI(x) = \mathbb{E}[I(x)] \quad (39)$$

$$= s(x) \{u\Phi(u) + \phi(u)\}, \quad (40)$$

while the variance (see (5) in [31]) is equal to:

$$VI(x) = \text{Var}[I(x)] \quad (41)$$

$$= s^2(x) \{(u^2 + 1)\Phi(u) + u\phi(u)\} - \{EI(x)\}^2. \quad (42)$$

The Scaled Expected Improvement acquisition function (see (6) in [31]) is defined as:

$$\text{ScaledEI}(x) = EI(x)/\sqrt{VI(x)}. \quad (43)$$

Selecting the next query point to be the maximizer of the ScaledEI acquisition function means evaluating the function at a point where we expect a high improvement and we are confident about the value of the improvement being high. This function corresponds to the *average improvement per unit of variance* and is, as desirable, a *dimensionless quantity*. The BO algorithm with the ScaledEI acquisition function is summarised in Algorithm 2.

Algorithm 2 Scaled Expected Improvement Bayesian optimization.

1: **Inputs:**

Initial design: $\mathcal{D}_{n_{\text{init}}} = \{(x_i, f_i)\}_{i=1}^{n_{\text{init}}}$
Budget of n_{max} function evaluations

2: **for** $n = n_{\text{init}}$ **to** $n_{\text{max}} - 1$ **do**

- 3: Update the GP: $f(x) \mid \mathcal{D}_n \sim GP(\hat{f}(x), s(x, x'))$
4: Compute the ScaledEI acquisition function: $ScaledEI_n(x)$
5: Solve the auxiliary optimization problem:

$$x_{\text{next}} = \underset{x \in \mathcal{X}}{\operatorname{argmax}} ScaledEI_n(x)$$

- 6: Query f at x_{next} to obtain f_{next}
7: Augment data: $\mathcal{D}_{n+1} = \mathcal{D}_n \cup \{x_{\text{next}}, f_{\text{next}}\}$
8: **end for**

9: **Return:**

Estimated minimum: $f_{\text{min}} = \min(f_1, \dots, f_{n_{\text{max}}})$
Estimated point of minimum: $x_{\text{min}} = \operatorname{argmin}(f_1, \dots, f_{n_{\text{max}}})$

3.7 Gaussian process regression (Model 3)

We propose to infer the parameters of Model 3 using a Gaussian process (GP) regression, separately estimated for each parameter. This approach is useful on its own merit but importantly it can be combined with approximate Bayesian computation (ABC) schemes to allow for extra flexibility as discussed in Section 3.8. Below we present the key ideas of the basic regression method and refer to [18] for the details of both approaches.

3.7.1 Gaussian process regression

A GP regression is a nonparametric Bayesian model explaining the observed output data y_i in terms of values of unobserved (latent) function f evaluated at given input points \mathbf{x}_i , i.e.

$$y_i = f(\mathbf{x}_i) + \varepsilon_i, \quad \varepsilon_i \stackrel{\text{iid}}{\sim} \mathcal{N}(0, 1), \quad i = 1, \dots, N,$$

where N is the sample size. The prior assumption on the function f is $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$, i.e. that it follows a Gaussian process² with the mean function $m(\mathbf{x})$ and the covariance function $k(\mathbf{x}, \mathbf{x}')$ (kernel). We refer to [36] for an extensive treatment of GPs. In our application we are interested in explaining each of the 10 parameters of the SDE system (7)–(11) using features extracted from the high-dimensional output from the model simulator. To this end before the competition data became available, we had trained (or estimated) 10 regressions on synthetic data obtained using the SDE system simulator.

²A GP is a stochastic process such that the joint distribution of any finite set of random variables from this process is multivariate Gaussian.

3.7.2 Experiment design

The initial design was based on 1000 points (θ vectors) generated from a Sobol’ sequence for training and on 100 points from a Latin hypercube for testing (sensitivity analysis with respect to the kernel specification). For each θ from the initial set we obtained simulator outputs. A substantial difficulty in setting up the experiment design was posed by no prior knowledge on the plausible parameter values except the default values of the simulator $\tilde{\theta}$ specified by [43]. Therefore we needed to adopt a working assumption on the parameter domain, which we set to $[0.5\tilde{\theta}, 2\tilde{\theta}]$. The carried out extensive exploratory analysis revealed that the chosen range was sufficient to obtain a broad variety of cell behaviour patterns and at the same time did not cause stability problems to the simulator, see [18].

3.7.3 Features

A substantial challenge in running a GP regression was related to the choice of appropriate explanatory variables \mathbf{x} . As discussed in Section 1.3, the simulator generates 5 types of output, each being high-dimensional and complex. Figure 6 illustrates the raw signals as well as some measures derived from them, e.g. the ratio of cell’s area to its perimeter (varying over time) or mean (over contour) of LA (varying over time). It can be seen that the simulator outputs cannot directly serve as regressors, so it is necessary to base the inference on a set of features extracted from the generated data.

However, in this case constructing appropriate measures is challenging due to three main issues. First, the system (7)–(11) is stochastic and so is the simulator output. Therefore it is necessary to find features being similar for the datasets generated using the same (or similar) values of θ , yet allowing for distinguishing between outputs corresponding to very different θ s. Second, the space dimension of the output is not fixed but may vary within a single simulation (for a given θ) and across different simulations (for different θ s). This is due to the way the finite element nodes are specified in the simulator, which are added or removed from the list depending on the current shape of the cell’s membrane. Third, the finite element nodes are independently drawn at each time point in the simulation. This means that there is no alignment between points in subsequent time points, which prohibits any time series analysis of local signals at a given space location.

After an extensive exploratory analysis of the outputs and their properties, we extracted a large number of features which we used as inputs to the GP regressions (56 and 31 covariates for Model 3 and Additional Challenge, respectively). See [18] for a thorough discussion of feature construction. An important role therein was played by features based on Fourier transform of various signals. Here signals mean either values of a certain output at a given time point (so representing values in space) or time series of descriptive statistics (over space) of a certain output at each time point. Fourier transform allows for scale, translation and rotation invariance, by which it captures the properties of the shape of a given signal. The shape of a signal is of interest for both cell contours and chemical signals. In the former case we need to filter out random movements of a cell, in the latter case to account for the lack of time alignment of finite element points at which signals are measured. Since the dimension of Fourier components was still prohibitive to directly use in the GP regression, we applied principal component analysis (PCA) for dimensionality reduction and used the scores corresponding to the first two principal components as regressors.

3.7.4 Model specification

Typically, the performance of a GP regression crucially depends on the chosen kernel. Hence, we tried 5 different kernel types – squared exponential, Matérn 3/2, Matérn 5/2, rational quadratic and neural network – allowing for automatic relevance determination (ARD), i.e. a different length scale for each input. ARD results in downweighting of less informative features, which in this case are likely to be different for each regression. Due to the time constraints in the competition timeline we estimated the kernel hyperparameters using maximum likelihood as provided by `fitrgp` function from MATLAB® Statistics and Machine Learning Toolbox™. We note, however, that Bayesian estimation would be preferable as it allows for capturing (hyper)parameter uncertainty. The final kernel configuration was based on the out-of-sample performance of the fitted kernels on the test dataset as measured by mean relative squared errors. The final kernel configuration was then retrained on a larger sample based on 2000 points from a Sobol’ sequence, initialised from the hyperparameter estimates obtained in the first fitting. We refer to [18] for more details on the sensitivity analysis and kernel specification.

3.8 Approximate Bayesian computation (Model 3 additional challenge)

The system described in equations 7 – 11 is stochastic and the likelihood is intractable, hence, classical inference methods can not be applied in this scenario. A different class of methods, called likelihood-free methods, can be applied to such scenarios. This section details an application of the Sequential Monte Carlo approximate Bayesian computation (SMC ABC) algorithm [3] to the SDE system describing cell chemotaxis in response to pseudopod formation. The steps in the algorithm are summarised in Algorithm 3.

In general, in an ABC algorithm, the likelihood calculation is replaced by steps which involve simulating artificial data from the model given different parameter values and comparing this data with the observed data. For high dimensional data, the comparison is typically made in terms of summary statistics, which are extracted from the artificial and the observed data set. The summary statistics are then compared, and if they are similar enough, the parameters used to generate the artificial data set are considered to be a sample from an approximate posterior distribution. While the ABC idea is simple and provides a tool to deal with parameter inference, one of the most important and problematic issues with these algorithms is finding suitable summary statistics that are approximately sufficient and low dimensional. Given the specific nature of each problem, it is impossible to specify a set of summary statistics that will work in every situation. Fearnhead and Prangle’s [12] solution to this problem consists of constructing linear regression models with the parameters as response variables and transformations of the data (summary statistics) as predictors:

$$\boldsymbol{\theta}_i = E(\boldsymbol{\theta}_i|\mathbf{y}) + \varepsilon_i, \text{ where } \varepsilon_i \text{ is zero mean noise} \quad (44)$$

Within the ABC algorithm, the regression models are used to generate predictions of each parameters using a set of summary statistics extracted from each synthetic data set. These predictions are then compared to the predictions obtained from the real data set, and if they are close, the sample is accepted as part of the approximate posterior distribution. We improve on Fearnhead and Prangle’s [12] idea by replacing the linear regression model in Equation 44 with a more flexible GP regression model, described in more detail in Section 3.7.

Algorithm 3 Sequential Monte Carlo approximate Bayesian computation (SMC ABC) [3].

- 1: At iteration $t = 1$
- 2: **for** $i = 1, \dots, N$ **do**
- 3: Simulate $\theta_i \sim \pi(\theta_i)$ and $x \sim f(x|\theta_i^{(1)})$ until $\rho(S(x), S(y)) < \epsilon_1$
- 4: Set $w_i^{(1)} = \frac{1}{N}$
- 5: **end for**
- 6: Take τ_2^2 as twice the empirical variance of the $\theta_i^{(1)}$'s

- 7: At iteration $2 \leq t \leq T$
- 8: **for** $i = 1, \dots, N$ **do**
- 9: pick θ_i^* from $\theta_j^{(t-1)}$'s with probabilities $w_j^{(t-1)}$
- 10: generate $\theta_i^{(t)}|\theta_i^* \sim \mathcal{N}(\theta_i^*; \tau_t^2)$ and $x \sim f(x|\theta_i^{(t)})$, until $\rho(S(x), S(y)) < \epsilon_t$
- 11: Set $w_i^{(t)} \propto \frac{\pi(\theta_i^{(t)})}{\sum_{j=1}^N w_j^{(t-1)} \mathcal{N}(\theta_i^{(t)}|\theta_j^{(t-1)}; \tau_t^2)}$
- 12: **end for**
- 13: Take τ_{t+1}^2 as twice the weighted empirical variance of the $\theta_i^{(t)}$'s

π is a prior distribution, f is the simulator, ρ is a distance metric, S represents the summary statistics and w are particle weights.

The algorithm described in Algorithm 3 is a sequential method which works by generating populations of N points at each iteration of the algorithm, t . These populations are used to produce better sampling proposals in later iterations. Lastly, the sequence of decreasing tolerance levels, $\epsilon_1, \dots, \epsilon_t$, ensures the populations get progressively more focused in areas with high posterior probability.

4 Results

This section will outline the different criteria used to rank the performance of participating methods, as well as the results of the leading participants. Different types of criteria will give different information as to a method's performance and it may be of interest to focus on particular type of criteria in future events in order to gain insight into a specific question of interest (such as convergence capabilities or classification of outcomes). The type of criteria used to assess the methods for this event was estimation accuracy of the data generating parameters. Accuracy was assessed in both the parameter and data domain. There were three criteria used to assess the results and participants received a rank for each based on their method's performance relative to the rest of the participants (for their chosen model i.e. methods were not compared between models). A competitor's overall score was the average rank across criteria. This information, as well as the assessment criteria themselves, were made available to participants at the beginning of the challenge.

Each participant has included further information about the implementation of their methods and their obtained results, which can be found in Sections 5.1-5.8 in the Appendix (omitted here for brevity).

4.1 Assessment criteria

The criteria were chosen in order to assess both how close point estimates were to the true parameters, as well as how much certainty a participant had in their estimate i.e. how peaked/dispersed the uncertainty intervals of the parameters were. The criteria were also chosen in order to not inherently favour a Bayesian/Frequentist approach to parameter inference.

Criterion 1. Parameter Domain, Weighted Root Mean Square (RMS):

Participants submitted a point estimate for each parameter in their chosen model and from these, the data generating parameters were subtracted. The differences were then weighted by the data generating parameters, in order to avoid different parameter magnitudes from dominating the calculation, and the RMS was then calculated.

$$\text{Weighted RMS} = \sqrt{\sum_{i=1}^p \left(\frac{\hat{\theta}_i - \theta_i}{\theta_i} \right)^2}$$

Here, p is the number of parameters in a chosen model, $\hat{\theta}_i$ is the i^{th} parameter point estimate from the participant and θ_i is the i^{th} parameter that generated the data. Methods that produced smaller weighted RMS values were ranked higher.

Criterion 2. Data Domain, Root Mean Square (RMS):

Participants used the point estimate(s) from their chosen model to solve the system of equations. From the resulting predicted signal the signal produced using the data generating parameters was subtracted and the RMS was then calculated.

$$\text{RMS} = \sqrt{\sum_{i=1}^N (\hat{X}_i - X_i)^2}$$

Here, N is the total number of datapoints, \hat{X}_i is the i^{th} point of the signal produced using the participant's estimate(s) and X_i is the i^{th} point of the signal produced using the parameters that generated the data. Methods that produced smaller RMS values were ranked higher. Note that Model 3 was not assessed using this criterion, since Model 3 is an SDE (see Section 1.3 for details) and solving the equations using the same parameter values can produce different signals due to the stochasticity present in the system.

Criterion 3. Parameter Domain, Likelihood:

This criterion was included in order to assess a participant’s uncertainty quantification in their parameters. Participants submitted a point estimate for each parameter in their chosen model and either a standard deviation/standard error corresponding to each parameter or a precision matrix corresponding to all parameters. The likelihood (assuming a Gaussian distribution) of the data generating parameter(s) given the participant’s estimates and standard deviation(s)/standard error(s)/precision matrix was then calculated.

$$\text{Likelihood} = \prod_{i=1}^p \mathcal{N}(\theta_i | \hat{\theta}_i, \sigma)$$

or

$$\text{Likelihood} = \mathcal{N}(\boldsymbol{\theta} | \hat{\boldsymbol{\theta}}, \boldsymbol{\Sigma}^{-1})$$

Here, p is the number of parameters in the chosen model, $\mathcal{N}(\dots)$ denotes the Gaussian distribution, θ_i is the i^{th} parameter that generated the data, $\hat{\theta}_i$ is the i^{th} parameter point estimate of the participant, σ is the i^{th} standard deviation/standard error corresponding to the i^{th} parameter estimate, $\boldsymbol{\theta}$ is the vector of parameters that generated the data, $\hat{\boldsymbol{\theta}}$ is the vector of parameter point estimates of the participant and $\boldsymbol{\Sigma}^{-1}$ is the precision matrix corresponding to all the estimated parameters of the participant’s chosen model. Methods that produce larger likelihood values are ranked higher.

4.2 Results: Model 1

Participant Method	Weighted RMS	RMS	Log Likelihood
Heuristic identification of practical parameter spaces (see Section 3.1 for details)	1.5506	75.2617	-2.8985e+05
Parameter transforms, global optimisation and MCMC (see Section 3.2 for details)	1.5480	38.4195	-22.1512
Bayesian Inference with emcee (see Section 3.3 for details)	0.9172	39.5400	-12.0482

Table 8: Participants’ scores for Model 1 (see Section 1.1 for details). For details of the assessment criteria, see Section 4.1.

4.3 Results: Model 2

Participant Method	Weighted RMS	RMS	Likelihood
Nearly exact Bayesian Inference (see Section 3.4 for details)	0.0065	1.2684	4.9367e-04
Emulation with nonlinear optimisation, MCMC and bootstrapping (see Section 3.5 for details)	0.0036	0.6991	6.6067e-04
The Scaled Expected Improvement Method (see Section 3.6 for details)	0.0032	0.6185	8.8216e-04

Table 9: Participants' scores for Model 2 (see Section 1.2 for details). For details of the assessment criteria, see Section 4.1.

4.4 Results: Model 3

Participant Method	Weighted RMS	Log Likelihood
Gaussian process regression (see Section 3.7 for details)	0.6847	33.9102
Approximate Bayesian computation* (see Section 3.8 for details)	1.1226	24.9057
Gaussian process regression (see Section 3.7 for details)	2.2872	31.1522
Approximate Bayesian computation (see Section 3.8 for details)	2.5021	27.8420
Approximate Bayesian computation* (see Section 3.8 for details)	1.5448	35.6111

Table 10: Participants' scores for Model 3 (see Section 1.3 for details). The first row of results pertains to the fully observed problem and the second row of results pertains to the partially observed problem i.e. the additional challenge (see Section 2.3 for details). For details of the assessment criteria, see Section 4.1. Note that results with a * were submitted after the event deadline and were not assessed as part of the competition. These results, however, took 7 days (each) to run and therefore are comparable to the other results in the table.

References

- [1] Alexander, D., Mo, M., Stent, A.: Arithmetic brownian motion and real options. Elsevier - European Journal of Operational Research **219**, 114–122 (2012)
- [2] Andrew, N., Insall, R.H.: Chemotaxis in shallow gradients is mediated independently of ptdins 3-kinase by biased choices between random protrusions. Nature Cell Biology **9** (2007). URL <https://doi.org/10.1038/ncb1536>
- [3] Beaumont, M.A., Cornuet, J.M., Marin, J.M., Robert, C.P.: Adaptive approximate bayesian computation. Biometrika **96**(4), 983–990 (2009)
- [4] Bezekci, B., Idris, I., Simitev, R.D., Biktashev, V.N.: Semianalytical approach to criteria for ignition of excitation waves. Physical Review E **92**(4) (2015). DOI 10.1103/physreve.92.042917. URL <https://doi.org/10.1103/physreve.92.042917>
- [5] Biktashev, V.N., Suckley, R.S., Elkin, Y.E., Simitev, R.D.: Asymptotic analysis and analytical solutions of a model of cardiac excitation. BMB **70**(2), 517–554 (2008)
- [6] Biktasheva, I., Simitev, R., Suckley, R., Biktashev, V.: Asymptotic properties of mathematical models of excitability. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **364**(1842), 1283–1298 (2006). DOI 10.1098/rsta.2006.1770. URL <https://doi.org/10.1098/rsta.2006.1770>
- [7] Boggs, P., Tolle, J.: Sequential quadratic programming for large-scale nonlinear optimization. Journal of Computational and Applied Mathematics **124**(1–2), 123 – 137 (2000). DOI [https://doi.org/10.1016/S0377-0427\(00\)00429-5](https://doi.org/10.1016/S0377-0427(00)00429-5). URL <http://www.sciencedirect.com/science/article/pii/S0377042700004295>. Numerical Analysis 2000. Vol. IV: Optimization and Nonlinear Equations
- [8] Carmeliet, E., Vereecke, J.: Cardiac Cellular Electrophysiology. Springer US (2002). DOI 10.1007/978-1-4615-1715-3. URL <https://doi.org/10.1007/978-1-4615-1715-3>
- [9] Clerx, M., Collins, P., de Lange, E., Volders, P.G.A.: Myokit: A simple interface to cardiac cellular electrophysiology. Progress in Biophysics and Molecular Biology **120**(1–3), 100–114 (2016)
- [10] Clerx, M., Robinson, M., Lambert, B., Lei, C.L., Ghosh, S., Mirams, G.R., Gavaghan, D.J.: Probabilistic inference on noisy time series (PINTS). Journal of Open Research Software **7**(1), 23 (2019). DOI 10.5334/jors.252
- [11] Colebank, M., Qureshi, M.U., Olufsen, M.S.: Sensitivity analysis and uncertainty quantification of 1D models of pulmonary hemodynamics in mice under control and hypertensive conditions. Int J Numer Meth Biomed Engng (2019). DOI <https://doi.org/10.1002/cnm.3242>
- [12] Fearnhead, P., Prangle, D.: Constructing Summary Statistics for Approximate Bayesian Computation: Semi-Automatic Approximate Bayesian Computation. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **74**(3), 419–474 (2012)
- [13] Fick, A.: On liquid diffusion. Elsevier - Journal of membrane science (1995 reprint)

- [14] Fisher, R.A.: On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* **222**(594-604), 309–368 (1922)
- [15] FitzHugh, R.: Impulses and physiological states in models of nerve membrane. *Biophys. J.* **1**, 445–466 (1961)
- [16] Foreman-Mackey, D., Hogg, D.W., Lang, D., Goodman, J.: emcee: the mcmc hammer. *Publications of the Astronomical Society of the Pacific* **125**(925), 306 (2013)
- [17] Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A., Rubin, D.B.: *Bayesian Data Analysis*, third edn. Texts in statistical science series. CRC Press, Taylor and Francis Group, Boca Raton London New York (2014)
- [18] Giurghita, D., Borowska, A., Husmeier, D.: Parameters inference for a stochastic differential equation system for chemotaxis. Tech. rep., arXiv (2018)
- [19] Goodman, J., Weare, J.: Ensemble samplers with affine invariance. *Communications in Applied Mathematics and Computational Science*, Vol. 5, No. 1, p. 65-80, 2010 **5**, 65–80 (2010). DOI 10.2140/camcos.2010.5.65
- [20] Hansen, N.: The cma evolution strategy: a comparing review. In: *Towards a new evolutionary computation*, pp. 75–102. Springer (2006)
- [21] Jabot, F., Faure, T., Dumoulin, N., Albert, C.: EasyABC: Efficient Approximate Bayesian Computation Sampling Schemes (2015). URL <https://CRAN.R-project.org/package=EasyABC>. R package version 1.5
- [22] John D’Errico: Bound constrained optimization using fminsearch (2012). URL <https://uk.mathworks.com/matlabcentral/fileexchange/8277-fminsearchbnd-fminsearchcon>
- [23] Jones, E., Oliphant, T., Peterson, P., et al.: SciPy: Open source scientific tools for Python (2001–). URL <http://www.scipy.org/>
- [24] Keener, J., Sneyd, J. (eds.): *Mathematical Physiology*. Springer New York (2009). DOI 10.1007/978-0-387-75847-3. URL <https://doi.org/10.1007/978-0-387-75847-3>
- [25] Lorenz, E.: Deterministic non-periodic flow. *Journal of the Atmospheric Sciences* (1963)
- [26] Lotka, A.: The growth of mixed populations: two species competing for a common food supply. *Journal of the Washington Academy of Sciences* **22**, 461–469 (1932)
- [27] McKay, M.D., Beckman, R.J., Conover, W.J.: Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**(2), 239–245 (1979)
- [28] Mockus, J., Tiesis, V., Zilinskas, A.: *The application of Bayesian methods for seeking the extremum*, vol. 2, pp. 117–129. North-Holland (1978)
- [29] Neilson, M.P., Veltman, D.M., van Haastert, P.J.M., Webb, S.D., Mackenzie, J.A., Insall, R.H.: Chemotaxis: a Feedback-Based Computational Model Robustly Predicts Multiple Aspects of Real Cell Behaviour. *PLoS biology* **9**(5), 1–11 (2011)

- [30] Noble, D.: A modification of the Hodgkin-Huxley equations applicable to Purkinje fibre action and pace-maker potentials. *JPL* **160**, 317–352 (1962)
- [31] Noè, U., Husmeier, D.: On a new improvement-based acquisition function for Bayesian optimization. arXiv preprint arXiv:1808.06918 (2018)
- [32] Olufsen, M.S., Hill, N.A., Vaughan, G.D.A., Sainsbury, C., Johnson, M.: Rarefaction and blood pressure in systemic and pulmonary arteries. *Journal of Fluid Mechanics* **705**, 280–305 (2012). DOI 10.1017/jfm.2012.220
- [33] Omlin, M., Reichert, P.: A comparison of techniques for the estimation of model prediction uncertainty. *Ecological Modelling* **115**(1), 45–59 (1999)
- [34] Powell, M.J.D.: An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal* **7**(2), 155–162 (1964). DOI 10.1093/comjnl/7.2.155. URL <http://dx.doi.org/10.1093/comjnl/7.2.155>
- [35] Qureshi, M.U., Colebank, M.J., Paun, L.M., Ellwein Fix, L., Chesler, N., Haider, M.A., Hill, N.A., Husmeier, D., Olufsen, M.S.: Hemodynamic assessment of pulmonary hypertension in mice: a model-based analysis of the disease mechanism. *Biomechanics and Modeling in Mechanobiology* **18**(1), 219–243 (2019). DOI 10.1007/s10237-018-1078-8. URL <https://doi.org/10.1007/s10237-018-1078-8>
- [36] Rasmussen, C.E., Williams, C.K.: Gaussian process for machine learning. MIT press (2006)
- [37] Sigg, D.C., Iaizzo, P.A., Xiao, Y.F., He, B. (eds.): *Cardiac Electrophysiology Methods and Models*. Springer US (2010). DOI 10.1007/978-1-4419-6658-2. URL <https://doi.org/10.1007/978-1-4419-6658-2>
- [38] Simitev, R.D., Biktashev, V.N.: Conditions for propagation and block of excitation in an asymptotic model of atrial tissue. *Biophysical Journal* **90**(7), 2258–2269 (2006). DOI 10.1529/biophysj.105.072637. URL <https://doi.org/10.1529/biophysj.105.072637>
- [39] Simitev, R.D., Biktashev, V.N.: Asymptotics of conduction velocity restitution in models of electrical excitation in the heart. *BMB* **73**(1), 72–115 (2011)
- [40] Strebel, O.: A preprocessing method for parameter estimation in ordinary differential equations. Elsevier - Chaos, Solitons and Fractals (2013)
- [41] Tabima, D., Roldan-Alzate, A., Wang, Z., Hacker, T., Molthen, R., Chesler, N.: Persistent vascular collagen accumulation alters hemodynamic recovery from chronic hypoxia. *J Biomech* pp. 799–804 (2012). DOI 10.1016/j.jbiomech.2011.11.020
- [42] Tomé, T., Oliveira, M.: *Stochastic Dynamics and Irreversibility*. Springer (2015)
- [43] Tweedy, L.: Meinhart Simulations on an Evolving Line. Tech. rep., Cside 2018 Supplementary Notes (2018)
- [44] Tweedy, L., Meier, B., Stephan, J., Heinrich, D., Endres, R.G.: Distinct cell shapes determine accurate chemotaxis. *Scientific Reports* **3**, 2606 (2013)

- [45] Vanderpool, R., Kim, A., Chesler, N.: Effects of acute Rho kinase inhibition on chronic hypoxia-induced changes in proximal and distal pulmonary arterial structure and function. *J Appl Physiol* p. 188– 198 (2011). DOI 10.1152/jappphysiol.00533.2010

5 Appendix

5.1 Heuristic identification of practical parameter spaces for ODE estimation (Results Model 1)

Figure 7 shows the timeseries plots of $E(Y_1)$, $h(Y_2)$ and $n(Y_3)$. These profiles demonstrate how each of the three measured state evolves over time.

As an illustration of the SEA, Figures 8 and 9 respectively depicts the case when the value of the parameter k_1 is change below and above the reference value whilst keeping the remaining parameters at their reference values. As can be seen in these figures, k_1 affects all the three state variables (E , h and n) and with $k_1 = -0.1$, the simulated dynamics of the state variables deviate from the dynamics present in the data (Figure 7) especially for h and n . Thus, the lower limit for k_1 can be set to be equal to $k_1 = 0$. Likewise for $k_1 = 0.3$ in Figure 9, the simulated dynamics deviate from the dynamics present in the data (Figure 7). Thus, the upper limit for k_1 can be set to be equal to $k_1 = 0.3$.

Following similar deductions for the remaining parameters, Table 11 contains the lower and upper limits defined for the numerical optimisation step. With these limits, the initialising parameter values are chosen using the LHS [27] for the optimisation of the objective function (15). The estimated model parameter values and their corresponding standard deviations are reported in Table 11. Figure 10 shows the true dynamics and model fit of E , h and n . These profiles demonstrate the performance of the estimation approach.

Table 11: Results of system exploratory analysis and nonlinear weighted least squares estimation

Parameter	True value	lower limit	upper limit	Estimate	Std
k_1	0.12	0	0.3	0.122	0.004
k_2	0.1	0.01	0.2	0.080	0.003
k_3	0.3	0.01	0.5	0.122	0.015
E_1	-65	-120	-40	-63.548	0.109
E_{Na}	60	60	70	62.674	0.291
E_{\dagger}	-60	-120	-40	-56.592	0.119
E_*	-20	-25	-5	-22.305	0.048
F_h	0.25	0.01	1	0.335	0.015
f_n	0.015	0	0.1	0.015	1.642×10^{-4}
G_{Na}	50	0	60	13.592	0.062
g_{21}	-7	-10	0	-7.583	33.849
g_{22}	-5	-15	-3	-3	0.006

5.2 Parameter transforms, global optimisation and MCMC (Results Model 1)

The inferred parameter distribution we obtained is visualised in Figure 11, which shows the 1D marginal plots of the obtained posterior distribution in the parameter transformed space, along with the MAP estimates and the values used to generate the competition data. Three

independent MCMC chains are shown, from runs initialised at different CMA-ES optimisation results.

Most of the true parameters are within our obtained posterior distribution, which indicates we successfully captured the uncertainty of the parameters in the competition data. Two parameters, k_3 and F_h , are at the edge of the marginal distribution. The reason for this is that F_h appears only in dh/dt which is a very fast component in the ODE system, and so only a few of the observed data points are sensitive to this parameter, giving us relatively little information. Similarly, the effect of k_3 can only be seen for a very short period during which $E > E_*$, where E_* is -30 mV.

To investigate correlations between parameters, we created the 2D pairwise marginal plots of the obtained posterior distribution shown in Figure 12. Only one of the three chains in Figure 11 is shown here. This figure shows that some of the parameters have a strong correlation in the MCMC chain. However, the true parameter values are all within each of the 2D marginal distribution.

The posterior model predictions are shown in Figure 13. The predictions (solid blue lines) are well within the competition data points (grey crosses), and seem to find a nice average. The predicted lines go very close to the ‘true’ simulation (red line). In some areas the blue predictions even appear to be a more likely source for the sampled data than the simulation using true values. We believe this is due to the noise added to the spike in the data which introduces a bias in the data that gets reflected in the inferred parameters. This is shown in more detail in the right panels of Figure 13, which zoom in on the initial rapid upstroke of the action potential.

Figure 14 shows the 1D marginal plots of the posterior distribution in the model parameter space. It shows that the marginal posteriors of some of the untransformed parameters are not Gaussian, in contrast to the more Gaussian distributions shown in Figure 11.

Finally, Figure 15 shows a 1D cut of the log-likelihood surface along a line connecting the true model parameter values to the obtained MAP model parameter values. This turns out to be a non-convex type hyperline, which clearly suggests that the log-likelihood function is non-Gaussian, confirming the non-Gaussian nature of our posterior. This suggests that our precision matrix approximation in the model parameters is not a good representation of our MCMC likelihood estimates.

5.3 Bayesian Inference with emcee (Results Model 1)

Our procedure results in a set of samples from the posterior. Running our procedure 4 times in parallel took approximately 8 hours in total using a 3.5 GHz Intel Xeon E3-1245 desktop CPU.

A fully Bayesian approach would be to use all of these samples to compute any expectations needed. However, the format of the competition required us to submit point estimates for the parameters, along with a covariance matrix for the parameters, and a point estimate for the true data-generating signal. The ranking criteria used for the competition means that we submitted the empirical mean of the parameter samples rather than the MAP estimate, as the mean maximises our expected score over our estimate of the posterior. Similarly, we submitted the empirical covariance matrix computed directly from the parameter samples. As the eigenvalues of this matrix spanned several orders of magnitude, we added a small value to the diagonal of the matrix to improve numerical stability. For a real application, it would be better to rescale the parameters to avoid this problem. For the signal estimate, we submitted the mean over each

output associated with each parameter sample, rather than running the solver once with the mean parameter estimate, which again maximises our expected score.

Our method was ranked first in the competition, which validates our approach. The use of tuning-free algorithms for both our MAP optimisation and our sampling process freed us to focus on modelling choices, whilst an appropriate parameterisation based on model knowledge allowed these algorithms to work effectively. A robust checking procedure as described in Section 3.3.4 allowed us to catch mistakes we made when specifying the model and ensured we were confident in our submitted parameter estimates.

5.4 Nearly exact Bayesian Inference (Results Model 2)

Based on a preliminary plot of the log-likelihood, we evaluated the posterior for a uniform prior on $k_3 \in \{72000, 72010, 72020, \dots, 77000\}$. The posterior is close to a normal distribution with mean 74510.9 and standard deviation 512.6, and these values were reported as estimates.

Along with this submission, we noted informally that the ‘nice’ number $k_3 = 75000$ is perfectly plausible under this posterior, and guessed that this was in fact the right answer (which it was). Using a prior that strongly favors numbers that someone is likely to pick by hand would have given a winning entry, but we felt such an entry wasn’t in the spirit of the competition.

Slight changes to the model (the prior or likelihood) result in slightly different posteriors. For example, estimating the noise levels rather than using the known levels, gives a posterior shifted slightly (≈ 200) to the right. Such an entry, although using a less informative model, would have ranked more highly in the competition on this particular occasion. However, the range of stiffnesses k_3 that we should consider plausible is similar under these different models.

5.5 Emulation with nonlinear optimisation, MCMC and bootstrapping (Results Model 2)

The parameter value which minimised the RSS was 74726 – see Figure 16; when the signals were reweighted by the corresponding noise variance, i.e. we optimised the negative log likelihood, we obtained an estimate of 74729.

Profile plots for the first vessel were generated with this estimate and can be seen in Figure 17. As one would expect, uncertainty quantification using the Bayesian and frequentist methods gave very similar results. Using bootstrapping, our standard deviation estimate was found to be 533 and using MCMC, 531.

5.6 The Scaled Expected Improvement Method (Results Model 2)

Figure 18 shows two of the six independent GPs over the data (t, Y_j^{obs}) , for $j = 1$ and 4. These corresponds to the pressure p_1 and flow q_1 in the first vessel. Similar plots are obtained for the remaining two vessels. Each GP has an estimated kernel hyperparameter σ_{signal} , which is used to obtain the corresponding estimate of the noise standard deviation $\hat{\sigma}_j$, shown in Table 12.

We ran the BO algorithm with the Scaled Expected Improvement acquisition function for a budget of $n_{\text{max}} = 300$ function evaluations, obtaining the following minimizer of (31):

$$x_{\text{min}} = 74760$$

Table 12: Estimated noise standard deviations

$\hat{\sigma}_1$	$\hat{\sigma}_2$	$\hat{\sigma}_3$	$\hat{\sigma}_4$	$\hat{\sigma}_5$	$\hat{\sigma}_6$
1.8641	1.6501	1.9032	0.0039	0.0004	0.0018

while an estimate of the standard deviation at the minimizer was obtained using the Cramer-Rao lower bound method:

$$\text{std} = 364$$

The optimization curve, i.e. the incumbent minimum f_{\min} vs n , is shown in Figure 19. The parameter estimate was obtained in approximately 30 minutes CPU time on a MacBook Pro with 2.6 GHz six-core Intel Core i7 processor.

The submission also required a forward simulation from the PDE model of the pulmonary circulation at the estimated parameter x_{\min} , shown in red in Figure 20 for the first vessel. Similar plots are obtained for the 2nd and 3rd vessels.

5.7 Gaussian process regression (Results Model 3)

The GP regressions discussed in Section 3.7.4 were used to infer the parameters of the model (7)–(11) for the two datasets (for Model 3 and Additional Challenge) provided by the Organisers. The submitted results, which in both cases ranked 1st in the Competition, are reported in Table 13 together with the resulting errors with respect to the true values (disclosed after the submission). The shaded fields refer to the cases for which the true parameter values were outside the prior domain discussed in Section 3.7.2, so that *ex post* the initial design turned out to be little informative for these parameters. In the remaining cases we can see that the true values are always in the estimated 2-standard deviation width credible intervals (with the sole exception of D_a for the Additional Challenge). This demonstrates the accuracy of the developed method for this challenging problem. As expected, chemical signals are generally highly informative about the model parameters, with the relative errors for Model 3 typically being several times lower than for the Additional Challenge.

The estimates, in particular the parameter means, can be used to generate “predicted datasets”. [18] discuss the qualitative and quantitative properties of outputs generated this way. For Model 3, they report a close accordance between the true Competition data and the “mean-predicted dataset”, both visually as well as in terms of several summary statistics. For the Additional Challenge the reported agreement is somewhat less evident, however it is still better than for the dataset generated using the default parameters $\tilde{\theta}$, demonstrating the usefulness of the proposed method even for the nontrivial case of partly observed system.

5.8 Approximate Bayesian computation (Results Model 3 additional challenge)

The ABC SMC algorithm, described in Section 3.8, is applied to the system described in equations 7 – 11. The implementation from the EasyABC package [21] in R was used to run the ABC SMC algorithm.

For each of the two challenges, *Model 3* and the *Additional challenge*, we use the GP regression estimates presented in section 5.7 to obtain summary statistics for each parameter. The

Euclidean distance is calculated between the summary statistics obtained from the GP regression prediction on the competition data and the GP regression prediction on particles proposed by the algorithm.

The ABC SMC algorithm is started from uniform priors for each parameter: $\mathcal{U}(0.3\theta, 2\theta)$. The tolerance levels for *Model 3* are chosen as $\epsilon_1 = 40, \epsilon_2 = 20, \epsilon_3 = 10, \epsilon_4 = 5$ and for the *Additional challenge*: $\epsilon_1 = 20, \epsilon_2 = 10, \epsilon_3 = 5, \epsilon_4 = 2.5$. Due to time constraints, the number of particles for each intermediary distribution was set to $N = 40$. This resulted in the following acceptance rates for each intermediary distribution: 97.56%, 66.67%, 16.06%, 3.68% for *Model 3* and 81.63%, 75.47%, 51.28%, 21.16% for the *Additional challenge*.

These results could be improved by increasing the number of particles and decreasing the tolerance level, however, this would result in significantly longer running times since one iteration of the algorithm lasts around 4-5 minutes. Furthermore, a wider prior distribution might be a fair consideration since the prior distribution used to obtain these results proved to be too narrow and 3 of the true parameter values (across both competitions) were outside the bounds. The bounds for the prior distribution were informed by the initial design used to train the GP regression estimates, so it is expected that predictions outside these bounds will not be very reliable.

Table 14 and Figures 21 and 22 present the MAP estimates obtained from both challenges.

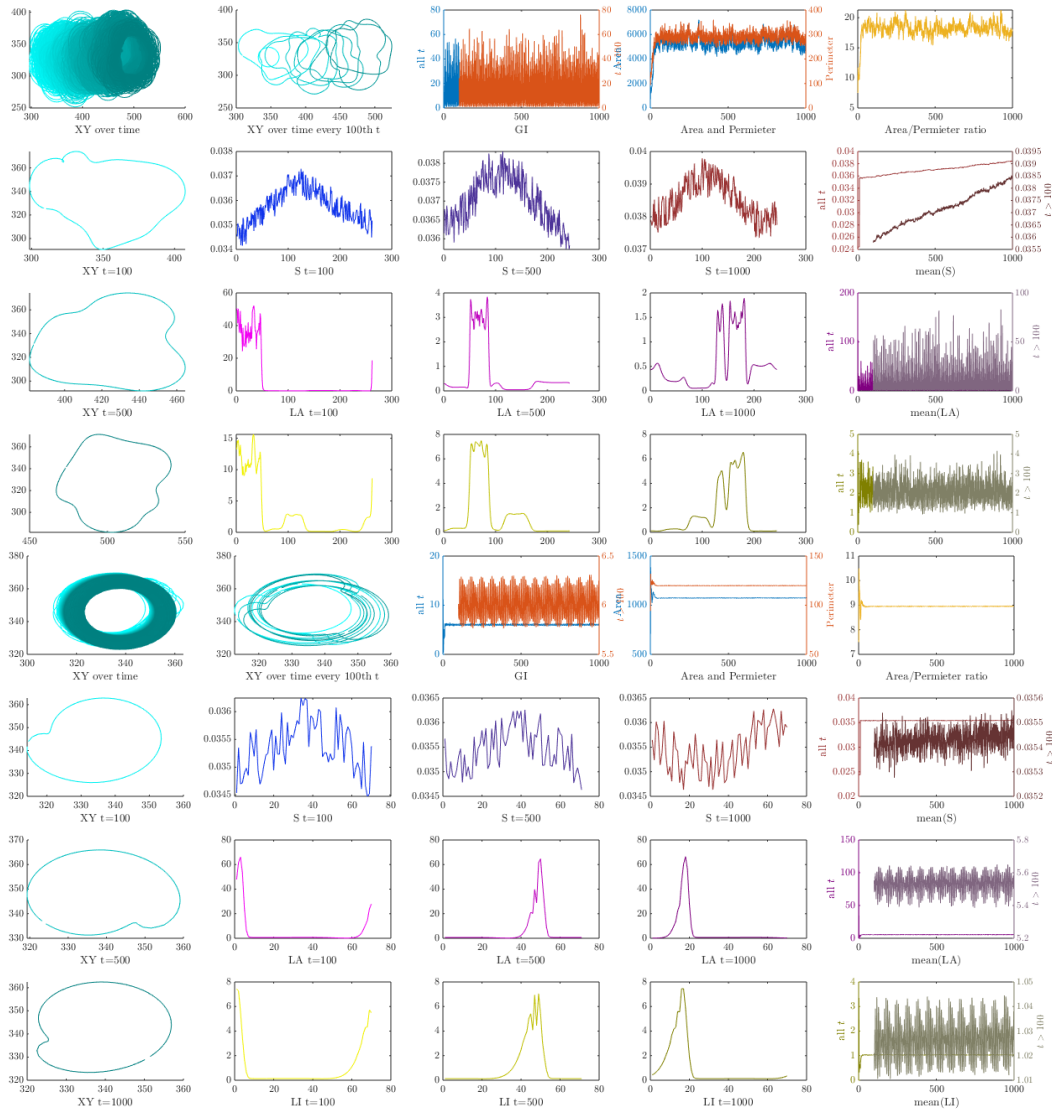


Figure 6: Outputs from two simulations run with different θ (top and bottom panel). Below, $\mathcal{T} = \{1, 2, \dots, 1000\}$. For each panel, from left to right, top row: cell membrane evolution in time for $t \in \mathcal{T}$ (the darker the shade, the later the time point); cell membrane evolution in time for $t = 1, 101, \dots, 901$; time series of GI for $t \in \mathcal{T}$ (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 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 $t \in \mathcal{T}$ (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 $t \in \mathcal{T}$ (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 $t \in \mathcal{T}$ (left axis) and for $t \geq 100$ (right axis).

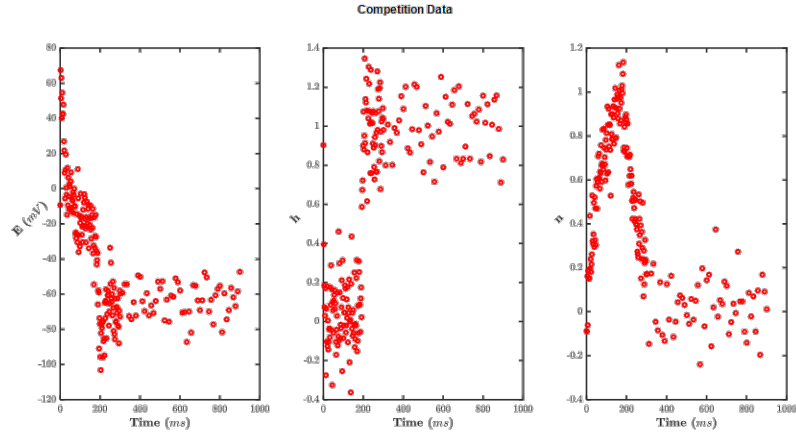


Figure 7: Timeseries plots of E (left), h (middle) and n (right).

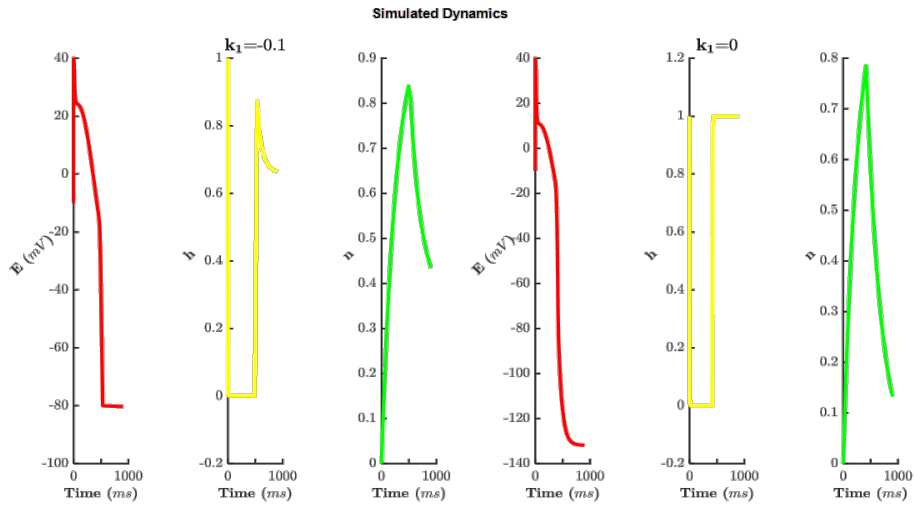


Figure 8: Simulated dynamics for two k_1 values below the reference value: $k_1 = -0.1$ (left panel) and $k_1 = 0$ (right panel). Each panel has contain timeseries plots of E (left), h (middle) and n (right).

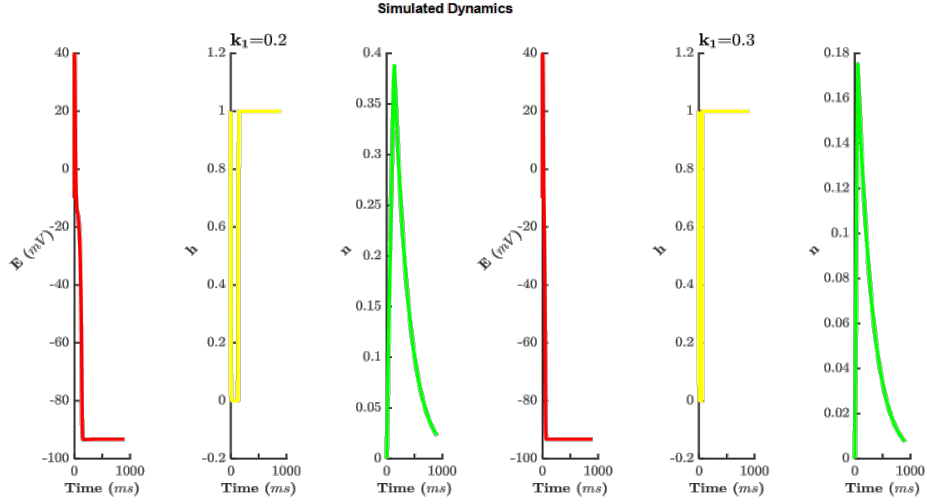


Figure 9: Simulated dynamics for two k_1 values above the reference value: $k_1 = 0.2$ (left panel) and $k_1 = 0.3$ (right panel). Each panel has contain timeseries plots of E (left), h (middle) and n (right).

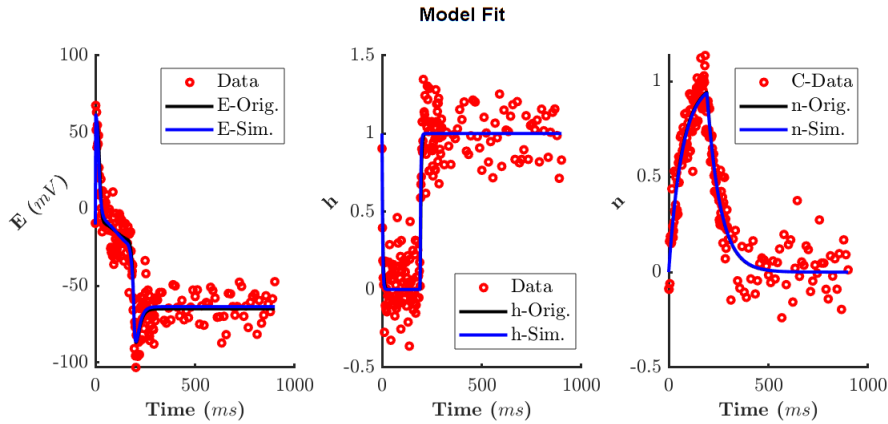


Figure 10: Original(solid black) and model(solid blue) fit of E , h and n using true and estimated parameters respectively while the measured data of E , h and n are the red dots.

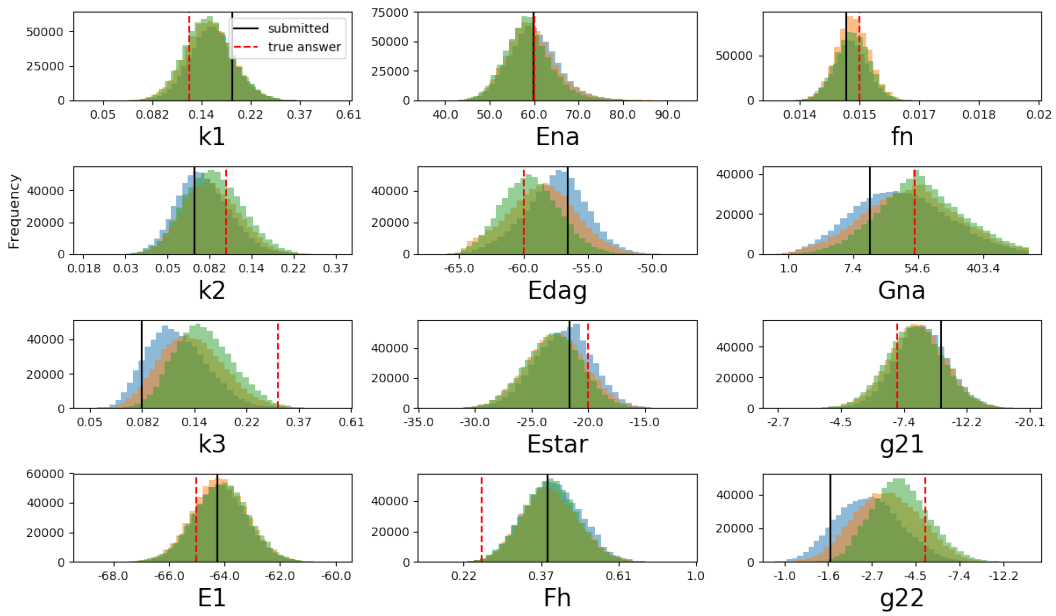


Figure 11: Marginal posterior distributions in the transformed parameter space, obtained from three independent MCMC chains. Black lines show the MAP estimate, which corresponds to the submitted parameter values. Red dashed lines indicate the values used to generate the competition data.

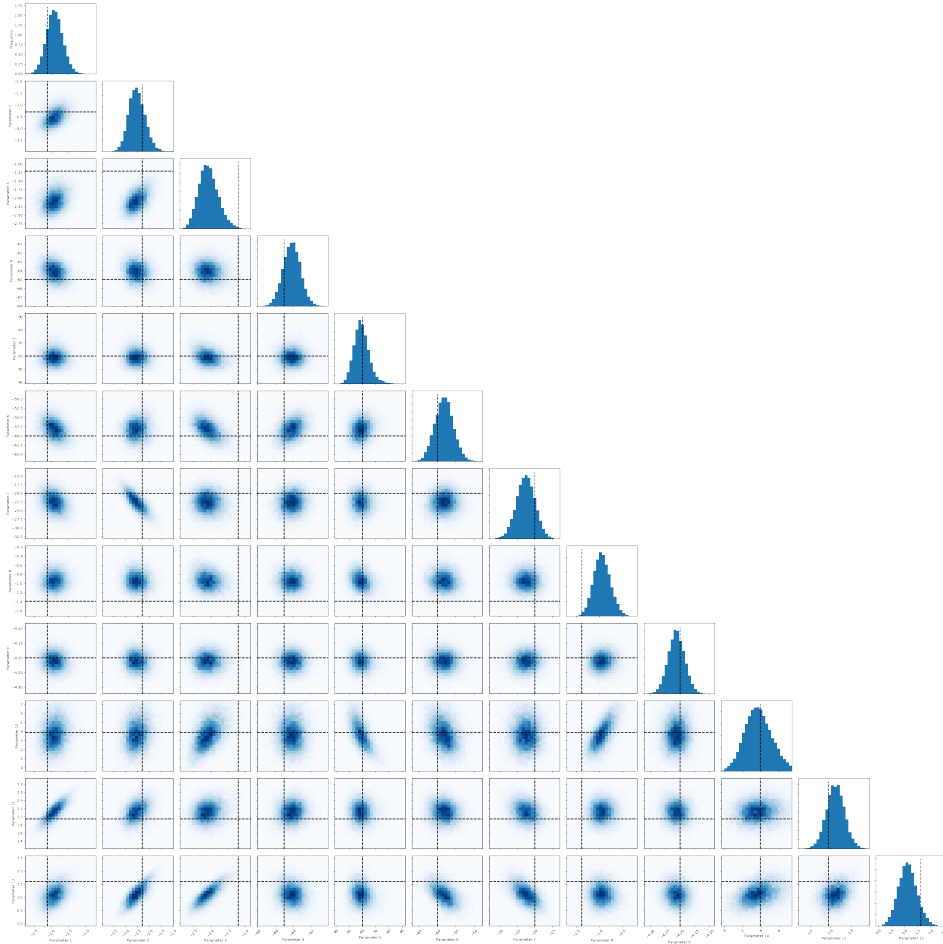


Figure 12: The 2D pairwise marginal plots for the posterior distribution in the parameter transformed space. One of the three independently run MCMC chains is shown. Black dashed lines show the true values in the competition.

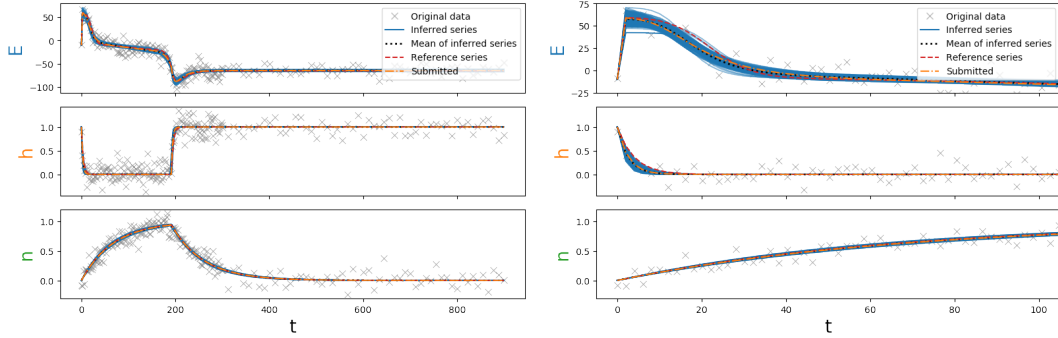


Figure 13: The posterior predictions of the model. The blue lines show the simulation outputs using the parameter samples from the MCMC chains, which represent our belief about possible model outputs underlying the given noisy data set, shown as grey crosses. The black dotted lines show the mean of the predictions, which should represent our best belief of the model output. The orange dash-dotted lines are our MAP estimate prediction, which are very close to the posterior predictive mean. Finally, the red dashed lines show the model output using the values used to generate the competition data. The panels on the right zoom in on the beginning of the simulation, showing the rapid upstroke of the action potential.

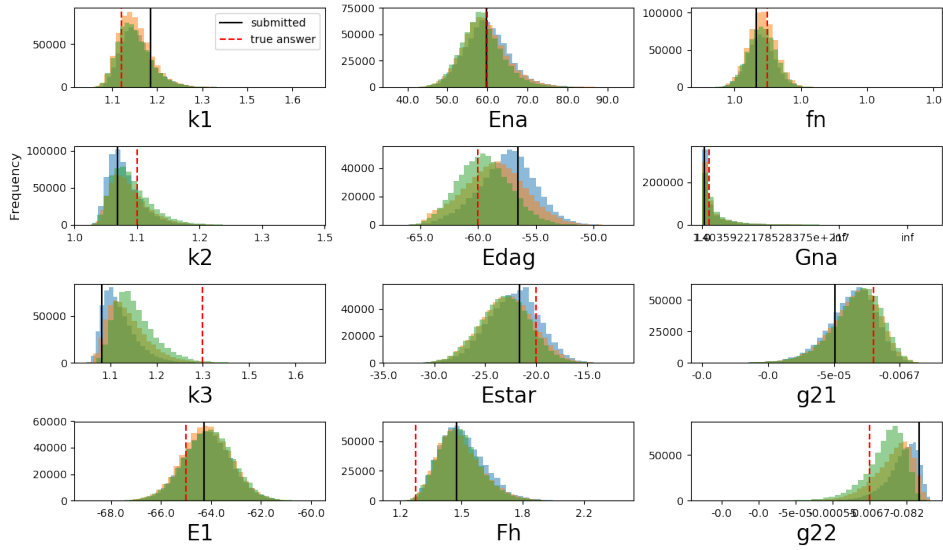


Figure 14: The 1D marginal plots for the posterior distribution in the model parameter space. The same three independently run MCMC chains in Figure 11 are shown. Black lines show the MAP estimate, which is our submitted parameter values. Red dashed lines indicate the true values in the competition.

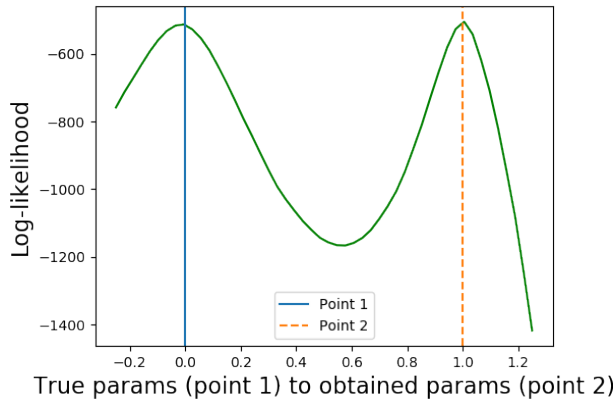


Figure 15: The 1D hyperline of the log-likelihood drawn from the true model parameter values to the obtained MAP model parameter values. A non-convex type hyperline suggests that the log-likelihood function is non-Gaussian and hence precision matrix approximation is not ideal.

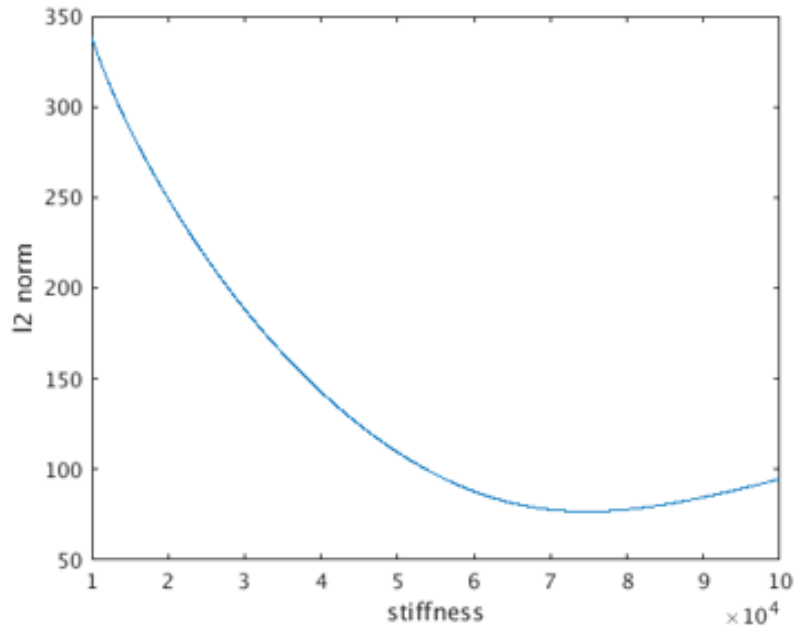


Figure 16: Residual sum-of-squares (RSS) against the stiffness parameter.

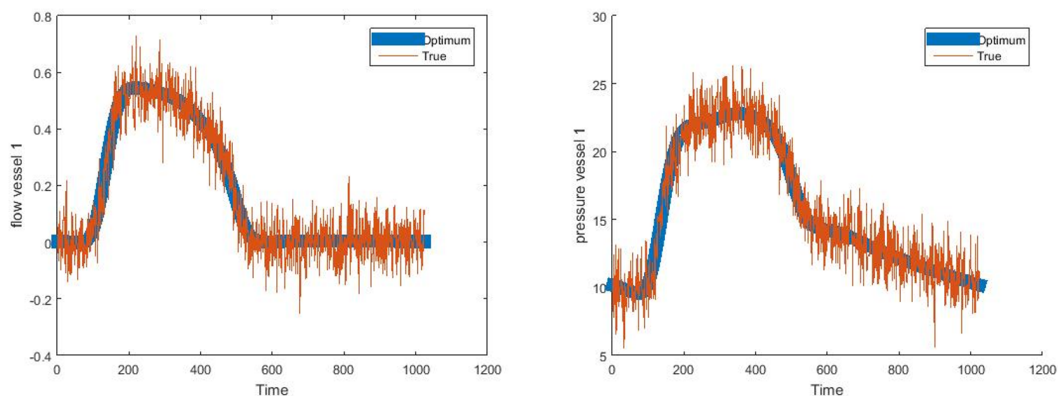


Figure 17: Optimised blood flow and pressure signals in the first pulmonary vessel (main pulmonary artery).

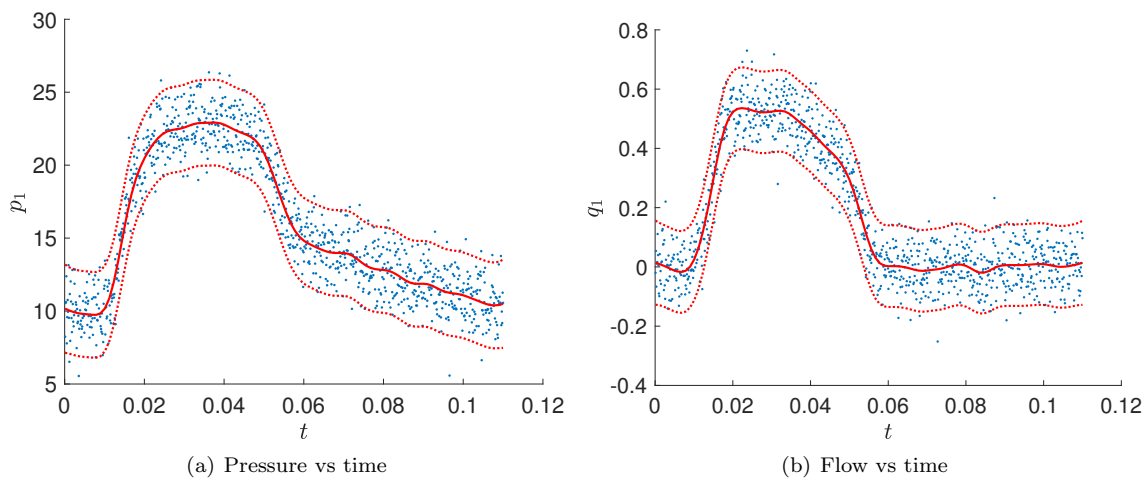


Figure 18: Univariate GPs for the first vessel. The observed data are shown as blue dots, the red line is the GP mean, while the dotted red lines denote the 95% confidence interval.

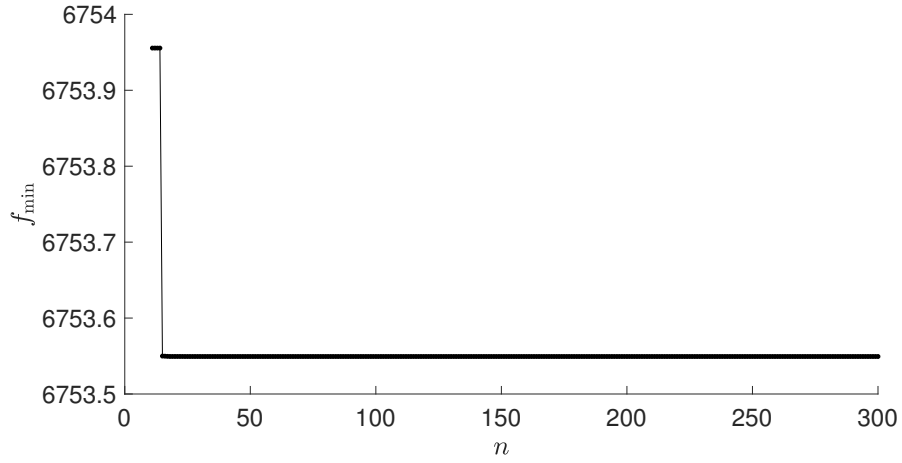


Figure 19: Optimization trace.

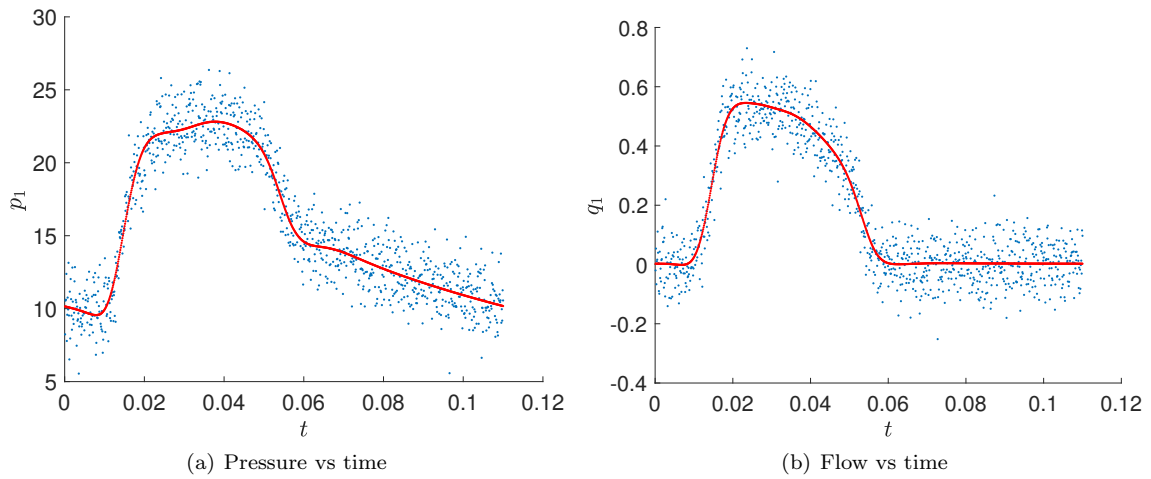


Figure 20: Best-fit pressure and flow curves in the first vessel using the the estimated parameter. The observed data are shown as blue dots, while the 1st and 4th columns of $\mathcal{M}(x_{\min})$ are shown as red curves.

Model 3					
	f_a	r_c	k_b	d_b	D_b
$\hat{\theta}_i$	0.0030	0.0541	0.0041	0.0142	0.0617
std($\hat{\theta}_i$)	(0.0010)	(0.0175)	(0.0010)	(0.0033)	(0.0132)
θ_i	0.0032	0.0500	0.0035	0.0130	0.0600
$\frac{ \hat{\theta}_i - \theta_i }{\theta_i}$	0.0730	0.0827	0.1840	0.0933	0.0290
	k_M	s_a	b_a	D_a	d_a
$\hat{\theta}_i$	0.2474	8.066e-5	0.1395	0.0299	0.0240
std($\hat{\theta}_i$)	(0.0433)	(0.0011)	(0.0417)	(0.0068)	(0.0029)
θ_i	0.3000	8.000e-5	0.3000	0.0250	0.0250
$\frac{ \hat{\theta}_i - \theta_i }{\theta_i}$	0.1754	0.0083	0.5351	0.1976	0.0395
Additional challenge					
	f_a	r_c	k_b	d_b	D_b
$\hat{\theta}_i$	0.0024	0.0801	0.0037	0.0167	0.0541
std($\hat{\theta}_i$)	(0.0010)	(0.0326)	(0.0011)	(0.0045)	(0.0204)
θ_i	0.0040	0.0900	0.0020	0.0100	0.0500
$\frac{ \hat{\theta}_i - \theta_i }{\theta_i}$	0.3921	0.1098	0.8286	0.6719	0.0816
	k_M	s_a	b_a	D_a	d_a
$\hat{\theta}_i$	0.2103	7.650e-5	0.1220	0.0366	0.0176
std($\hat{\theta}_i$)	(0.0635)	(0.0010)	(0.0458)	(0.0101)	(0.0105)
θ_i	0.1000	7.000e-5	0.0750	0.0150	0.0200
$\frac{ \hat{\theta}_i - \theta_i }{\theta_i}$	1.1034	0.0929	0.6267	1.4398	0.1201

Table 13: Submitted results: means $\hat{\theta}_i$ and standard deviations (in parentheses) for Model 3 and Additional Challenge, together with the ground truth values θ_i and relative absolute error. Shaded fields indicate the cases for which the ground truth values were outside the prior range. Table adapted from [18].

	f_a	r_c	k_b	d_b	D_b	k_M	s_a	b_a	D_a	d_a
Model 3										
$\hat{\theta}_i$	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
$\frac{ \hat{\theta}_i - \theta_i }{\theta_i}$	0.1718	0.3698	0.1562	0.3773	0.2540	0.3913	0.2591	0.6644	0.1850	0.4085
Additional challenge										
$\hat{\theta}_i$	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
$\frac{ \hat{\theta}_i - \theta_i }{\theta_i}$	0.4349	0.3192	0.5731	0.1522	0.3353	1.0725	0.3278	0.2430	0.5580	0.0575

Table 14: ABC SMC results submitted to the Cside competition: MAP estimates $\hat{\theta}_i$ and standard deviations (in parentheses) for Model 3 and Additional Challenge, together with the ground truth values θ_i and relative absolute error. Shaded fields indicate the cases for which the ground truth values were outside the prior range. The relative absolute error for *Model 3* is 3.237 and for the *Additional challenge* is 4.438.

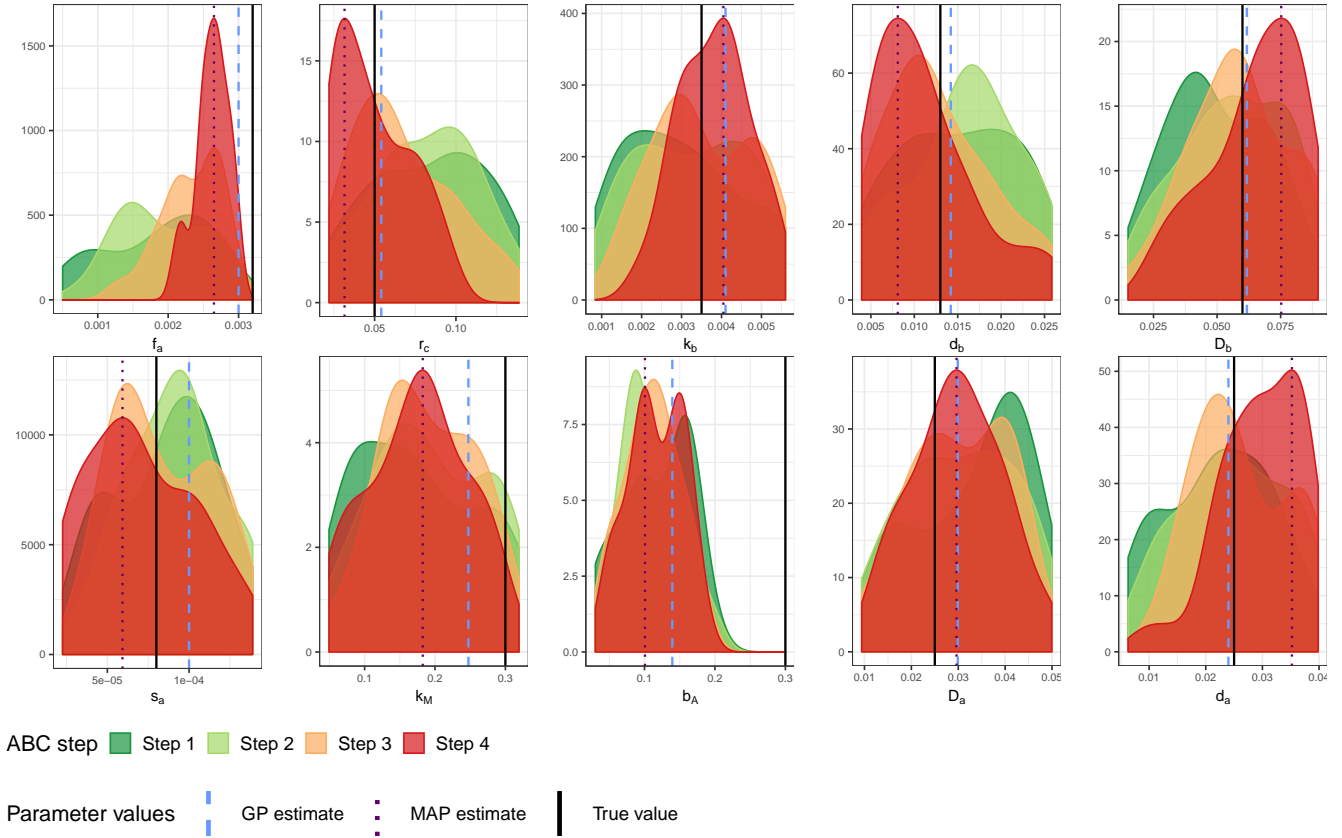


Figure 21: ABC SMC results for the pseudopod model, *Model 3*, for each of the ten unknown parameters: $f_a, r_c, k_b, d_b, D_b, s_a, k_M, b_A, D_a, d_a$. Each plot shows 4 intermediary posterior distributions, from Step 1 (green) to Step 4 (red), corresponding to decreasing tolerance levels $\epsilon_1 = 40, \epsilon_2 = 20, \epsilon_3 = 10, \epsilon_4 = 5$ and $N = 40$ particles. The final MAP estimates, obtained from the approximate posterior at Step 4 are shown as a dotted line and the true values as a continuous line.

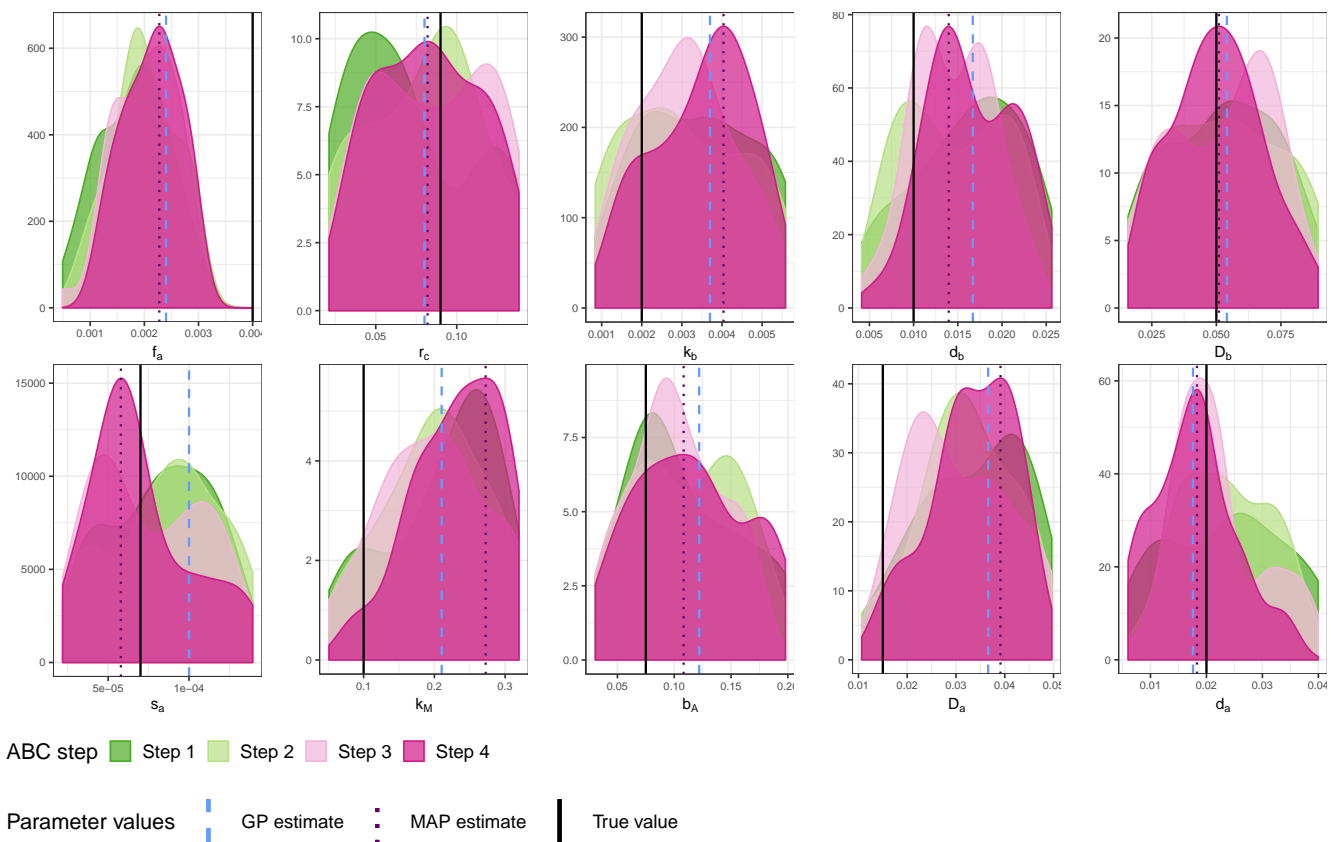


Figure 22: ABC SMC results for the pseudopod model, *Additional challenge*, for each of the ten unknown parameters: $f_a, r_c, k_b, d_b, D_b, s_a, k_M, b_A, D_a, d_a$. Each plot shows 4 intermediary posterior distributions, from Step 1 (green) to Step 4 (red), corresponding to decreasing tolerance levels $\epsilon_1 = 20, \epsilon_2 = 10, \epsilon_3 = 5, \epsilon_4 = 2.5$ and $N = 40$ particles. The final MAP estimates, obtained from the approximate posterior at Step 4 are shown as a dotted line and the true values as a continuous line.