Refining Combined Models with Gaussian Processes and Approximate Bayesian Methods: Applications in Integral Projection Models



Zhixiao Zhu

Keble College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2024

Acknowledgements

As I complete this thesis, I often think of the words written by Shi Tiesheng in *The Temple of Earth and Me*. In his essay, he described wandering alone in the garden, observing the changes of the seasons. Everything seemed unrelated to him, yet everything seemed to exist because of him. He wrote, “The tranquillity of each fallen leaf carries the mercy of time." This understated sentence has stayed with me ever since I first read it.

Over the years of my DPhil, I felt like I was sitting in my own "garden", thinking, questioning, and stubbornly trying to find answers in solitude. The garden was not large, but it always seemed endless. In spring, it filled me with hope; in summer, I believed I was nearing the finish line; in autumn, the wind scattered unfinished drafts, reminding me of unresolved problems; in winter, the snow buried every familiar path, leaving me unsure of where to go. At all times, the garden was silent, and I kept walking alone.

I thought this was a journey I had to face by myself. Late nights spent coding, analysing data, and starting over after failure felt like a series of solitary battles. Each step forward felt like a burden only I could carry. Exhaustion, isolation, and uncertainty about the future made the garden feel cold and empty. I often asked myself: what is the point of all this? Like Shi Tiesheng pushing his wheelchair through the Temple of Earth, I felt the path had no end, and the world outside seemed distant and disconnected.

Then, one night, I was at my desk, repeatedly revisiting my method, my foggy mind weighed down with exhaustion. My partner Xiao quietly placed a cup of hot tea on the desk without saying a word and left the room. The steam rose gently in the warm light of the desk lamp. I picked up the tea and took a sip, realizing that the cup had already gone cold—I had no idea how long it had been sitting there.

After finishing the tea, I went to the kitchen and saw there was still warm water in the kettle. Xiao hadn’t gone to bed; sitting nearby with a book, quietly flipping through the pages and occasionally glancing at the clock. I paused and asked, “Why are you still awake?” Xiao smiled and said, “Just waiting to see if you might want another cup.”

In that moment, I thought of Shi Tiesheng’s words: “Everywhere in the garden, there are the tracks of my wheelchair. And where there are tracks of my wheelchair, there are also the footprints of my mother.” For so long, I believed I was alone,

carrying all the weight myself. But as I walked with my head down, I hadn't noticed the footprints beside mine. Those small acts—bringing me tea, cooking dinner, sitting quietly nearby—were like sunlight filtering through the trees in the garden, silently illuminating every step I took.

The DPhil journey is long, long enough to make me forget the support around me. That night, I finally understood that I was never truly alone. People often fail to notice the quiet warmth around them until everything becomes still. It is these small moments of care that give us the strength to keep going.

As a child, I believed my parents were naturally capable of everything. I thought their support was simply part of life. But during my DPhil, I slowly came to realize that behind their "capability" were countless moments of uncertainty and struggle. Every time my parents said, "Take care of yourself" or "Don't push too hard" on the phone, they were standing outside my garden, careful not to ask too much, silently offering all the support they could. I know now that this thesis doesn't just hold the marks of my own efforts; it also carries their footprints.

My partner, Xiao, has been like a ray of sunlight in the garden. No matter how lost I felt, Xiao was always there to listen, to offer quiet encouragement, and to lighten the weight I carried. The solitude of my DPhil became less of a burden and more of a challenge we faced together.

To my friends, Hanwen Zhang, Qingze Gu, Zhongying Yang, Yuxin Fu, Qianyi Deng, Xi Lin, Yutong Lu, Daniel Phillips, Zhongyi Hu, Hanwen Xing, Linying Yang... Thank you all for teaching me to pause in the garden and look around. You laughed with me about unfinished plans and shared the small joys of life, bringing lightness and laughter into my work.

Lastly, I am deeply grateful to my supervisors David Steinsaltz and Maria Christodoulou. Over the years, your encouragement and guidance have been my driving force. "I'm impressed with how thoroughly you marked it up. You certainly know the material better than I do now!", "Congratulations, this is wonderful feedback!", "At present he is producing excellent work.", "His writing style and clarity has improved tremendously.", "Zhixiao has been making excellent progress on improving both the efficiency and the interpretability of his model", "Zhixiao worked very hard this term", "I am sure he will do great!", "As always, Zhixiao has impressed me with his creativity, hard work,innovation, and excellent attitude when dealing with research challenges." These words have stayed with me. They remind me that I was never truly struggling alone but walking toward something greater.

Looking back now, I think Shi Tiesheng was telling us that the tranquillity of the fallen leaf lies in how time transforms everything —its struggles, its growth, its descent — into a quiet grace. As I reflect on my journey, I see now that every step — however lonely or uncertain — was never wasted. Time has softened the edges of solitude and revealed, with quiet patience, the care and love that were there all along.

Abstract

Studying complex systems often involves combined models that integrate multiple sub-models to represent different processes. Traditional approaches, however, struggle to capture interactions between sub-models, as they typically rely on fitting each sub-model independently. This thesis introduces a novel framework that combines Gaussian Processes (GPs) and Approximate Bayesian Computation (ABC) to address these challenges, offering a flexible and efficient way to model complex systems.

The framework is applied to Integral Projection Models (IPMs), a widely used tool in population biology. By replacing traditional parametric sub-models with non-parametric GPs, it captures non-linear dependencies and provides probabilistic outputs. To account for interactions between sub-models, joint information is used as summary statistics within the ABC framework, enabling the inclusion of population-level insights that are often overlooked in traditional approaches.

To improve computational efficiency, an enhanced ABC-PMC sampler is introduced, which leverages pre-computed MCMC samples to reduce computational time from weeks to minutes. A systematic procedure for selecting summary statistics is also proposed, helping others apply ABC in their studies.

Through a simulation study and two real-world applications, the developed framework demonstrates great improvements in accuracy and flexibility over traditional GLM-based IPMs.

To support broader use, the Python package `ABC_GP_IPM` is developed. The package simplifies our proposed methods into several lines of `Python` code, making the method more accessible and allowing researchers to focus on their biological inquiries.

Contents

List of Figures **viii**

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Background . . . 3
1.3 Thesis outline . . . 6

2 Methods **9**
2.1 Integral projection models . . . 10
2.2 Gaussian Process models . . . 11
2.2.1 Vital ratess in IPMs . . . 11
2.2.2 GP model overview . . . 12
2.2.3 GP Regressions . . . 13
2.2.4 GP Models with Non-Gaussian Likelihoods . . . 14
2.2.5 GP hyperparameters . . . 15
2.2.6 Experimental Study: Parameter perturbation in GP models 17
2.3 Individual-based models . . . 21

3 Approximate Bayesian Computational methods **22**
3.1 ABC algorithms . . . 24
3.1.1 Non-iterative ABC methods . . . 24
3.1.2 Iterative ABC methods . . . 27
3.2 Summary statistics in ABC methods . . . 31
3.3 A systematic workflow for selecting summary statistics . . . 37
3.4 Modified ABC PMC . . . 41
3.5 Detailed explanations for the ABC-PMC Modifications . . . 47

4 ABC_GP IPM with Case study 1: *Cryptantha Flava* **56**
4.1 ABC_GP IPM . . . 57
4.2 The study species . . . 58
4.3 Simulation case study of the enhanced IPM . . . 60
4.3.1 Simulation setup . . . 61
4.3.2 Evaluation metrics . . . 62

4.3.3 Results 63
4.4 Application of the enhanced IPM with real datasets 73
4.4.1 Results 74
4.5 Discussion 80

5 A Python package: ABC_GP_IPM 83
5.1 Object-oriented programming in Python 86
5.2 Case study 2: *Ovis aries* 87
5.2.1 The study site 87
5.2.2 Interplay between Weather, Population density, and Vegetation resilience 88
5.2.3 Data collection and the Study dataset 89
5.2.4 Data analysis and Model settings 90
5.3 Getting start with ABC_GP_IPM 93
5.3.1 Initialisation 93
5.3.2 SS selection 94
5.3.3 Cache 96
5.3.4 ABC sampling 97
5.3.5 Forecasting 99
5.4 Case study 2: Results 100
5.5 Case study 2: Discussion and Future works 104
5.6 Discussion 104

6 Discussion and Future works 106
6.1 Algorithm discussion 107
6.1.1 ABC's role in revealing interactions detected among vital rates 107
6.1.2 Managing background knowledge integrated in ABC GP IPM 108
6.1.3 Employing MCMC samples as ABC priors 109
6.1.4 Lacking sufficiency in non-parametric modelling 111
6.2 Future works 112
6.2.1 A new sampler with a more appropriate structure 112
6.2.2 A new sampler with a less-restricted prior 118

Appendices

A Appendix for Chapter 2 123

B Appendix for Chapter 4 **128**
B.1 More details about the Simulation case study 128
B.1.1 A full description for candidate summary statistics and distance metrics 128
B.1.2 Implementation details and more results 131
B.2 More details about the Real case study 136
B.2.1 Candidate summary statistics 136
B.2.2 Implementation Details 137
B.3 Efficiency explorations and more results with the real-world dataset 141
B.3.1 Marginal posteriors in real case study 146

C Appendix for Chapter 5 **149**

References **151**

List of Figures

2.1 The plots display samples obtained from GP models with kernel function k_{Gau}. The left plot displays the samples obtained by varying the scale factor r from 1 to 6 while fixing the lengthscale l at 1. Conversely, the right panel illustrates the effect of increasing l from 1 to 6, while keeping r fixed at 1. 16

2.2 Given the age (y-axis) and logged size (x-axis) in the current year, the contour plots represent the predicted mean for logged size in the next year. Predictions were made through GP models with hyperparameters shown in Table A.1 in Appendix A. Plot (b) and (d) were produced by perturbing the lengthscale for the log size by the same **absolute** amount in (a) and (c) respectively. Purple dots represent for the values of covariates for the toy dataset. 18

2.3 Contour plots showing the posterior predictive variance of the GP models, given the same hyperparameter settings used in Figure 2.2. 19

3.1 Density histograms of ABC posterior approximations. The approximations were generated through ABC Rejection samplers with the Euclidean distance metric and two different bandwidth values: $h = 0.01$ and $h = 0.3$. The ABC acceptance kernel $\mathcal{K}$ (in Equation 3.1) was defined as $\mathcal{K}(\boldsymbol{d}(\boldsymbol{s}^{obs}, \boldsymbol{s}^*)) = \mathbb{1}_{\boldsymbol{d}(\boldsymbol{s}^{obs}, \boldsymbol{s}^*)<h} = \mathbb{1}_{|\bar{y}^{obs}-\bar{y}^*|<h}$ for plots on the left column, and $\mathcal{K}(\boldsymbol{d}(\boldsymbol{s}^{obs}, \boldsymbol{s}^*)) = \mathbb{1}_{|\bar{y}^{obs}-\bar{y}^*|<h}\mathbb{1}_{|v^{obs}-v^*|<h}$ for plots on the right column. The red curves indicate the true posterior distribution $\Gamma(6, 6)$. 33

3.2 Density histograms of ABC posterior approximations. These two histograms are identical to the plots on the bottom of Figure 3.1. The additional orange shadow areas are the ABC posterior approximations based on the weighting strategy, where the acceptance kernel is defined as $\mathcal{K}(\boldsymbol{d}(\boldsymbol{s}^{obs}, \boldsymbol{s}^*)) = \mathbb{1}_{(((\bar{y}^{obs}-\bar{y}^*)*w_1)^2+((v^{obs}-v^*)*w_2)^2)^{1/2}<h}$ with $h = ((0.3 * w_1)^2 + (0.3 * w_2)^2)^{1/2}$ to make the histograms comparable (from some sense). 35

4.1 Workflow chart illustrating the enhanced IPM construction process: This diagram demonstrates the integration of GP models and ABC in the development of IPMs. Grey arrows depict the steps involved in constructing a traditional Bayesian IPM. Blue arrows highlight the replacement of standard GLMs with GP models. Orange arrows illustrate the innovative application of the ABC methods to synthesise GP IPMs. Please note that all data points and content in this chart are for illustrative purposes only and do not represent any actual data values or practical implementations. 57

4.2 Predictions based on datasets D_{glm} and D_{gp}. Predicted λs, along with their corresponding density histograms (shown as shaded areas), were generated using 5,000 samples draw from the estimated posterior distribution of the three fitting methods. The red line represents the λ computed from the true discretized IPM. The pipe ('|') in the bottom of each plot marks the corresponding mean. 64

4.3 Predictive performance evaluations for the three model fitting methods applied to datasets D_{glm} (**left**) and D_{gp} (**right**). Different colours represent various settings of the growth model. The pipe ('|') in the bottom of each plot marks the corresponding mean. 65

4.4 Model fitting results for D_{glm}. The true and estimated marginal distributions for the five vital rates, complete with their 95% confidence intervals in matching colours and the corresponding scatters data points in blue. 65

4.5 Using the 5,000 posterior samples that generated estimated λ in Figure 4.2 left, we extracted the corresponding survival and reproduction models and evaluated them on 50 mesh points (representing z from 0 to 5). The survival–reproduction pairs shown in (A) and (B) were evaluated jointly within each posterior sample at each mesh point, so the joint structure between vital rates was preserved. **(A)** Survival–reproduction pairs from GLM and ABC_GP (each line corresponds to a posterior sample evaluated over the mesh points). **(B)** Survival–reproduction pairs from GP and ABC-GP. **(C)** Sample means of survival and reproduction rates at each mesh point, shown for GLM, GP, ABC-GP, and the true model. 66

4.6 **Left:** The ratio of P_c^{accep} and P_c^{prior} over simulation time c. The grey horizontal line cross the point representing $P_c^{accep}/P_c^{prior} = 1$. **Right:** The logarithmic ratio between the cumulative time spent on ABC-PMC, represented as t_c^{cumu}, and the expected implementation time, estimated as t_c^{prior}. The t_c^{cumu} is obtained by adding up individual times for each single iteration t_c until the current iteration c, $t_c^{cumu} = \sum_{i=0}^{c} t_i$. Instead of timing in real practices, t_c^{prior} was estimated using the formula $t_c^{prior} = (N_c/(P_c^{prior} * 10000290)) * 530.5$ based on the 10,000,290 particles (see Table 4.3). 71

4.7 The mean (dashed line) and median (solid line) for the 5 selected summary statistics, s_1 to s_5, over the ABC-PMC simulation time c ($x-$axis). The orange lines describing the information of summary statistics for accepted particles only, while the blues stand for all particles including those rejected. 72

4.8 Predicted distribution of λ for each ABC-PMC iteration. Vertical lines indicate the true population growth rate, λ, (in red) and corresponding predicted mean for each iteration. Note that, at step $c = 10$, where we applied the most restrict threshold, leading to a extreme high rejection rate and a limited yield of only 120 samples within a feasible time frame. Due to these constraints, the results from this step were excluded from our evaluations of predictive performance and algorithm efficiency. We present the findings about $c = 10$ here to illustrate the potential of the proposed method for achieving even better results. 73

4.9 Density histograms of estimated marginal posterior distributions for all hyperparameters under methods GP and ABC_GP. 75

4.10 Absolute differences between the observed and predicted population size (medians of the predictions). The models were trained using five datasets from years 2003-2004 to 2007-2008 (before the grey dashed vertical line). For **long-term** forecasting, the model used the initial population structure from Year 2003 as the starting point and generated predictions based on simulated populations from the previous year. For example, the predicted population size for 2005 was based on the simulated population for 2004 (which was generated using the true observed population for 2003). For **1-step** forecasting, the model used the true observed population structure from the previous year to generate predictions. That is, for example, to make a prediction for 2005, the true structure for 2004 was plugged into the IBM. 76

4.11 Two types of population size predictions, along with their corresponding 95% PI (shaded areas), were generated using GP, ABC_GP and GLM samples through 5,000 IBM simulations. The blue predictions were based on MCMC samples, while the orange and grey ones were based on ABC_GP and GLM posterior samples, respectively. The widths of the corresponding 95% PI are indicated by their respective colours on the left-hand side of the figure. The point estimations are represented by solid lines in their corresponding colours. The red line represents the true observed population size. 77
4.12 Histograms represent the density of logarithmic simulated population growth rates generated through **1-step** forecasting. The associated 95% prediction intervals are denoted by angle brackets, each in its corresponding colour, positioned at the top of each histogram. The red vertical line represents the true observed population growth rate. 78
4.13 Exactly the same setting as Figure 4.11, but additional simulations generated using two other sets of ABC posterior samples are depicted here. The new simulations are visually represented in green and purple. 79

5.1 A general workflow when applying GP and ABC to IPM. The processes within the green box have been encoded in our package. . 85
5.2 Two types of population size predictions, along with their corresponding 95% P.I. (shaded areas), were generated using GP, ABC_GP and GLM samples through 1,000 IBM simulations. The blue predictions were based on MCMC samples, while the orange and grey ones were based on ABC_GP and GLM posterior samples, respectively. The point estimations (means) are represented by solid lines in their corresponding colours. The red line represents the true observed population size. The model training process utilised data from 1987 to 1993 (before the grey dashed vertical line). The red line represents the true observed population size. 101
5.3 Absolute differences between the observed and predicted population size (means of the predictions) relative to the true observed population size. 102
5.4 Histograms represent the density of simulated population growth rates generated through **1-step** forecasting. The associated 95% prediction intervals are denoted by angle brackets, each in its corresponding colour, positioned at the top of each histogram. The red vertical line represents the true observed population growth rate. 103

A.1 Given the current age (y-axis) logged size (x-axis), the contour plots represent the predicted mean for logged size in the next year. Predictions were made through Gaussian process models with hyperparameters shown in Table A.2 in Appendix A. Plot (b) and (d) were produced by perturbing the same hyperparameter with the same **proportional** amount in (a) and (c) respectively. Purple dots represent for the values of covariance for the training data points. . 124

A.2 Contour plots for the predicted variance corresponding to the predicted mean shown in Figure A.1. 125

A.3 The GP kernel values for data points $(\boldsymbol{x_0}, \boldsymbol{x_j})$ for $j = 1, ..., 181$. The calculations were performed using Gaussian kernel with four different sets of hyperparameters, which were used in the models that generated the prediction surface depicted in Figure 2.2. The green and red lines represent the kernel values computed before and after perturbation, respectively, for model **M1**, while the blue and orange lines represent those for model **M2**. The set of values $\{\boldsymbol{x_i}\}_{i=0,...,181}$ was obtained through the meshgrid function, resulting in periodic patterns in the plots. The two plots are essentially the same, except that different regions have been highlighted. Notice that, this figure only illustrates how perturbations influence the GP covariance kernels. These changes in covariance also impact the predictive distributions. However, the prediction surface's response depends on the specific dataset. As a non-parametric method, GP models rely on the positions of the data points to evaluate these effects. Using the conditional density formula for a multivariate normal (MVN) distribution, one may calculate how covariance changes influence predictions in GP regression models. 127

B.1 GLM based IPM predictions with a larger sized training dataset, D_{glm2}. Predicted λs, along with their corresponding density histograms (shown as shaded areas), were generated using 5,000 samples draw from the estimated posterior distribution of GLMs. The red line represents the λ computed from the true discretized IPM. The pipe ('|') in the bottom of each plot marks the corresponding mean. 134

B.2 Model fitting results for D_{glm} at simulation step 5 and 6. The estimated marginal distributions for the five vital rates, complete with their 95% confidence intervals in matching colours and the corresponding scatters data points in blue. 134

B.3 Using the 5,000 samples that generated predicted λ in Figure 4.3 left, we extracted the corresponding vital rate models and evaluated them on 50 mesh points (representing size from 0 to 5). Sample means of vital rates at each mesh point, shown for GLM, GP, ABC-GP, and the true model. 135

B.4 **Left:** The ratio of P_c^{accep} and P_c^{prior} over simulation time c. The grey horizontal line cross the point representing $P_c^{accep}/P_c^{prior} = 1$. **Right:** The logarithmic ratio between the cumulative time spent on ABC-PMC, represented as t_c^{cumu}, and the expected implementation time, estimated as t_c^{prior}. The t_c^{cumu} is obtained by adding up individual times for each single iteration t_c until the current iteration c, $t_c^{cumu} = \sum_0^c t_c$. Instead of timing in real practices, t_c^{prior} was estimated using the formula $t_c^{prior} = (N_c/(P_c^{prior} * 10000190)) * 2541$ based on the 10,000,190 particles (see Table B.7). 142

B.5 The mean (dashed line) and median (solid line) for the 25 summary statistics over the ABC-PMC simulation time c ($x-$axis). The orange lines describing the information of summary statistics for accepted particles only, while the blues stand for all particles including those rejected. **s1** to **s5** represent the five types of summary statistics, **t0**, **t1**, ..., **t4** indicate year 2003-2004, 2004-2005, ..., 2007-2008. 143

B.6 Density histograms of estimated marginal posterior distributions for all hyperparameters under methods GP and ABC_GP. 144

B.7 We created 1,000 simulations for each of 200 samples drawn from the ABC posterior distribution. The median was then computed from the summary statistics of each set of simulations, yielding 200 median values. These medians were used to construct the **trace** seen in each plot. The **green** dot in each plot represents the overall median of these 200 medians. Meanwhile, the **orange** and **red** dots denote the median of the simulated summary statistics vectors for the accepted particles only, and for all particles (including rejected ones) during the ABC-PMC sampling process, respectively. The labels 's1' through 's5' represent the five different types of summary statistics used in our analysis. The labels 't0', 't1', through 't4' correspond to the years from 2003-2004 to 2007-2008. 145

B.8 Posteriors across five vital rates and nine years under three modelling methods (GLM, GP, and ABC_GP). For posterior visualisation, we fixed age at 2, discretised logged size into 50 mesh points between 0 and 5, and included the real year-specific climate information as model inputs. In the figure, each row represents a different vital rate, and each column corresponds to a different year. 148

C.1 Histograms represent the density of simulated population growth rates generated through **Long-term** forecasting. The associated 95% prediction intervals are denoted by angle brackets, each in its corresponding colour, positioned at the top of each histogram. The red vertical line represents the true observed population growth rate. 150

1 Introduction

Contents

1.1 Motivation **1**
1.2 Background **3**
1.3 Thesis outline **6**

1.1 Motivation

Giant systems have become increasingly common across scientific research (Caswell 2000; Siciliano et al. 2008; Ellner, Childs, et al. 2016; Figueiredo et al. 2018; Sato 2023). To study these complex systems, researchers often use *combined models*, breaking down the system into smaller sub-models, each representing a specific process within the larger system. This approach typically starts with independent fitting of each sub-model, then combines them to gain a holistic understanding of the system as a whole. This "fit individually, then integrate" approach is prevalent across various fields:

1. In robotic systems, accurate manipulation requires precise calibration of various subsystems. It involves initially using simpler models to calibrate basic components, followed by more complex models to refine the calibration

and capture intricate interactions between the robot's mechanical and sensory systems (Sato 2023).

2. In energy system simulations, models are often calibrated for multiple years using historical data. Each year's data is used to fit individual linear regression models. These models are then integrated into a multi-year model that accounts for annual variations, such as changes in weather conditions and other systemic factors over time (Figueiredo et al. 2018).

3. In biology, to understand population dynamics over time, models integrate various vital rate sub-models, such as survival, growth, and reproduction, each calibrated independently using field data. The sub-models are then combined to project the population structure over time, accounting for the interactions between different life stages and environmental factors (Caswell 2000; Ellner, Childs, et al. 2016).

While widely used, this "fit individually, then integrate" approach has notable limitations. The main challenge lies in its assumption — that accurately modeling each component will yield a reliable representation of the entire system. When fitted separately, sub-models do not account for interactions that naturally arise. Such interactions — arising from dependencies between processes or responses to shared conditions — can lead to system-level behaviours that isolated model fitting fails to consider. Models lacking such joint structure, thereby, may not fully capture the complexity of those giant systems.

Joint modeling of all sub-models within a single, fully integrated framework might potentially capture these dependencies more effectively. However, constructing such a comprehensive "giant model" is generally infeasible. Complex systems with (potentially) many interacting components often lack well-defined likelihood functions, making it challenging to specify a unified framework representing the entire system. Without clear likelihoods and manageable computational demands, creating a fully integrated model becomes impractical.

In this thesis, we explore an approach that aims to capture system-level interactions within combined models more effectively, while addressing practical considerations. We demonstrate our method using a popular population model in biology, *Integral Projection Model*. While our example demonstrates the method's effectiveness in ecology, it holds potential for applications across various fields that require complex combined models.

1.2 Background

Life cycles define an organism's journey from birth to death, highlighting key transitions like from juvenility to adulthood and from non-breeding to breeding. These stages are quantified mathematically by vital rates, like survival and growth rates. Influenced by environmental conditions and inherent traits, the rates interact to optimise life strategies while inherently constraining each other, illustrating the biological trade-offs present in nature (Stearns 1989; McNamara and Houston 1996; Obeso 2002). These demographic processes and trade-offs sculpt unique life paths for individuals. Over time, these paths converge, forming cohorts and populations, shaping the evolutionary history of the species.

Integral Projection Models (IPMs) serve as a fundamental tool in these demographic studies, offering key insights into natural selection and improving predictions of population trends (Picó et al. 2008). Demographic studies are inherently context-dependent, with variations across species and research objectives. To cater to these diverse requirements, IPMs have evolved to incorporate more detailed multi-dimensional state variables (Childs et al. 2003; Coulson, Tuljapurkar, et al. 2010; Stubberud et al. 2019), and factors like population density (Rose et al. 2005; Rebarber et al. 2012; Coles et al. 2023) and environmental stochasticity (Childs et al. 2004; Dahlgren, Bengtsson, et al. 2016; Félix-Burruel et al. 2024). These developments reflect the ongoing efforts to tailor IPMs to the intricate realities of ecological systems. Nevertheless, current IPMs still encounter challenges that compromise their abilities. We are presenting three key obstacles that require attention and resolution.

(1) Inadequacy of base vital rate models: Traditional IPMs typically employ (generalised) linear and mixed-effects models to estimate vital rates (M. R. Easterling et al. 2000; Ellner and Rees 2007), favoured for their mathematical tractability and straightforward interpretability. While Bayesian extensions have been developed to better address uncertainties (Rees and Ellner 2009; Merow, Latimer, et al. 2014; Messerman et al. 2023), both parametric schools — Bayesian and Frequentist — tend to oversimplify or misrepresent the relationships between vital rates and state variables as linear or fixed, despite substantial evidence suggesting otherwise in nature (Lingjærde et al. 2001; Martorell and Martínez-Ballesté 2019; Bonnaffé and Coulson 2022). Non-parametric approaches like splines (Dahlgren, García, et al. 2011; Ellner, Childs, et al. 2016; Macdonald et al. 2023) provide an alternative when linearity is inappropriate, but their inability to provide probabilistic outputs hampers their effectiveness. This limitation shifts the focus from a broader distribution that encapsulates inherent variability of vital rates to a narrow, precise, but potentially misleading point estimate, thereby restricting the depth and accuracy of ecological modeling (Ellison 2004; Hobbs and Hilborn 2006; Elderd and Miller 2016).

(2) Neglect of interactions among vital rates: Traditional IPM strategies often model vital rates separately before integration into the comprehensive IPM, assuming that accurately modeling each rate will result in a reliable overall population model. Such a piecemeal approach stands in contrast with the reality that vital rates are often interdependent, like reproductive costs and maturity thresholds (Kelly 1992; Hilborn et al. 2006; Struckman et al. 2019). While some studies have employed Bayesian frameworks to try to detect such interactions through the joint distributions of multiple vital rates (Elderd and Miller 2016), the likelihoods for the vital rates typically share no parameters, making their joint optimisation equivalent to maximising each separately (Plard et al. 2019).

Interactions among vital rates can occur at different levels — ranging from interactions within individuals due to biological heterogeneity to those impacting the entire population (Vindenes et al. 2008). The IPM, as a population model, may primarily focus on those that manifest at the population-level. Relying solely

on individual-level data to independently inform vital rates can risk overlooking these broader interactions. It may be beneficial to move beyond the limitations of individual-level data and incorporate population-level insights.

For example, grouping individuals based on their status — such as growth or shrinkage among breeders and non-breeders — enables the capture of aggregated effects of vital rates across different groups. This kind of population-level information can offer new perspectives on how vital rates might collectively influence population dynamics — providing insights that traditional, independent analyses in IPM might overlook, without the need for additional datasets.

(3) Rigidity of model structures: Lastly, the present-day IPM structures are rigid, often complicating the integration of users' empirical insights and limiting their flexibility to address new research queries. Scientists gain valuable background knowledge about their research species through years of dedicated investigation. Incorporating their expertise could hold immense potential for achieving more realistic and informed population models. However, translating detailed biological knowledge into statistical frameworks remains challenging, often complicated by the intricacies of formulations, coding, and the practical implementation of sophisticated, context-specific models. Current IPMs generally lack the flexibility to incorporate such detailed knowledge, resulting in a gap between model capabilities and biological realities. Consequently, there is a critical need for more adaptable IPM frameworks that facilitate the integration of diverse biological data and insights, empowering scientists with a more accessible approach to implement customised IPMs, even in complex scenarios.

In response to the limitations, we propose a novel methodology that enhances the robustness of the IPMs. Our method begins by replacing traditional base models with a Bayesian non-parametric model: the *Gaussian process* (GP) model. This model introduces a new level of flexibility — capturing complex dependencies without predefined functional forms and providing probabilistic outputs to better reflect natural variability (C. K. Williams and Rasmussen 2006). We further incorporate an *Approximate Bayesian Computation* (ABC) sampler, enabling researchers to infuse

their specific expertise during model synthesis. The ABC eliminates the complexity of translating intricate biological insights into statistical models, providing users a nearly automated solution to customise their own version of IPMs. As long as the interested scenarios can be represented through a data-generating process, researchers can develop realistic and applicable IPMs that align with their desired scenarios. Our approach utilises summary statistics in ABC that incorporate population-level data, ensuring vital rates' joint behaviours are captured at the population scale. The proposed method presents a reasonable and viable solution to the current limitations of IPMs and offers users increased flexibility to tailor their own version of population models.

In comparison to existing methods, Fung et al. 2022 recently introduced a hierarchical modeling framework for IPMs that considers interdependencies among vital rates. However, this approach still relies on parametric assumptions, making the results sensitive to the chosen model structures. On a related note, Plard et al. 2019 offered a model that integrates population-level data without employing ABC, using parametric assumptions to bridge individual- and population-level data. This simplifies implementation but at the cost of a fully data-driven approach, potentially introducing biases from the initial model assumptions.

1.3 Thesis outline

1. This thesis begins by introducing foundational models integral to our study, including IPMs, GP models, and Individual-Based Models (IBMs) in Chapter 2. We also present an experimental study that explores parameter perturbation in GP models, establishing a basis for the work in Chapter 3.

2. In Chapter 3, we provide an in-depth introduction to the foundational concepts of ABC, with a focus on their acceptance kernel $\mathcal{K}$ to thoroughly explain various ABC methods. This chapter also details the role of summary statistics within ABC frameworks and their importance in guiding the inference process.

After that, recognising the inherent challenge of selecting appropriate summary statistics in novel ABC applications, we propose a systematic procedure tailored to navigate this complexity effectively when applying ABC and GP to IPMs.

Furthermore, we introduce an enhanced ABC-PMC sampler for combined models using a 'drop-in' approach. Our method uses pre-computed MCMC samples to define the parameter space, enabling efficient navigation through high-dimensional combined models, greatly improving both accuracy and computational efficiency. Specifically, the computational time for 10,000 sampling attempts is reduced from at least 20 days to under one minute once MCMC samples are pre-computed and stored. Our findings demonstrate the potential of the modified ABC-PMC algorithm in managing combined models effectively, particularly those involving computationally intensive sub-models like GP models and other machine learning techniques.

3. In Chapter 4, we develop and demonstrate a novel IPM that integrates GP models with our modified ABC-PMC sampler. GP models imbue IPMs with adaptability across diverse scenarios, mitigating issues of model structural sensitivity. The incorporation of ABC methods empowers IPMs to integrate additional population-level information without the need for extra datasets. The information can be user-modified, facilitating the tailoring of IPMs to specific requirements or domain-specific knowledge.

   Next, using simulated datasets with known 'truth', we evaluate our method by comparing traditional IPMs to our enhanced IPMs using simulated datasets with known 'truth'. This evaluation validates the method's robustness and reliability across popular population metrics under controlled conditions. Additionally, we examine the efficiency of fitting ABC_GP IPMs through the ABC-PMC algorithm, revealing a pattern of increasing accuracy across iterations. This provides key insights into why our method produced noticeably improved results.

With insights gained from simulations, we then extend our methodology to real-world data from *Cryptantha flava*. In the real-world case study, while ensuring pronounced accuracy in point estimates, the ABC_GP IPM approach provides a more honest representation of uncertainties in predictions, particularly when the true relationship between predictors and the response is complex or unknown.

4. In Chapter 5, we introduce `ABC_GP_IPM`, a `Python` package that offers a streamlined, user-friendly interface for selecting ABC summary statistics and implementing the ABC-PMC sampler when constructing the novel IPM, making our proposed methods more accessible to other researchers. We further demonstrate the utility and performance of our package through another case study involving real-world biological data, showcasing how our tool simplifies complex modeling processes and data management in IPMs.

5. The report concludes in Chapter 6 with insights on the interpretation of our proposed method and future research directions.

2
Methods

Contents

2.1 Integral projection models 10
2.2 Gaussian Process models 11
2.2.1 Vital ratess in IPMs 11
2.2.2 GP model overview 12
2.2.3 GP Regressions 13
2.2.4 GP Models with Non-Gaussian Likelihoods 14
2.2.5 GP hyperparameters 15
2.2.6 Experimental Study: Parameter perturbation in GP models 17
2.3 Individual-based models 21

This chapter introduces the foundational models and methodologies in this thesis. We begin with Integral projection models (IPMs), which are widely used for modeling population dynamics and serve as a key framework in this study (Section 2.1). Next, we delve into Gaussian Process (GP) models (Section 2.2), focusing on their potential to replace traditional Generalized Linear Models within IPMs. This section provides an overview of GP models, explores their applications to vital rates in IPMs, and discusses extensions to handle non-Gaussian likelihoods and hyperparameter tuning. To further investigate the sensitivity analysis in GP models, we include an experimental study on parameter perturbation (Section 2.2.6), which explores how changes in hyperparameters affect model behaviour.

This study forms a basis for selecting summary statistics in ABC in Chapter 3. Finally, Section 2.3, we introduce Individual-Based Models, a simulation approach that tracks the life trajectories of individuals in a population. We employ the method to generate simulated datasets in this thesis.

2.1 Integral projection models

IPMs generalize the standard matrix population models (Caswell 2000), allowing populations to be described using continuous state variables. This approach can be more biologically sensible, when there are no natural breakpoints to discretise state variables. IPMs assume that the individuals in the target population can be completely summarised by an attribute z, which can be multi-dimensional with both discrete and continuous variables (like age-size) (Rees, Childs, J. C. Metcalf, et al. 2006; Ellner and Rees 2006). The distribution of individuals at time t with attribute z is denoted by $n(z,t)$. With this, in IPMs, the population at the next time step with attribute z' can be described by

$$\begin{aligned} n(z',t+1) &= \int \mathcal{T}(z'|z)n(z,t)dz \\ &= \int (P(z'|z) + F(z'|z))n(z,t)dz, \end{aligned}$$

i.e. individuals within state z go to state z' through a 'transition function' $\mathcal{T}(z'|z)$. In this thesis, we assume that individuals contribute to population growth only through two systems: the *survival-growth kernel*, $P(z'|z)$, and the *reproduction kernel*, $F(z'|z)$. These sub-kernels can be further decomposed into individual *vital rate functions*, such as survival and growth, to align with the actual life cycle events of the study species.

For example, in our case study of *Cryptantha flava* (Chapter 4), depending on a vector of environmental factors $\boldsymbol{\omega}$, $F(z'|z,\boldsymbol{\omega})$ starts with considering the probability of breeding, $p_f(z|\boldsymbol{\omega})$; conditional on flowering, a state z individual would produce an average of $n_f(z|\boldsymbol{\omega})$ flowering stalks; each stalk leads to a certain count of seeds establishing and growing to size z' by the next census, $r_{est}r_{size}(z')$. That is, $F(z'|z,\boldsymbol{\omega}) = p_f(z|\boldsymbol{\omega})n_f(z|\boldsymbol{\omega})r_{est|\boldsymbol{\omega}}r_{size}(z'|\boldsymbol{\omega})$. The survival-growth kernel, $P(z'|z,\boldsymbol{\omega})$, considers

the reproduction's influence on growth dynamics. It assumes that an individual with current state z will survive to the next year with probability $s(z|\boldsymbol{\omega})$, but reproducing or not would lead it to different fates in growth, $G_f(z'|z,\boldsymbol{\omega})$ or $G_{nf}(z'|z,\boldsymbol{\omega})$. That is, $P(z'|z,\boldsymbol{\omega}) = s(z|\boldsymbol{\omega})p_f(z|\boldsymbol{\omega})G_f(z'|z,\boldsymbol{\omega}) + s(z|\boldsymbol{\omega})(1-p_f(z|\boldsymbol{\omega}))G_{nf}(z'|z,\boldsymbol{\omega})$. The sequence of life cycle events in these proposed sub-kernels, $F(z'|z,\boldsymbol{\omega})$ and $P(z'|z,\boldsymbol{\omega})$, align with the actual census process conducted in Salguero-Gomez et al. 2012.

2.2 Gaussian Process models

2.2.1 Vital ratess in IPMs

In IPMs, vital rate functions (i.e., those sub-models defined within $P(z'|z)$ and $F(z'|z)$) serve as the critical link between individual-level observations and population-level projections. These functions capture and summarize variations and responses among individual observations, which are then incorporated into sub-kernels P and F to facilitate population projections. As highlighted by Merow, Dahlgren, et al. 2014, "IPM predictions can be only as good as the assumptions and inferred transitions of these vital rate models". The vital rates thereby need to be fitted with caution to avoid misleading population-level inferences.

Traditionally, vital rates in IPMs have been parameterized using conventional statistical methods, like linear regressions for growth and logistic regressions for flowering probability etc.. These parametric methods assume that the data points follow pre-defined functions, and then determine the optimal parameters within these functional forms. While effective in some scenarios, these methods can be restrictive and misleading when the true relationships in the data deviate from the pre-assumed models — a situation that commonly occurs in the real life.

Today, modern computational advancements enable models to autonomously determine the most suitable functions from datasets, eliminating the need for pre-assumed functional forms. One of such methods is the Gaussian Process (GP) models, also known as *Kriging method* in geostatistics. Unlike some other machine learning methods, GP models are well-established and provide clear and interpretable frameworks (C. K. Williams and Rasmussen 2006). Although GPs have

been increasingly applied in ecological studies, like species distribution modeling (Golding and Purse 2016; Ingram et al. 2020), to the best of our knowledge, their integration into IPMs remains unexplored. In IPMs, Dahlgren, García, et al. 2011 have demonstrated that the cubic splines have a relatively good performance when linearity assumption between vital rates and state variables cannot be satisfied. GP models can be viewed as a more flexible extension of spline-based methods (see Kimeldorf and Wahba 1970; Bay et al. 2016), offering greater potential to capture complex non-linear relationships in IPMs.

2.2.2 GP model overview

In the context of GPs, instead of having a finite set of random variables, we think of an infinite set of random variables. Each finite subset of these variables follows a multivariate normal distribution (MVN). Consider a finite realisation of a GP, f_{GP}, defined by the hyperparameter $\theta \in \mathcal{P}$:

$$f_{GP}|\theta = L_\theta x, \quad x \sim \mathcal{MVN}(0, I),$$

where I is an identity matrix and L_θ is the Cholesky decomposition of the GP kernel matrix $K_\theta = L_\theta L_\theta^\top$. By considering x as infinite-dimensional and $L_\theta x$ as a linear transformation applied to x, we can extend the notion of GPs to distributions over functions. This means that instead of just modeling distributions over finite-dimensional random vectors, GPs allow us to model distributions over an entire space of functions. This formulation highlights GPs as Bayesian non-parametric models, providing a flexible distribution over numerous potential functions that align well with the observed data, rather than adhering to a predetermined distribution shape.

The kernel matrix K_θ represents the covariance structure of the GP. The strength of GPs lies in their ability to measure data point similarities through the kernel K_θ. Generally speaking, GP models assume that data points closer together in the input space are more similar in the output space. The input space similarities are algebraically transformed to reflect output space correlations, enabling GPs to provide a predictive distribution for new data points based on observed datasets.

To mimic such complex 'correlations' observed in real data, GP models potentially have an infinite number of parameters, allowing the size of the models to grow with the data size (Y. Li et al. 2019). Although GPs are fundamentally non-parametric, they incorporate hyperparameters θ to regulate model complexity, thereby preventing over-fitting.

2.2.3 GP Regressions

Let $\mathcal{D}$ denote the dataset at hand, $\mathcal{D} = (X, \mathbf{y}) = \{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in \mathcal{X}^d, y_i \in \mathcal{Y}, i = 1, ..., n\}$, where y_i is a target value and $\mathbf{x}_i = \{x_1, ..., x_d\}$ is the list of corresponding explanatory variables. We can treat the GP as a collection of random functions $f(\cdot)$, such that, once drawn from its space, $f(\cdot)$ would imply a deterministic mapping from $\mathcal{X}^d$ to $\mathcal{Y}$. The random functions $f(\cdot)$ is also called *latent variables* in the context of GP.

Let $\mathbf{f} = \{f_1, ..., f_n\} = \{f(\mathbf{x}_1), ..., f(\mathbf{x}_n)\}$ be the latent variables corresponding to $\mathbf{y} = \{y_1, ..., y_n\}$, $m(\mathbf{x})$ be the mean function, and $K_\theta = k(\mathbf{x}, \mathbf{x}')$ be the GP kernel function describing the covariance. In this case, by the property of GPs that every finite subset of $f(\cdot)$ follows a joint normal distribution, we have that

$$\begin{bmatrix} f_1 \\ \vdots \\ f_n \end{bmatrix} \sim \mathbf{MVN} \left(\begin{bmatrix} m(\mathbf{x}_1) \\ \vdots \\ m(\mathbf{x}_n) \end{bmatrix}, \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \dots & k(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & \dots & k(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix} \right). \tag{2.1}$$

Now, consider the special case of regression. Suppose that $y_i = f(\mathbf{x}_i) + \varepsilon_i$ for $i = 1, ..., n$, where $\varepsilon_i \overset{iid}{\sim} \mathcal{N}(0, \sigma_n^2)$. Thus, target vales y only differ from latent variables f by additive noises ε, and $y_i | f_i \sim \mathcal{N}(f_i, \sigma_n^2)$. Let $\mathbf{x}^* \in \mathcal{X}^d$ denote a new data point, and $f^* = f(\mathbf{x}^*)$ be the corresponding latent variable. Based on the key assumption in GP modelling,

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \\ f^* \end{bmatrix} \sim \mathbf{MVN} \left(\begin{bmatrix} m(\mathbf{x}_1) \\ \vdots \\ m(\mathbf{x}_n) \\ m(\mathbf{x}^*) \end{bmatrix}, \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) + \sigma_n^2 & \dots & k(\mathbf{x}_1, \mathbf{x}_n) & k(\mathbf{x}_1, \mathbf{x}^*) \\ \vdots & \ddots & \vdots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & \dots & k(\mathbf{x}_n, \mathbf{x}_n) + \sigma_n^2 & k(\mathbf{x}_n, \mathbf{x}^*) \\ k(\mathbf{x}^*, \mathbf{x}_1) & \dots & k(\mathbf{x}^*, \mathbf{x}_n) & k(\mathbf{x}^*, \mathbf{x}^*) \end{bmatrix} \right).$$

By using the conditional property of MVN, the conditional distribution $p(f^* | \mathbf{x}^*, \mathcal{D}) = \mathcal{N}(\mu_0, \sigma_0^2)$ could then be computed. Note: the formulae used to calculate the mean

and variance are omitted here, as they are not essential for understanding GP models. The point prediction for y^* is the expected value and could directly be calculated through the iterated conditional expectation formula, $\hat{y}^* = \mathbb{E}_{f*}(\mathbb{E}(y^*|f^*)) = \mathbb{E}_{f*}(f^*) = \mu_0$. As a Bayesian approach, GP models also give us the information about the entire predictive distribution,

$$\begin{aligned} p(y^*|\mathbf{x}^*, \mathcal{D}) &= \int p(y^*|f^*)p(f^*|\mathbf{x}^*, \mathcal{D})df^* \\ &= \int \mathcal{N}(f^*, \sigma_n^2)\mathcal{N}(\mu_0, \sigma_0^2)df^* \\ &= \mathcal{N}(\mu_0, \sigma_0^2 + \sigma_n^2). \end{aligned} \tag{2.2}$$

Looking at the final expression of $p(y^*|\mathbf{x}^*, \mathcal{D})$, the variance has been decomposed into two distinct “sub-variances". The σ_0^2 tells us the noise-free spatial correlations, and another stands for the magnitude of white noise (Gramacy 2020).

2.2.4 GP Models with Non-Gaussian Likelihoods

We have just focused on the regressions with normally distributed noises, but how to use GP models to explore, for example, the survival probability or the number of flowering stalks? The basic idea for these GP models with non-Gaussian likelihoods is similar to that of the Generalized linear models (GLMs) — relating the latent variable with the response variable via a link function π. Then, after getting $p(f^*|\mathbf{x}^*, \mathcal{D})$, the predictions can be easily calculated by using the iterated conditional expectation formula, $\hat{y}^* = \mathbb{E}_{f*}(\mathbb{E}(y^*|f^*)) = \mathbb{E}_{f*}(\pi(f^*))$. However, as

$$\begin{aligned} p(f^*|\mathbf{x}^*, \mathcal{D}) &= \int p(f^*|\mathbf{x}^*, X, \mathbf{f})p(\mathbf{f}|\mathcal{D})d\mathbf{f} \\ &= \int p(f^*|\mathbf{x}^*, X, \mathbf{f})\frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|X)}{p(\mathbf{y}|X)}d\mathbf{f}, \end{aligned}$$

non-Gaussian likelihoods would make the integral too intractable to make predictions. In this case, we make approximate inferences instead. Let $p(f^*|\mathbf{x}^*, \mathcal{D}) \approx g(\cdot|\mathbf{x}^*, \mathcal{D})$, the prediction would then be estimated through $\hat{y}^* \approx \mathbb{E}_g(\pi(f^*))$. Popular approximation approaches include the Expectation propagation (Minka 2013) and Laplace approximation (C. K. Williams and Barber 1998), both of which approximate the non-Gaussian distribution with a Gaussian one (C. K. Williams and Rasmussen

2006). Solutions based on Markov chain Monte Carlo methods are also suggested by Bernardo et al. 1998.

2.2.5 GP hyperparameters

A Gaussian process is specified by $m(\mathbf{x})$ and $k(\mathbf{x}, \mathbf{x}')$. One of the assumption in the previous introduction of GP regressions is that the *mean* $m(\mathbf{x})$ and GP *kernel function* $k(\mathbf{x}, \mathbf{x}')$ are known. In practice, $m(\mathbf{x})$ and $k(\mathbf{x}, \mathbf{x}')$ are functions with hyperparameters and need to be specified by users. In the vast majority of cases, especially for the situations with absence of prior knowledge, $m(\mathbf{x})$ is simply set to as a flat zero-offset — whether the trend is upwards or downwards is equally unsure (S. Roberts et al. 2013). On the other hand, specifying $m(\mathbf{x})$ to other forms would only have noticeable effects on the predictions for regions far away from the training data (S. Roberts et al. 2013; Chen 2017), but this will definitely increase the computational complexity (as the number of hyperparameters increased). Thus, many people set $m(\mathbf{x}) = 0$ directly for simplicity (C. K. Williams and Rasmussen 2006).

The GP models gain their predictive power by measuring similarities between data points, while this crucial ingredient is defined by the kernel function k. Especially, when the mean function is assumed to be zero, the reliability of the model would be fully controlled by the GP kernel function. There are many potential kernel functions, and, in general, any symmetric and positive semi-definite function can serve as a GP kernel. Among kernel-based methods, the most commonly applied kernel is probably the *Gaussian Kernel*, also known as *the Squared Exponential* or *Radial Basis Function kernel*, and defined by

$$k_{Gau}(x, x') = r^2 \exp(-\frac{(x - x')^2}{l^2}), \tag{2.3}$$

where x and x' are two 1-dimensional inputs. This k_{Gau} takes two parameters.

- The scale factor r^2 controls the average distance of the results away from their mean. For example, looking at Figure 2.1 left, the 'amplitude' increases when raising r from 1 to 6. r^2 is speaking for the output variance.

- The lengthscale l^2 is used to adjust the strength of correlation between data points (as measured by the squared Euclidean distance). From Figure 2.1 right, a shorter lengthscale would bring us a “more wiggly" curve when keeping r^2 fixed. As a smaller length scale implies the data points far away impact less to the current point, then results a more rapidly varying function. A extreme large value in l therefore corresponds to a linear covariate.

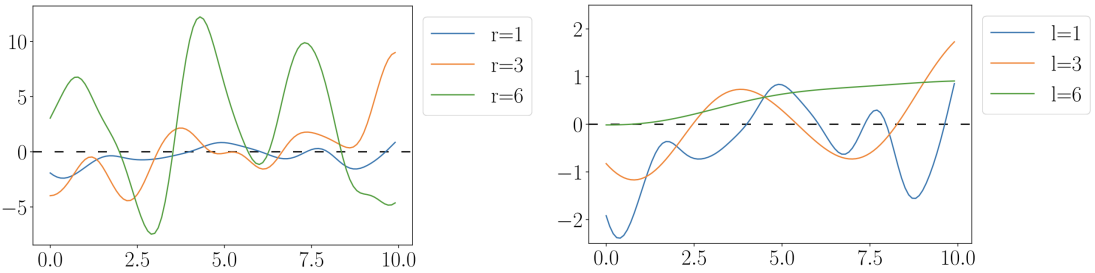


Figure 2.1: The plots display samples obtained from GP models with kernel function k_{Gau}. The left plot displays the samples obtained by varying the scale factor r from 1 to 6 while fixing the lengthscale l at 1. Conversely, the right panel illustrates the effect of increasing l from 1 to 6, while keeping r fixed at 1.

In many situations, the inputs are not 1-dimensional; for example, when the state variable z in an IPM describes both age and size. Using the current k_{Gau} with a single lengthscale can be problematic. For instance, a difference of 2 units in size should not be expected to have the same effect as a difference of 2 units in age. In such cases, automatic relevance determination (ARD) can be employed,

$$k_{Gau}^{ARD}(x, x') = r^2 \exp\left(-\sum_{\delta=1}^{\Delta} \frac{(x_\delta - x'_\delta)^2}{l_\delta^2}\right). \tag{2.4}$$

GP kernel functions with ARD assign separate lengthscales to each dimension, allowing the model to modulate the similarity magnitude independently for each input coordinate. In a GP regression with Δ-dimensional inputs and k_{Gau}^{ARD} kernel function, the hyperparameters are the noise variance σ_n^2, output variance r^2 and lengthscales l_δ for dimension $\delta = 1, ..., \Delta$. GP models estimate the values for these hyperparameters through maximising the marginal likelihood for observations, $p(\mathbf{y}|X) = \int p(\mathbf{y}|\mathbf{f}, X)p(\mathbf{f}|X)d\mathbf{f}$. As discussed previously, these estimations for GP

regressions are analytically straightforward, but hyperparameters in GP models with non-Gaussian likelihoods can only be computed through approximation methods.

A range of other GP kernel functions is discussed by C. K. Williams and Rasmussen 2006, Duvenaud 2014 and Gramacy 2020, like the *Matern* and *neural network* kernel. Apart from these kernel functions already constructed, users can also build GP kernels by themselves (as long as the kernels are symmetric and positive semi-definite). For example, the easiest way to build a new kernel would be combining two or more valid single kernel functions by adding them together. See Chapter 4 in C. K. Williams and Rasmussen 2006 for more details.

2.2.6 Experimental Study: Parameter perturbation in GP models

Understanding how small changes in GP hyperparameters affect model behaviour is important for quantifying model sensitivity — especially in the context of ABC, where particle distances depend on how models respond to parameter changes. While standard perturbation approaches are often used to explore such sensitivity, their interpretation can be unreliable in GP models due to interactions between hyperparameters. This section investigates these effects to inform the construction of more stable similarity measures in later ABC analyses.

GP models are highly flexible. When perturbing a single hyperparameter, the corresponding effect is highly dependent on the values of the other hyperparameters.

To explore this, we first use a toy dataset to perform Bayesian inference on a GP regression model predicting breeders' growth (G_f). Two representative samples are selected from the posterior, labelled as **M1** and **M2**. Both samples correspond to the same GP regression model and exhibit similar Negative Log-Likelihood (NLL) scores, indicating comparable performance in data fitting. Next, we implement standard perturbation to the hyperparameters in **M1** and **M2**, by adjusting one hyperparameter at a time while keeping the others fixed. The predicted mean and variance for the models, before and after the perturbation, are shown in Figures 2.2 and 2.3 respectively.

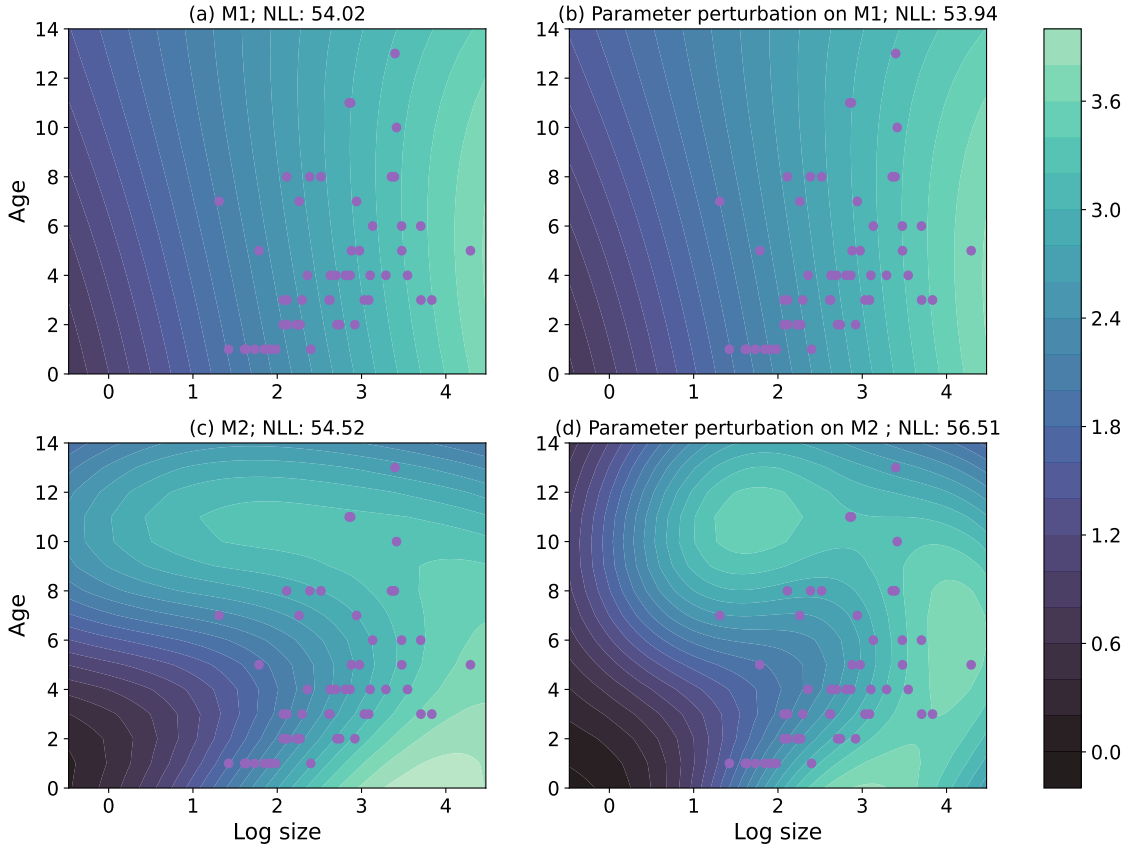


Figure 2.2: Given the age (y-axis) and logged size (x-axis) in the current year, the contour plots represent the predicted mean for logged size in the next year. Predictions were made through GP models with hyperparameters shown in Table A.1 in Appendix A. Plot (b) and (d) were produced by perturbing the lengthscale for the log size by the same **absolute** amount in (a) and (c) respectively. Purple dots represent for the values of covariates for the toy dataset.

When the same hyperparameter of **M1** and **M2** are perturbed with the same absolute amount, the contour plots of the predicted mean/variance for **M1** show no obvious difference before and after the perturbation (see plot (a) and (b) in Figure 2.2/2.3), whereas **M2** produces dramatic changes. Its predicted mean begin to show multiple peaks (plot (c) and (d) in Figure 2.2), and the predicted variance fall much more sharply as the logged size increased (plot (c) and (d) in Figure 2.3). When quantifying the effect of the perturbations, we observe that the NLL score for **M1** decreases from 54.02 by 0.08 to 53.94. However, for **M2**, the score changes from 54.52 to 56.51, with a difference of +1.99. Similar results can be also found in Figure A.1 and Figure A.2 in Appendix A, which, instead, considers

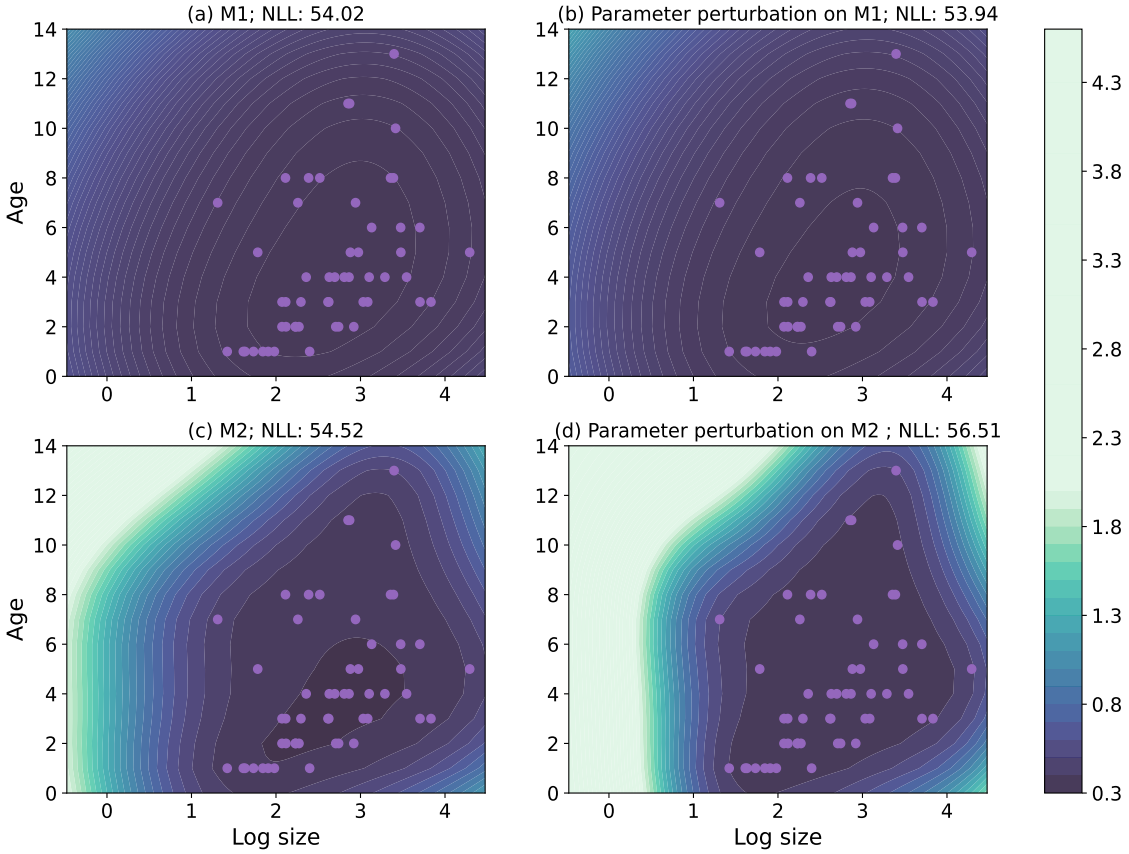


Figure 2.3: Contour plots showing the posterior predictive variance of the GP models, given the same hyperparameter settings used in Figure 2.2.

the response to the effects of proportional perturbations.

The impact of perturbations varies due to interactions among hyperparameters in GP kernels. These interactions become even more complex as the input dimension increases.

For example, in this exploration of **M1** and **M2**, we have the case of 2-dimensional inputs, e.g. $\boldsymbol{x_i} = (x_{1,i}, x_{2,i})^\top$. Based on Equation 2.4, we can introduce a separate lengthscale for each dimension. The kernel function k_{Gau} with this ARD setting can be expressed in two forms:

$$\begin{aligned} k_{Gau}(\boldsymbol{x_i}, \boldsymbol{x_j}) &= r^2 \exp(-\frac{(x_{1,i} - x_{1,j})^2}{2l_1^2}) \exp(-\frac{(x_{2,i} - x_{2,j})^2}{2l_2^2}) \\ &= \gamma(r, l_2, x_{2,i}, x_{2,j}) \exp(-\frac{(x_{1,i} - x_{1,j})^2}{2l_1^2}); \end{aligned} \tag{2.5}$$

or,

$$= \gamma(r, l_1, x_{1,i}, x_{1,j}) \exp(-\frac{(x_{2,i} - x_{2,j})^2}{2l_2^2}), \tag{2.6}$$

with $\gamma(r, l_\delta, x_{\delta,i}, x_{\delta,j}) = r^2 \exp(-\frac{(x_{\delta,i}-x_{\delta,j})^2}{2l_\delta^2})$ for $\delta = 1, 2$. Let us look at how the interpretation of lengthscales l_1 and l_2 can be differently from what we see with 1-dimensional inputs in Equation 2.3. Specifically, consider reducing l_1 while keeping all other parameters constant:

- **From the perspective of the 1*st* dimension (Equation 2.5):** The term $\gamma(r, l_2, x_{2,i}, x_{2,j})$ acts as an unaffected scale factor. In this case, reducing l_1, similar to what is observed in Figure 2.1 right, results in "more wiggly" curves. A smaller l_1 increases the influence of nearby points on the curve, making their impact more pronounced.

- **From the perspective of the 2*nd* dimension (Equation 2.6):** The term $\gamma(r, l_1, x_{1,i}, x_{1,j})$ becomes an affected scale factor. Reducing l_1 lowers this factor due to the decrease in $\exp(-\frac{(x_{1,i}-x_{1,j})^2}{2l_1^2})$. As shown in Figure 2.1 left, this reduction decreases the covariance between points, regardless of their distances.

- It is also important to note that these perturbations of lengthscales would not affect the variance of the points themselves (i.e. $k_{Gau}(\boldsymbol{x_i}, \boldsymbol{x_i})$), as scale factor r^2 does not really change.

For a visual representation of the explanations discussed above, please refer to the example provided in Appendix A.

The exploration in this section demonstrates the difficulties of conducting perturbation analysis in GP models. Specifically, changes to a single hyperparameter can have complex, non-local effects because of interactions with other hyperparameters. In this case, the standard perturbations, which are typically intended to cause minor adjustments, can sometimes lead to unexpectedly large differences, resulting in

models that behave very differently. This variability makes it challenge to generalise findings from the standard perturbation.

Such challenges are particularly relevant in the context of ABC, where the effectiveness of summary statistics depends on capturing model sensitivity to parameter changes. To address this, in Section 3.3, we propose using Negative Log Posterior Probability (NLPP) scores to redefine model similarities. These scores help by summarizing the overall effects of hyperparameter changes on the posterior distribution, reducing the influence of extreme cases and providing a more balanced view of model sensitivities.

2.3 Individual-based models

In this thesis, we generated all simulated population datasets using Individual-based models (IBM) (see reviews in Grimm et al. 2006; DeAngelis and Grimm 2014). IBM is a population simulation technique that tracks the life trajectory of each individual in the population while accounting for demographic stochasticity.

In our implementations, demographic events occur in accordance with the life cycle of the study species and are treated as random, with probabilities parameterized from GP/GLM-modelled vital rates. This aims to make the IBMs mimic real-life conditions, ensuring that the simulated demography is roughly comparable to the observed population (Ellner, Childs, et al. 2016). Furthermore, using IBM in our ABC framework allows us to generate population outcomes under specific hyperparameter values of IPMs, preparing them ready for integration into the ABC-PMC algorithm discussed in the next chapter.

3

Approximate Bayesian Computational methods

Contents

3.1 ABC algorithms 24
3.1.1 Non-iterative ABC methods 24
3.1.2 Iterative ABC methods 27
3.2 Summary statistics in ABC methods 31
3.3 A systematic workflow for selecting summary statistics 37
3.4 Modified ABC PMC 41
3.5 Detailed explanations for the ABC-PMC Modifications 47

ABC methods is a set of sampling methods for Bayesian inference, applied effectively across various ecological and biological studies (e.g. Jabot and Chave 2011; Scranton et al. 2014; Daly et al. 2017; J. Zhang et al. 2017; Ruiz-Suarez et al. 2020; Lux 2021). Traditional Bayesian inference requires explicit calculation of the likelihood function to provide an exact posterior distribution. However, in many biological studies, while it is feasible to simulate datasets, converting these simulations into an evaluable likelihood function is often difficult or impractical. ABC is particularly useful for implementing Bayesian analysis in these cases. It bypasses the need for an explicit likelihood function by using simulations to approximate the posterior distribution. The core process of ABC involves

four main steps:

1. **Simulate data:** Sample parameter values from a prior distribution, $\pi(\boldsymbol{\theta})$, and simulate datasets.

2. **Compare to observed data:** Compare each simulated dataset to the observed dataset. Typically, this comparison employs a set of *summary statistics* instead of the full dataset to reduce dimensionality and computational demands.

3. **Accept or Reject:** Accept parameters if summary statistics from simulated data, $\boldsymbol{s}^*$, closely match those from observed data, $\boldsymbol{s}^{obs}$, based on a set of predefined threshold. Otherwise, reject.

4. **Posterior approximation:** Treat the accepted parameter values as samples from an approximation of the posterior distribution, $\pi(\boldsymbol{\theta}|\boldsymbol{s}^{obs}) \approx \pi_{ABC}(\boldsymbol{\theta}|\boldsymbol{s}^{obs})$.

ABC methods can be broadly categorized into non-iterative and iterative approaches. In this chapter, we begin by exploring non-iterative methods (Section 3.1.1) and gradually building our understanding through the concept of the ABC *acceptance kernel* $\mathcal{K}$, leading us to iterative approaches (Section 3.1.2). Section 3.2 explores the role of summary statistics in ABC methods, explaining their importance in capturing key information from datasets. We also discuss the difficulties currently faced and use a toy example to illustrate these issues, which provides insights to develop our version of ABC sampler. Recognising the challenge of selecting appropriate summary statistics in ABC applications with GP models, we propose a systematic workflow in Section 3.3, as a guideline to also help others to navigate this complexity. In Section 3.4, we introduce a developed version of ABC-PMC method to alleviate common challenges such as high rejection rates and sampling inefficiency in combined models, like IPMs. A comprehensive explanation of the modifications for this developed ABC-PMC method is presented in Section 3.5.

3.1 ABC algorithms

3.1.1 Non-iterative ABC methods

Assume that Bayesian inference is considered for the parameter vector $\boldsymbol{\theta} \in \Theta$ under a model with the likelihood $f(y_{obs}|\boldsymbol{\theta})$ for the observed dataset $y_{obs} \in \mathcal{D}$. Let $\pi(\boldsymbol{\theta})$ denote the prior density, then the inference in this case relies on the target posterior density $\pi(\boldsymbol{\theta}|y_{obs}) \propto f(y_{obs}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})$. We further assume that evaluating the likelihood $f(y_{obs}|\boldsymbol{\theta})$ is computationally expensive or numerically impossible, but generating simulations from it is straightforward. When $\mathcal{D}$ is a countable or finite set, an intuitive non-iterative ABC method (Algorithm 1) was introduced by Tavaré et al. 1997.

Algorithm 1: ABC Rejection sampling method

Input : The target posterior density $\pi(\boldsymbol{\theta}|y_{obs}) \propto f(y_{obs}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})$; an integer $N > 0$.

Output : A set of accepted parameter vectors $\boldsymbol{\theta_1}, \boldsymbol{\theta_2}, ..., \boldsymbol{\theta_N} \sim \pi_{ABC}(\cdot|y_{obs})$.

Sampling :

```
for n = 1, ..., N do
    repeat
        Draw independently θ* ~ π(θ).
        Simulate independently y* ~ f(y|θ*).
    until y* = y_obs;
    Set θ_n = θ*.
end
```

As an extension the standard Rejection sampling algorithm, Algorithm 1 accepts the proposed parameter $\boldsymbol{\theta}^*$ if and only if it could generate an identical dataset that $y^* = y_{obs}$. While this approach theoretically provides exact posterior results (i.e., $\pi_{ABC}(\boldsymbol{\theta}|y_{obs}) = \pi(\boldsymbol{\theta}|y_{obs})$), the requirement for identical datasets makes acceptance highly unlikely, especially in complex scenarios. This can lead to sampling inefficiency due to the computational overhead of handling large mismatches. Additionally, if the data is continuous, the acceptance probability will be exactly zero — a situation more commonly encountered in real-world scenarios (S. A. Sisson, Fan, and Beaumont 2018).

In this case, a typical practical implementation of ABC analysis replaces the full dataset y_{obs} with a low-dimensional vector of summary statistics $\boldsymbol{s}^{obs} = S(y_{obs}) \in \mathbb{R}^{\mathscr{d}_s}$. To circumvent potential zero acceptance probability, $\boldsymbol{\theta}^*$ would be accepted as long as the observed and simulated summary statistics, $\boldsymbol{s}^{obs}$ and $\boldsymbol{s}^*$, are close enough measured by certain distance metrics, such as the Euclidean distance (Fu and W.-H. Li 1997; Wall 2000; Tishkoff et al. 2001). When applying these modifications to Algorithm 1, line 5 is replaced by $d_i({}^i s^*, {}^i s^{obs}) < h_i$ for i in 1, ..., $\mathscr{d}_s$, where d_i and h_i denote the distance metric and acceptance criterion for the ith summary statistic respectively. (Note: for the sake of simplicity, we assume that the dimension for the summary statistics $\mathscr{d}_s = 1$ for now, but we will continue to use boldface to remind readers that the summary statistics can be a multidimensional vector).

As a result, the outcomes from the updated Algorithm 1 are now distributed to $\pi_{ABC}(\boldsymbol{\theta}|\boldsymbol{s}^{obs})$, where $\lim_{\boldsymbol{h}\to 0} \pi_{ABC}(\boldsymbol{\theta}|\boldsymbol{s}^{obs}) = \lim_{\boldsymbol{h}\to 0} \pi_{ABC}(\boldsymbol{\theta}|d(\boldsymbol{s}^*, \boldsymbol{s}^{obs}) < \boldsymbol{h}) = \pi(\boldsymbol{\theta}|\boldsymbol{s}^{obs})$. Furthermore, if the summary statistic $\boldsymbol{s}$ is *(Bayes) sufficient* (Cox and Hinkley 1979) for the parameter $\boldsymbol{\theta}$, then $\pi(\boldsymbol{\theta}|\boldsymbol{s}^{obs}) \equiv \pi(\boldsymbol{\theta}|y^{obs})$. In other words, the accepted $\boldsymbol{\theta}^*$ values from the updated Algorithm 1 are samples drawn from the true correct posterior density. We will revisit this and provide a more detailed explanation of summary statistics in ABC methods in Section 3.2.

Generally speaking, finding a sufficient summary statistic for y_{obs} in real practice is exceedingly difficult. Consequently, ABC samplers in practice typically settle for the next best thing, assuming that $\pi(\boldsymbol{\theta}|y^{obs})$ can be approximated by $\pi(\boldsymbol{\theta}|\boldsymbol{s}^{obs})$ when using insufficient summary statistics. Given this assumption, our objective is instead to approximate the distribution $\pi(\boldsymbol{\theta}|\boldsymbol{s}^{obs})$. We do this by initially sampling from the joint distribution of $\boldsymbol{\theta}$ and $\boldsymbol{s}^*$, $\pi(\boldsymbol{\theta}, \boldsymbol{s}^*|\boldsymbol{s}^{obs})$, and subsequently discarding the marginal $\boldsymbol{s}^*$, that is, $\pi_{ABC}(\boldsymbol{\theta}|\boldsymbol{s}^{obs}) = \int \pi_{ABC}(\boldsymbol{\theta}, \boldsymbol{s}^*|\boldsymbol{s}^{obs}) d\boldsymbol{s}^*$ (S. Sisson and Fan 2018). Especially, for non-iterative ABC samplers, the joint distribution $\pi(\boldsymbol{\theta}, \boldsymbol{s}^*|\boldsymbol{s}^{obs})$ is approximated by

$$\pi_{ABC}(\boldsymbol{\theta}, \boldsymbol{s}^*|\boldsymbol{s}^{obs}) \propto \mathcal{K}(\boldsymbol{d}(\boldsymbol{s}^{obs}, \boldsymbol{s}^*)) p(\boldsymbol{s}^*|\boldsymbol{\theta}) \pi(\boldsymbol{\theta}), \tag{3.1}$$

where $\mathcal{K}$ stands for the acceptance probability of the proposed value $\boldsymbol{\theta}$.

Various ABC sampling strategies were suggested by modifying the form of $\mathcal{K}$. For instance, $\mathcal{K}$ can be a smoothing kernel function, K_h, describing about the closeness (in some manner) between the simulated and observed dataset. That is, $\mathcal{K}(\cdot) = K_h(\cdot) = \frac{1}{h}K(\frac{\cdot}{h})$, with $h > 0$ corresponding to the bandwidth or the scale parameter of the smoothing kernel. For example, in the case of ABC rejection sampler, an uniform kernel is employed, $K_h(\cdot) = K_h(d(\boldsymbol{s}^{obs}, \boldsymbol{s}^*)) = \frac{1}{h}\mathbb{1}_{d(\boldsymbol{s}^*, \boldsymbol{s}^{obs})<h}$, where $\mathbb{1}_c$ is an indicator function about condition c. When $K_h(x)$ is replaced by some other more generic smoothing kernels, such as Gaussian and Epanechnikov kernel, this standard ABC rejection sampler can be extended to be able to distinguish between those cases where, for instance, the simulated summary statistic is exactly identical to the observed, $\boldsymbol{d}(\boldsymbol{s}^{obs}, \boldsymbol{s}^*) = 0$, and where $\boldsymbol{s}^*$ is farthest away from $\boldsymbol{s}^{obs}$ within the acceptance region, i.e. $\boldsymbol{d}(\boldsymbol{s}^{obs}, \boldsymbol{s}^*) = h$ (S. A. Sisson, Fan, and Beaumont 2018). For application examples and a range of forms for the common smoothing kernel employed in ABC scenario, please check Beaumont, W. Zhang, et al. 2002 and S. Sisson and Fan 2018.

If we go one step further and consider a more generic form of $\mathcal{K}$, by setting $\mathcal{K}(\boldsymbol{d}(\boldsymbol{s}^{obs}, \boldsymbol{s}^*)) = K_h(\boldsymbol{d}(\boldsymbol{s}^{obs}, \boldsymbol{s}^*))\frac{q(\boldsymbol{\theta})}{q(\boldsymbol{\theta})}$ with some proposal distribution $q(\boldsymbol{\theta})$ such that $q(\boldsymbol{\theta}) > 0$ if $\pi(\boldsymbol{\theta}|\boldsymbol{s}^{obs}) > 0$, we then obtain an non-iterative ABC importance sampler. Accordingly, Approximation 3.1 becomes to

$$\begin{aligned}\pi_{ABC}(\boldsymbol{\theta}, \boldsymbol{s}^*|\boldsymbol{s}^{obs}) &\propto K_h(\boldsymbol{d}(\boldsymbol{s}^{obs}, \boldsymbol{s}^*))\frac{\pi(\boldsymbol{\theta})}{q(\boldsymbol{\theta})}p(\boldsymbol{s}^*|\boldsymbol{\theta})q(\boldsymbol{\theta}) \\ &\propto K_h(\boldsymbol{d}(\boldsymbol{s}^{obs}, \boldsymbol{s}^*))w(\boldsymbol{\theta})p(\boldsymbol{s}^*|\boldsymbol{\theta})q(\boldsymbol{\theta}). \qquad (3.2)\end{aligned}$$

We can still interpret this as a standard ABC rejection sampler, but now it comes with an additional correction scheme based on importance weights $w(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta})/q(\boldsymbol{\theta})$.

In real life, the posterior distribution is often quite different from the prior distribution, $\pi(\boldsymbol{\theta})$, or the proposal distribution, $q(\boldsymbol{\theta})$. Under such circumstances, most proposed values are likely to fall within regions of low posterior probability (Marin et al. 2012), which results in a high rejection rate and makes the ABC sampler we discussed highly inefficient. Naturally, one might wonder: once we have

accepted parameter values (from rejection or importance sampling), can these values guide us towards better subsequent proposals that are more likely to be accepted?

ABC Markov Chain Monte Carlo (ABC-MCMC), introduced by Marjoram, Molitor, et al. 2003, seeks to enhance sampling efficiency in this manner. It does so by formulating better-informed proposal distributions, drawing from previously accepted values. Specifically, ABC-MCMCs often employ 'local' proposals that suggest new candidates situated just 'around' the current accepted values. We could expect that parameter values producing simulations close to the observed data are visited and accepted more often in ABC-MCMC, while this increased acceptance rate comes at the cost of potentially dependent simulation outcomes (Marjoram, Molitor, et al. 2003). Note that using the results of rejection or importance samplers as the initialisation of MCMC samplers could be unnecessary, since the outcome of ABC-MCMC samplers would be a Markov chain with the equilibrium distribution. So, in theory, outcomes would forget where they started when they reached the equilibrium state, although this is a difficult condition to verify (Marjoram and Tavaré 2006; Csilléry, Blum, et al. 2010). See Marjoram, Molitor, et al. 2003; Wegmann et al. 2009; Del Moral et al. 2012; Lenormand et al. 2013; Andrieu, Lee, et al. 2018 for more examples and discussion about the ABC MCMC.

3.1.2 Iterative ABC methods

Utilising the accepted parameter values, also known as the accepted *particles*, to build more efficient proposals is not exclusive for ABC-MCMC. Iterative ABC samplers are also designed to exploit accepted particles for the purpose of sampling in high probability regions of the final target posterior (S. Sisson and Fan 2018). In these iterative algorithms, a simulated population of N accepted particles is generated in each iteration. Then, instead of just utilising a single accepted particle as in ABC-MCMC, a population of weighted accepted particles produced in the previous iteration is re-used to construct a more efficient proposal distribution for the subsequent iteration (Beaumont, Cornuet, et al. 2009; Toni et al. 2009). Therefore, the proposal distribution used in the final sampling step in these iterative

methods is corrected step by step via a sequence of slowly varying intermediate distributions about those simulated populations.

Based on just a single particle, the output from ABC-MCMC might be potentially unreliable. Especially, when the correlated nature in outcomes meets the potentially low acceptance rate, the chain could get stuck in the low probability region for a really long time (Toni et al. 2009). This challenge is greatly relieved in the iterative ABC samplers when the population of accepted particles generated in each iteration is under a sufficient large size.

Algorithm 2 is an example of iterative ABC samplers which was developed based on population Monte Carlo (PMC) (Cappé et al. 2004) and is called ABC-PMC (Beaumont, Cornuet, et al. 2009; Toni et al. 2009; S. A. Sisson, Fan, and Tanaka 2009). For simplicity, uniform smoothing kernel functions were employed in Algorithm 2 to accept all eligible particles, but it can theoretically be replaced by other more general smoothing kernels, see, for example, Peters et al. 2012.

When implementing ABC-PMC only once, $C = 0$, it is nothing more than a simple ABC importance sampler. When $C > 0$, ABC-PMC is in essence an iterative version of the ABC importance sampler (Cappé et al. 2004; Lintusaari et al. 2017). That is (to align with Equation 3.2), for each iteration, $c = 0, ..., C$, ABC-PMC draw samples from the target

$$\begin{aligned}\pi^{c}_{ABC}(\boldsymbol{\theta}, \boldsymbol{s}^{*}|\boldsymbol{s}^{obs}) &\propto K_{h_c}(\boldsymbol{d}(\boldsymbol{s}^{obs}, \boldsymbol{s}^{*}))\frac{\pi(\boldsymbol{\theta})}{q_c(\boldsymbol{\theta})}p(\boldsymbol{s}^{*}|\boldsymbol{\theta})q_c(\boldsymbol{\theta}) \\ &\propto \mathbb{1}_{\boldsymbol{d}(\boldsymbol{s}^{obs},\boldsymbol{s}^{*})<h_c} w_c(\boldsymbol{\theta})p(\boldsymbol{s}^{*}|\boldsymbol{\theta})q_c(\boldsymbol{\theta}), \end{aligned} \tag{3.3}$$

where bandwidth $\{h_c\}_{c=0,...,C}$ should be a decreasing sequence $h_0 > \cdots > h_C \geqslant 0$, so that each successive intermediate targets can be less diffused and gradually converge to the final target distribution, i.e., $\lim_{h_C \to 0} \pi^{C}_{ABC}(\boldsymbol{\theta}, \boldsymbol{s}^{*}|\boldsymbol{s}^{obs}) = \lim_{h_C \to 0} \pi_{ABC}(\boldsymbol{\theta}, \boldsymbol{s}^{*}|\boldsymbol{d}(\boldsymbol{s}^{obs}, \boldsymbol{s}^{*}) < h_C) = \pi(\boldsymbol{\theta}, \boldsymbol{s}^{*}|\boldsymbol{s}^{obs})$ (Toni et al. 2009; S. Sisson and Fan 2018). In ABC-PMC, the actual sampling tasks are all handled by the ABC importance sampler operated in the very last iteration. However, the performance of this final sampling procedure depends on the effort made in all previous iterations, through the improved proposal $q_C(\boldsymbol{\theta})$. In other words, when $c > 0$, the currently

Algorithm 2: Conventional ABC-PMC

Input : Simulation length C; the number of accepted particles N; a decreasing sequence of scale parameters $h_0 > \cdots > h_C \geqslant 0$.

Output : A set of accepted particles $\{\boldsymbol{\theta}_C^i\}_{1\leqslant i\leqslant N}$ with the corresponding normalised weights $\{w_C^i\}_{1\leqslant i\leqslant N}$ drawn from $\pi_{ABC}(\boldsymbol{\theta}|\boldsymbol{s}^{obs}) \propto \int \mathbb{1}_{\boldsymbol{d}(\boldsymbol{s}^{obs},\boldsymbol{s}^*)<h_C} w_C(\boldsymbol{\theta}) p(\boldsymbol{s}^*|\boldsymbol{\theta}) q_C(\boldsymbol{\theta}) d\boldsymbol{s}^*$.

Sampling :

for c = *0, ..., C* **do**
  $i \leftarrow 1$ **while** $i \leqslant N$ **do**
    Draw independently $\boldsymbol{\theta}^* \sim q_c(\boldsymbol{\theta})$, where

$$q_c(\boldsymbol{\theta}) = \begin{cases} \pi(\boldsymbol{\theta}) & \text{if } c = 0; \\ \sum_{j=1}^N w_{c-1}^j \mathcal{P}(\boldsymbol{\theta}|\boldsymbol{\theta}_{c-1}^j) & \text{otherwise,} \end{cases}$$

    with the perturbation kernel $\mathcal{P}(\cdot|\boldsymbol{\theta}_{c-1}^j)$ and normalised weights $\{w_{c-1}^j\}_{1\leqslant j\leqslant N}$.
    if $\pi(\boldsymbol{\theta}^*) = 0$ **then**
      Reject $\boldsymbol{\theta}^*$;
      $i \leftarrow i$
    else
      Simulate independently $y^* \sim f(y|\boldsymbol{\theta}^*)$ and compute summary statistics $\boldsymbol{s}^*$.
      if $\boldsymbol{d}(\boldsymbol{s}^{obs}, \boldsymbol{s}^*) \leqslant h_c$ **then**
        Accept and set $\boldsymbol{\theta}_c^i = \boldsymbol{\theta}^*$;
        Compute the corresponding weight $\omega_c^i = \pi(\boldsymbol{\theta}_c^i)/q_c(\boldsymbol{\theta}_c^i)$;
        $i \leftarrow i + 1$
      else
        Reject $\boldsymbol{\theta}^*$;
        $i \leftarrow i$
      end
    end
  end
  Normalise weights to have $\{w_c^i\}_{1\leqslant i\leqslant N}$;
  $c \leftarrow c + 1$
end

adopted proposal distribution $q_c(\boldsymbol{\theta})$ is dependent on the population of accepted particles in the previous iteration $c-1$. $q_c(\boldsymbol{\theta})$ is constructed by discarding the poor particles and only recycling the good ones to be able to adapt to a narrower acceptance region of the current intermediate target. This viewpoint makes ABC-PMC appear to be a naive but valid adaptive ABC sampler.

Algorithm 2 leaves the choice of bandwidth $\{h_c\}_{c=0,\ldots,C}$ up to the users themselves, but it could be difficult to specify or tune it well. An approach introduced by Drovandi and Pettitt 2011 can select $\{h_c\}_{c=1,\ldots,C}$ adaptively. The method sets $\{h_c\}_{c=0,\ldots,C}$ to be the α quantile of the accepted distances at the previous iteration and simply assigns $h_0 = \infty$ to accept all particles at the very beginning. If we apply this approach in Algorithm 2, we need to store the distance $d^i = \boldsymbol{d}(\boldsymbol{s}^{obs}, \boldsymbol{s}^*)$ just after Step 10; and then set h_{c+1} to α quantile of $\{d^i\}_{1 \leqslant i \leqslant N}$ after Step 19. Note that when using this method, the notation of the bandwidth changes from h_{c+1} to δ_c to highlight that the threshold used in the $(c+1)th$ iteration is actually based on the estimation from the cth iteration.

Due to its adaptability, the approach allows users to implement ABC-PMC in a very straightforward way, which makes it perhaps the most popular method of choosing the thresholds $\{h_c\}_{c=0,\ldots,C}$ at present. Nevertheless, it may also introduce other potential risks in our study, which will be discussed in more details in Section 3.5. Many other alternatives also exist for setting $\{h_c\}_{c=0,\ldots,C}$ in an adaptive manner, for example, Del Moral et al. 2012; Numminen et al. 2013; Silk et al. 2013; Prangle 2017; Simola et al. 2021.

Additionally, it is worth noting that ABC-PMC is sometimes also called 'ABC-SMC' (Sequential Monte Carlo) in certain literature, such as (Toni et al. 2009; Filippi et al. 2013; Scranton et al. 2014; Bonassi and West 2015). However, 'ABC-SMC' is usually used to refer to another iterative ABC sampler proposed by Del Moral et al. 2012. A major difference is that, in ABC-SMC by Del Moral et al. 2012, there is a resampling step before the proposed move (Step 3 in Algorithm 2). In this case, a Markov transition kernel, such as the Metropolis–Hastings kernel, must be applied in ABC-SMC to keep the new samples within the MCMC realm. Readers should not be confused with one another. We stick to call the approach described in Algorithm 2 as ABC-PMC throughout this paper, for a detailed explanation of ABC-SMC, please see Del Moral et al. 2012; Marin et al. 2012; Lenormand et al. 2013; S. Sisson and Fan 2018.

3.2 Summary statistics in ABC methods

Together with the choice of the sampling method, the selection of summary statistics has long been recognised as critical to the quality of ABC results. They all have a direct impact on the quality of the ABC approximation (Prangle 2017; S. A. Sisson, Fan, and Beaumont 2018). ABC methods approximate the posterior distribution based on a set of assumptions, which include ensuring that the selected summary statistics $\boldsymbol{s}$ are representative (enough), the tolerance h is small (enough), then a good (enough) approximation of the posterior can be generated (Marin et al. 2012).

Among the assumptions, the most tricky one is probably that we can expect that inferences about $\boldsymbol{\theta}$ should involve the data only through the summary statistics $\boldsymbol{s}$. This assumption can be relaxed when the summary statistics $\boldsymbol{s}$ are (joint) sufficient for $\boldsymbol{\theta}$. In this special case, there is no longer an 'approximation', while the outputs are exactly produced from the true posterior density, i.e. instead of $\pi(\boldsymbol{\theta}|y) \approx \pi(\boldsymbol{\theta}|\boldsymbol{s})$, we have

$$\begin{aligned} \pi(\boldsymbol{\theta}|y) &= \pi(\boldsymbol{\theta}|y, \boldsymbol{s}) && \leftarrow \pi(\boldsymbol{\theta}|\boldsymbol{s}, y) \text{ is independent of } y \\ &= \pi(\boldsymbol{\theta}|\boldsymbol{s}). && \text{if } \boldsymbol{s} \text{ is sufficient.} \end{aligned}$$

However, in a situation where the likelihood function is even unknown beyond a data generation procedure (otherwise, there would be no need for implementing ABC methods), finding low-dimensional sufficient statistics is a big challenge and could even be (nearly) impossible (Marin et al. 2012; Blum et al. 2013; S. A. Sisson, Fan, and Beaumont 2018). Not to mention, we are still not sure if such low-dimensional sufficient statistics exist in the context of ABC.

According to the Pitman-Koopman-Darmois theorem, for independent identically distributed data points from a model *not* belonging to the exponential family, the dimension of any sufficient statistic remains unbounded and keeps increasing with the sample size. Given that ABC methods are generally not required for models from the exponential family (since their likelihoods are usually not intractable), the theorem strongly suggests that the low-dimensional sufficient statistics we are

looking for are not likely to exist (Prangle 2018). As using sufficient statistics is often not very likely in real-life situations, the choice of summary statistics play an even more important role in this case. It is directly responsible for the quality of the posterior approximation in ABC scenarios, based on the assumption $\pi(\boldsymbol{\theta}|y) \approx \pi(\boldsymbol{\theta}|\boldsymbol{s})$. Consequently, methods for systematically selecting low-dimensional insufficient summaries are necessarily required.

In the absence of sufficient statistics in practice, it is intuitive to increase the number of summary statistics employed in the ABC samplers (to increase the amount of information available and thereby reduce the inference bias due to the irrevocable loss of information). More formally, relying on the rough idea of sufficiency, we may continue to add summary statistics and run ABC until the resulting posterior approximation stabilises (which is called Approximate sufficiency in Joyce and Marjoram 2008). However, it is not always feasible to keep adding more and more summary statistics to the ABC samplers. The most apparent drawback of this approach is that it may suffer from the curse of dimensionality. The main issue is that, even with the correct model and the true parameter values, high-dimensional simulations rarely match fixed observations well due to numerous random components that must match the target simultaneously (Prangle et al. 2018). This is why selecting appropriate summary statistics still remains one of the most challenging aspects of implementing ABC in practice. It is not an easy task to balance the need for a large number of summary statistics — to capture most of the information about the observed dataset — with the desire to keep the dimensionality low — to minimise random discrepancies between the summary statistics of the observed and simulated data.

Another potential risk of keeping adding more summary statistics with brute force is that it may introduce additional noise, leading to biased inference results (Beaumont, W. Zhang, et al. 2002; Csilléry, Blum, et al. 2010; S. A. Sisson, Fan, and Beaumont 2018). The toy example considered in S. A. Sisson, Fan, and Beaumont 2018 illustrated the issue.

A toy example:

The observed data $\{y_i\}_{1\leqslant i\leqslant n}$ were assumed to follow a Poisson distribution, $y_i \sim Poi(\lambda)$, with parameter $\lambda \sim \Gamma(\alpha, \beta)$. This setting results in a closed-form posterior distribution for λ, $\lambda|y_1, ..., y_n \sim \Gamma(\alpha + n\bar{y}, \beta + n)$, where $\bar{y}$ is the sample mean. Although the sample mean is a sufficient statistic for λ, S. A. Sisson, Fan, and Beaumont 2018 also considered using the sample standard deviation v as an additional (insufficient) summary statistic to estimate λ using ABC, given that the mean and variance of a Poisson distribution are both equal to λ. Given the parameters $\alpha = \beta = 1$ and observations $\{y_i\}_{1\leqslant i\leqslant 5} = (0, 0, 0, 0, 5)$, the ABC approximations are shown in Figure 3.1. Notice that, this setting is, in general, not going to happen in the real practice with ABC. There is no need to require ABC when likelihood is tractable.

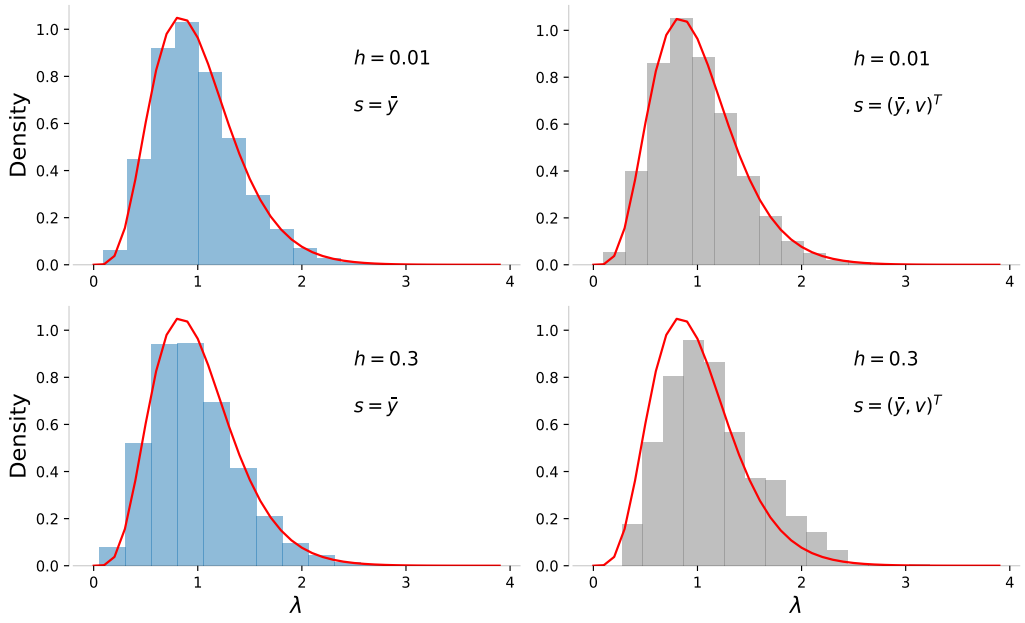


Figure 3.1: Density histograms of ABC posterior approximations. The approximations were generated through ABC Rejection samplers with the Euclidean distance metric and two different bandwidth values: $h = 0.01$ and $h = 0.3$. The ABC acceptance kernel $\mathcal{K}$ (in Equation 3.1) was defined as $\mathcal{K}(\boldsymbol{d}(\boldsymbol{s}^{obs}, \boldsymbol{s}^*)) = \mathbb{1}_{\boldsymbol{d}(\boldsymbol{s}^{obs},\boldsymbol{s}^*)<h} = \mathbb{1}_{|\bar{y}^{obs}-\bar{y}^*|<h}$ for plots on the left column, and $\mathcal{K}(\boldsymbol{d}(\boldsymbol{s}^{obs}, \boldsymbol{s}^*)) = \mathbb{1}_{|\bar{y}^{obs}-\bar{y}^*|<h}\mathbb{1}_{|v^{obs}-v^*|<h}$ for plots on the right column. The red curves indicate the true posterior distribution $\Gamma(6, 6)$.

When the sufficient statistic $\bar{y}$ was solely employed for the ABC, the approximation near-perfectly matched the true posterior under a very narrow bandwidth

value of $h = 0.01$. However, when the bandwidth got slightly wider $h = 0.3$, a minor fat-tailed pattern emerged in the approximation, deviating slightly from the true posterior distribution $\Gamma(6, 6)$. The difference, while minor, emphasised the fact that the set of λ capable of reproducing $\bar{y}^{obs}$ within a non-zero tolerance is (possibly similar, but still) different from the set that can exactly reproduce $\bar{y}^{obs}$. On the other hand, when the ABC approximation was jointly determined by sample mean and sample standard deviation, a strong restriction of $h = 0.01$ made the effect of the new summary statistics v almost negligible, leading to a similar approximation as using the sample mean only. However, when the ABC algorithm allowed for a more generous matching with $h = 0.3$, insufficient statistics v had additional opportunities to contribute more to the ABC approximation. The extra information provided by v^{obs} acted as 'noise', pushing the resulting posterior approximation away from the true posterior and producing a biased inference result. Thus, we should not expect that ABC methods based on the idea of approximate sufficiency would provide us reliable approximations (S. A. Sisson, Fan, and Beaumont 2018), or at least not always.

Let us delve deeper into this example to see if we can gain any additional insights into ABC applications. The sample data $\{y_i\}_{1 \leqslant i \leqslant 5} = (0, 0, 0, 0, 5)$ yields a sample mean of $\bar{y}^{obs} = 1$ and a sample standard deviation of $v^{obs} = \sqrt{5}$. $(v^{obs})^2 \gg \bar{y}^{obs}$ indicates an over-dispersed pattern, which is inconsistent with the common observations generated through a Poisson model. When both the sample mean and sample standard deviation are employed in ABC, increasing the threshold h results in more simulated parameters λ^* stand for larger values due to the term $\mathbb{1}_{|v^{obs} - v^*| < h}$ in $\mathcal{K}(\boldsymbol{d}(\boldsymbol{s}^{obs}, \boldsymbol{s}^*))$, meanwhile, there are still quite a part speak for small values due to $\mathbb{1}_{|\bar{y}^{obs} - \bar{y}^*| < h}$. Then, it is not surprising to obtain a such ABC approximation that was skewed towards larger values when $h = 0.01$ was increased to $h = 0.3$ (the histograms on the right column in Figure 3.1).

We now gain a better understanding of the 'noise' generated by the summary statistic v. In this example, given the true model and the corresponding parameter, a set of observations containing information that was 'inconsistent' with the true

model was generated. At that moment, an insufficient summary statistics bringing that piece of inconsistent information was employed into ABC methods, which, therefore, produced inference results that were not consistent with the actual posterior, namely the biased results. Now, the question is whether we can devise a strategy to reduce the impact of such summary statistics containing inconsistent information, and then obtain less biased inference results?

However, unlike in this given example, identifying which summary statistics are likely to bringing inconsistent information is already a big challenge when addressing real-life problems, let alone trying to reduce their impact. The task requires the knowledge of the true model behind, which might never be obtainable in the common practice. In this case, all we can do is, as with other approaches doing statistical inference, assume that the model is correct and make inference only about the parameter values. Under this assumption, summary statistics which are more intend to be unusual can be detected by simulating datasets from the given 'true' model. Then, relatively small weights could be assigned to these statistics to mitigate the impact of inconsistent information in the resulting ABC approximations.

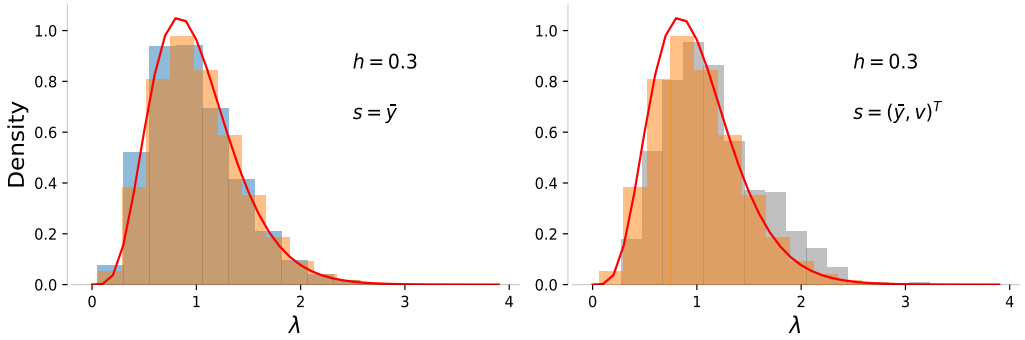


Figure 3.2: Density histograms of ABC posterior approximations. These two histograms are identical to the plots on the bottom of Figure 3.1. The additional orange shadow areas are the ABC posterior approximations based on the weighting strategy, where the acceptance kernel is defined as $\mathcal{K}(\boldsymbol{d}(\boldsymbol{s}^{obs}, \boldsymbol{s}^*)) = \mathbb{1}_{(((\bar{y}^{obs}-\bar{y}^*)*w_1)^2+((v^{obs}-v^*)*w_2)^2)^{1/2}<h}$ with $h = ((0.3 * w_1)^2 + (0.3 * w_2)^2)^{1/2}$ to make the histograms comparable (from some sense).

Figure 3.2 displays the posterior approximation generated through an ABC rejection sampler that utilises a Euclidean metric-based weighting strategy, which has been commonly employed in previous studies, such as Hamilton et al. 2005; Luciani et al. 2009. The weights $w_1 = 1/0.8$ and $w_2 \approx 1/1.604$ were defined as the

inverse of median absolute deviations for $(\bar{y}^{obs} - \bar{y}^*)$ and $(v^{obs} - v^*)$ respectively (Csilléry, François, et al. 2012). By giving a higher weight to $\bar{y}$, the new ABC posterior approximation was pulled back from the higher values (Figure 3.2 right). Although the impact of v was not completely eliminated (Figure 3.2 left), we did not observe a clear difference between the estimated and actual mode. The approach employed here automatically evaluates the stability or stiffness of two different types of the "difference" (i.e. the summary statistics) between the simulated and observed data and assigns weights accordingly.

This weighting strategy could not be useful for non-informative summary statistics, as they may be 'steadily' far from the observed values, but based on the assumption that all summary statistics employed are sensitive to the parameters, this strategy makes intuitive sense. On the other hand, by viewing the weights as the inverse of some factors, this approach can be also considered as a normalisation method. In this case, the method uses normalisation factors to bring all the individual 'differences' to the same scale, i.e. $\frac{(\bar{y}^{obs}-\bar{y}^*)}{(1/w_1)}$ and $\frac{(v^{obs}-v^*)}{(1/w_2)}$ in this example, and to make them comparable for later use. We will discuss the weighting strategy in more details in Section 3.5. In that section, following Prangle 2017, we will also discuss how the strategy can be developed to adaptive update their weights in each iteration of ABC-PMC and applied to increase the efficiency of ABC-PMC in our study.

There has been much progress regarding summary statistics in ABC scenarios. Some provide strategies to select or construct summary statistics that strike a balance between dimensionality and informativeness (Joyce and Marjoram 2008; Wegmann et al. 2009; Drovandi, Pettitt, and Faddy 2011; Fearnhead and Prangle 2012; Scranton et al. 2014); some concentrate on assigning appropriate weights for different statistics when they are under different scales or not equally informative (Beaumont, W. Zhang, et al. 2002; Luciani et al. 2009; Prangle 2017; Harrison and Baker 2020; Schälte and Hasenauer 2022); some even bypass summary statistics altogether and compare complete datasets directly (Bernton et al. 2017; Gutmann

et al. 2018; Bernton et al. 2019). We refer Blum et al. 2013 and Prangle 2018 which provide detailed reviews of the ABC summary statistics.

3.3 A systematic workflow for selecting summary statistics

In biology, researchers are particularly interested in developing summary statistics that are informative or interpretable (Newman et al. 2022). The selection of summary statistics is often closely related to scientists' particular research interests and the inference problem to be addressed, and therefore, heavily depends on their own knowledge about the studied system (Marjoram, Molitor, et al. 2003; Hey and Machado 2003; Marin et al. 2012). In areas with extensive ABC applications, summary statistics can be selected from previous studies. However, in new applications of ABC, selecting the appropriate summary statistics is more challenging and requires careful consideration (Scranton et al. 2014). For instance, the fixation index, a well-established measure of population differentiation, could be very informative in one model (Wright 1949), but becomes less useful when switching to another new model (Nielsen and Wakeley 2001; Csilléry, Blum, et al. 2010).

In ABC, summary statistics encapsulate essential features and are then compared using a distance metric to quantify dissimilarities. Different distance metrics result in varying acceptance outcomes and, consequently, produce differences in the posterior approximations (Bernton et al. 2019). The choice of distance metric in ABC depends on the nature of the corresponding summary statistic, so it is often necessary to experiment with multiple metrics, comparing the results, to identify the most appropriate one (S. Li and Jakobsson 2012).

To address these challenges in our novel ABC application in IPMs, we propose a set of candidate statistics considering the system's background. An example is provided in Table 3.1. (The table here is just for illustration purpose. Further details will be discussed in Chapter 4.) Each item in the left column of Table 3.1 can be paired with an item on the right to create a unique combination of summary statistic

and distance metric, producing 31 combinations in total. For brevity, we also refer to these combinations as “summary statistics" throughout the rest of the thesis.

Summary statistics	Distance measures
The empirical population distribution in sizeNext groups	Bhattacharyya distance; Hellinger distance; EMD; K-L distance; Hilbert projective metric
Total number of individuals in sizeNext groups; Total number of flowering stalks in size groups; Total number of breeders in size groups; Total number of survivors in size groups;	χ^2 distance metric; Sum of squared differences
Average number of flowering stalks in size groups; Average number of breeders in size groups; Average number of survivors in size groups;	Sum of squared differences
breeders' sizeNext distribution; non-breeders' sizeNext distribution; breeders' and non-breeders' survival distribution	Bhattacharyya distance; Hellinger distance; EMD; K-L distance; Hilbert projective metric

Table 3.1: The list of candidates about the summary statistics and the corresponding distance measures. To calculate the majority of candidate summary statistics under consideration, individuals need to be grouped first according to their size. sizeNext represent the individuals' size in the next year. Please see Appendix B.1.1 for a full description.

To determine the optimal subset of summary statistics, we follow Scranton et al. 2014 to evaluate how well each candidate distinguishes dataset pairs generated from models with ‘slightly different’ hyperparameters. We use the Maximum Likelihood Estimates (MLEs) from the fitted model as a baseline and simulate datasets based on these estimates. Perturbed versions of the MLE model, generated by adjusting hyperparameters, are then used to simulate corresponding datasets for the comparisons.

The standard perturbation approach would vary one hyperparameter at a time, while keeping others fixed. However, as we previously explored in Section 2.2.6, this approach runs into difficulties when applied to GPs, in particular:

- **Hyperparameter interaction:** Hyperparameters in GPs often interact in complex, non-linear ways, where small changes in one can affect others and may amplify their overall impact on the model. Standard perturbation methods fail to capture these interactions, resulting in a mismatch between the scale of the perturbation and the model's response.

- **Dependency on hyperparameter magnitude:** The impact of perturbations depends heavily on the initial hyperparameter value. Different regions of the hyperparameter space may exhibit varying levels of sensitivity, making it difficult to generate consistent and reasonable perturbations.

Given these challenges, we adopt an alternative approach, using Negative Log Posterior Probability (NLPP) scores to define model similarity, that is

$$\text{NLPP} = -\log p(\theta \mid \mathcal{D}_{train}) = -\left[\log p(\mathcal{D}_{train} \mid \theta) + \log \pi(\theta)\right],$$

with the GP hyper-parameters θ, training dataset $\mathcal{D}_{train}$, and prior distribution $\pi(\theta)$.

In the cases of flat prior distributions, since NLPP scores are derived from the log-likelihood of the training data points, similar NLPP scores indicate that the training data points are similarly probable under each model. When users have additional beliefs or knowledge about the hyper-parameters' values before observing the data (informative priors), the NLPP integrates this belief, reflecting it in the score. By keeping the training data points fixed, this method makes stable and consistent comparisons in GP models, alleviating the limitations of parameter-driven perturbation approaches.

Under this setting, models with comparable NLPP scores are considered similar, regardless of differences in their hyperparameter values or predictive surfaces. This method involves varying multiple hyper-parameters simultaneously to create new models with NLPP scores close to the original.

We then evaluate each candidate summary statistic based on its ability to distinguish datasets generated from the model with the lowest NLPP from its perturbed variants. Each summary statistic acts as a classification tool, with its

effectiveness assessed through the area under the receiver operating characteristic (ROC) curve (AUC). The AUC measures classification performance by comparing true positive and false positive rates across varying thresholds. An AUC of 1.0 indicates perfect classification, while an AUC of 0.5 corresponds to random guessing (Hastie et al. 2009). The detailed workflow for this approach is outlined in Workflow 3.1.

Workflow 3.1

For each sub-model modelled by GP (for example, vital rates in our IPMs), we repeat the following steps while keeping the others fixed at MLEs:

1. **GP Model-fitting:** Fit a GP model to the observed dataset within a Bayesian framework. This yields, e.g., 5,000 sets of MCMC samples for hyper-parameters for the target vital rates.

2. **Optimal NLPP:** Determine the NLPP score for each MCMC sample. The one with the smallest score is denoted as $\boldsymbol{\theta}_{opt}$.

3. **Simulations at the Optimum:** With $\boldsymbol{\theta}_{opt}$, simulate, for example, 1,000 pairs of complete datasets through one-step forward projection independently (i.e., 2,000 datasets). For each pair, calculate distances using all 31 candidate summary statistics (listed in Table 3.1), yielding 31 distributions $F^1_{\boldsymbol{\theta}_{opt}}, ..., F^{31}_{\boldsymbol{\theta}_{opt}}$ that represent baseline variability under the optimum.

4. **NLPP Perturbations** & **Re-simulations:** From the MCMC samples, select the top 2% closest to $\boldsymbol{\theta}_{opt}$ in NLPP (e.g., 100 samples), denoted as $\{\boldsymbol{\theta}_k\}_{k=1,...,100}$. Treat them as NLPP perturbation outcomes for $\boldsymbol{\theta}_{opt}$. For each $\boldsymbol{\theta}_k$:

   - Simulate 1,000 datasets under $\boldsymbol{\theta}_{opt}$ and another 1,000 datasets under the $\boldsymbol{\theta}_k$.
   - Pair corresponding datasets (by index) and compute distances across the summary statistics (as in Step 3).

   For each statistic i and each perturbation k, this yields a distribution of distances $F^1_{\boldsymbol{\theta}_1}, F^2_{\boldsymbol{\theta}_1}, ..., F^{31}_{\boldsymbol{\theta}_1}, F^1_{\boldsymbol{\theta}_2}, ..., F^{30}_{\boldsymbol{\theta}_{100}}, F^{31}_{\boldsymbol{\theta}_{100}}$.

5. **Summary statistics Selection:** For each statistic i, compute the AUC value between $F^i_{\boldsymbol{\theta}_{opt}}$ and each of $\{F^i_{\boldsymbol{\theta}_k}\}_{k=1,...,100}$, denoted as AUC^k_i. Then count how many times AUC^k_i exceeds a specified threshold ρ across all k, $\sum_k \mathbb{1}_{\text{AUC}^k_i \geqslant \rho}$. The statistic with the highest count is selected as the most sensitive for the corresponding vital rate.

Notice that, to illustrate the procedure, we describe the workflow using specific numbers (e.g., 1,000 simulations, 2% closest samples). These are illustrative and can be adjusted depending on computational resources and application needs.

As illustrated in step 5 in Workflow 3.1, we prioritize summary statistics that demonstrate consistent performance across different hyperparameter settings (through $\sum_k \mathbb{1}_{AUC_i^k \geqslant \rho}$), rather than those that perform well only in specific situations. This is because, during ABC sampling, model parameters vary across iterations, and statistics that remain informative across a range of plausible models are expected to perform more reliably. Depending on specific modeling goals, alternative selection criteria could also be considered.

In step 5, we set a classification threshold on the AUC and assess how frequently each summary statistic exceeds it across perturbed models. A threshold around 0.75 is often used as a reference for acceptable classification performance (e.g., Hosmer Jr et al. 2013), although, in practice, different thresholds may be applied depending on the study objectives (e.g., Iyer et al. 2016; Gail and Pfeiffer 2018; Çorbacıoğlu and Aksel 2023). Summary statistics that surpass this threshold more frequently are considered more robust and are selected for our subsequent ABC sampling.

3.4 Modified ABC PMC

Conventional ABC-PMC methods (Beaumont, Cornuet, et al. 2009; Toni et al. 2009; S. A. Sisson, Fan, and Tanaka 2009) are powerful, but face challenges in high-dimensional models like the IPM (a combined model) in our case. These limitations include the difficulty of identifying appropriate perturbation kernels and prior distributions, which often lead to high rejection rates and thereby inefficiency in sampling. Our modifications aim to alleviate these issues, improving parameter inference in complex, multi-component models.

We propose a new sampling process: initially using standard Bayesian tools and MCMC samples to gain insights into the posterior distribution, then applying the insights to optimize ABC-PMC efficiency. This approach prioritizes MCMC sample combinations, aiming for higher acceptance rates and faster proposal processes.

Specifically, the computational time for generating 10,000 sampling *proposals* (not *acceptances*) is reduced from at least 20 days to under 1 minute, once MCMC samples are pre-computed and stored. Our developed version of the ABC-PMC method, detailed in Algorithm 3 and Table 3.2, operates as follows:

1. **Initialisation:** Start with particles, drawn from an *informative* prior based on pre-calculated MCMC samples, simulating datasets and calculating initial distances using summary statistics. Set initial weights for all particles equally.
2. **Iterative Process:**
   - For each step $c = 1$ to C, resample particles based on their weights from step $c - 1$ and perturb them by randomly altering one vital rate parameter using MCMC samples.
   - Simulate a new dataset from each perturbed particle and recalculate the distance to the observed data.
   - Accept particles if their recalculated distance is below a dynamically computed quantile-based threshold. Update and normalise weights of accepted particles.
3. **Output:** Continue until the final iteration, producing a weighted set of particles that approximate the posterior distribution.

Compared to the conventional ABC-PMC method (Algorithm 2), we highlight three key modifications for this developed ABC-PMC (Algorithm 3). For more detailed explanations about these modifications please refer to the next section (Section 3.5).

1. **We employ tailor-designed prior distribution $\pi(\boldsymbol{\theta})$ and perturbation kernel $\mathcal{P}(\cdot|\cdot)$.** After obtaining MCMC samples for each sub-model individually — to gain insights into the posterior distributions of their own parameters, we then

- $\pi(\cdot)$: re-use the MCMC samples for each individual sub-model to build up an informative prior for the entire big model. Specifically, after obtaining, for example, M different MCMC samples for each of V individual sub-models, the resulting informative prior has a support (sample space) containing M^V possible points.
- $\mathcal{P}(\cdot|\cdot)$: Given a particle ($\boldsymbol{\theta}_{c-1}^i$) accepted in the previous iteration, a new movement is proposed by randomly selecting one of the V sub-models and replacing all corresponding parameters with a randomly chosen sample from the M MCMC samples (Line 15 in Algorithm 3).

2. Illustrated by the toy example in Section 3.2, through an adaptive weighting strategy proposed by Prangle 2017,**the selected summary statistics are combined into a single 'universal' distance** to measure the difference between the observed and simulated dataset. In summary, the 'universal' distance between the observed and a simulated dataset at cth iteration is defined by

$$\widetilde{d}^* = \left[\sum_{i=1}^{\mathscr{d}_s} \left(\frac{{}^i\widetilde{S}^*}{{}^i\sigma_{c-1}}\right)^2\right]^{1/2}, \tag{3.4}$$

where ${}^i\sigma_{c-1}$ is the estimated median absolute deviation (MAD) (based on $c-1$th generation of particles) for the ith summary statistic ${}^i\widetilde{S}^*$. Notice that, for simplicity, we refer to ${}^i\widetilde{S}^*$ as a 'summary statistic'. However, it actually represents the distance between two summary statistics: one calculated from the simulated dataset and the other from the observed dataset. The vector, defined through the list of selected combinations of summary statistics and distance metrics, is thus expressed as $\widetilde{\boldsymbol{S}}^* = ({}^1\widetilde{S}^*, ..., {}^{\mathscr{d}_s}\widetilde{S}^*)$. For clarity and simplicity, we will continue to use this term and refer to it as 'summary statistics' throughout the report.

3. Furthermore, we pay attention to studying various strategy about acceptance regions, and ultimately choose **a decreasing sequence of quantiles** $\{\alpha_i\}_{0\leqslant i\leqslant C-1}$**.** To balance computational time, an additional modification is made by **gradually reducing total number of accepted particles required,** $\{N_i\}_{0\leqslant i\leqslant C}$. A

proper balance between these two factors enable us to propagate only the most promising particles, providing stability for the future samples in Algorithm 3.

It is worth noting that, while both the sub-model inference and the ABC posterior approximation rely on the same underlying dataset, they draw on different levels of information. The sub-models are fitted using individual-level observations — like age, size, and reproductive outcomes — whereas the ABC step compares simulations to observations using population-level summary statistics that do not retain individual trajectories (e.g., empirical distributions in sizeNext groups or aggregated breeder survival rates in our case studies). As such, the information used to estimate the sub-model posteriors is not directly reused in the ABC sampling. Although this setup does not represent a fully independent use of data, we consider the two stages to target distinct aspects of the dataset. In this sense, it differs from cases where the same data are reused twice.

Notation	Description
$\mathcal{M}$	The target combined model.
$\top$	The transpose of a matrix.
C	The total number of simulation steps for ABC-PMC (apart from the initialisation).
c	The current simulation step; $c = 0$ for initialisation.
N_c	The number of accepted particles required for the simulation step c.
α_c	The quantile employed to decide the acceptance region for the next simulation $c + 1$.
V	The total number of sub-models in the target combined model.
v	A realisation of a discrete uniformly distributed random variable indicating the current sub-model.
M	The number of MCMC samples for each of the V sub-models. So, there are M^V different combinations in total.
m_v	Given the current sub-model v, m_v is a realisation of a discrete uniformly distribu--ted random variable representing m_vth MCMC sample for vth sub-model.
$\boldsymbol{\theta}$	A particle: a vector including parameters for all of the sub-models.
$\boldsymbol{\theta}_c^i$	The ith particle at cth simulation step.
$\widetilde{\boldsymbol{\theta}}$	An intermediate particle.
$\widetilde{\boldsymbol{\theta}}_v$	A vector only includes vth sub-model's parameters. So, $\widetilde{\boldsymbol{\theta}}^\top = (\widetilde{\boldsymbol{\theta}}_1^\top, ..., \widetilde{\boldsymbol{\theta}}_V^\top)$.
$\widetilde{\boldsymbol{\theta}}^*$	A perturbed intermediate particle.
$\pi(\boldsymbol{\theta})$	The prior for $\boldsymbol{\theta}$, which is the joint distribution for m_1, ..., m_V.
ω_c^i	Unnormalised weight for $\boldsymbol{\theta}_c^i$
w_c^i	Normalised weight for $\boldsymbol{\theta}_c^i$
y^*	A simulated dataset.
y_{obs}	The observed dataset.
$\mathscr{d}_s$	Dimension of $\boldsymbol{S}_c^i$ and $\widetilde{\boldsymbol{S}}^*$.
$\boldsymbol{S}_c^i$	$\boldsymbol{S}_c^i = ({}^1S_c^i, {}^2S_c^i, {}^3S_c^i, {}^4S_c^i..., {}^{\mathscr{d}_s}S_c^i)$, where $({}^jS_c^i)$ is jth summary statistics for ith particle at step c.
$\widetilde{\boldsymbol{S}}^*$	A vector of summary statistics produced by $\widetilde{\boldsymbol{\theta}}^*$, $\widetilde{\boldsymbol{S}}^* = ({}^1\widetilde{S}^*, ..., {}^{\mathscr{d}_s}\widetilde{S}^*)$.
$\boldsymbol{\sigma}_c$	A vector of MADs for all of the selected summary statistics, $\boldsymbol{\sigma}_c = ({}^1\sigma_c, ..., {}^{\mathscr{d}_s}\sigma_c)$.
d_c^i	$d_c^i = (\sum_{j=1}^{\mathscr{d}_s}({}^jS_c^i/{}^j\sigma_c)^2)^{1/2}$, which represent the 'universal' distance between y_{obs} and a simulated dataset produced by $\boldsymbol{\theta}_c^i$.
$\widetilde{d}^*$	$\widetilde{d}^* = (\sum_{j=1}^{\mathscr{d}_s}({}^j\widetilde{S}^*/{}^j\sigma_{c-1})^2)^{1/2}$.
δ_c	α_cth quantile of $\{d_c^i\}_{1 \leqslant i \leqslant N_c}$; the quantile-based acceptance threshold at step c.
$\mathbb{1}_\tau$	An indicator function, $\mathbb{1}_\tau = 1$ if the statement τ is true; $\mathbb{1}_\tau = 0$ otherwise.
$a \rightarrow b$	This is a statement about whether it is possible to transit from state a to state b by the perturbation kernel at line 15 in Algorithm 3.

Table 3.2: The description of notations used in Algorithm 3.

Algorithm 3: ABC-PMC (see Table 3.2 for the descriptions of the notations)

Input : Simulation length C; quantiles $\{\alpha_i\}_{0\leqslant i\leqslant C-1}$; numbers of particles required $\{N_i\}_{0\leqslant i\leqslant C}$; M individual MCMC samples for each of the sub-models.

Sampling :

$c = 0$: initialisation

for *i = 1, ..., N_0* **do**
  Sample $\boldsymbol{\theta}_0^i \sim \pi(\boldsymbol{\theta})$;
  Simulate a dataset $y^* \sim \mathcal{M}(\boldsymbol{\theta}_0^i)$;
  Calculate the distance vector $\boldsymbol{S}_0^i$ between y^* and y_{obs}.
end
Calculate MADs $\boldsymbol{\sigma}_0$ based on $\{\boldsymbol{S}_0^i\}_{1\leqslant i\leqslant N_0}$;
Combine the elements on $\boldsymbol{S}_0^i$ into a single quantity $\{d_0^i\}$ through $\boldsymbol{\sigma}_0$ for $1 \leqslant i \leqslant N_0$;
Calculate the quantile-based acceptance threshold δ_0;
Set equal weight for each particle, $\omega_0^1 = \omega_0^2 = ... = \omega_0^{N_0} = 1$;
Normalise weights to have $\{w_0^i\}$ for $1 \leqslant i \leqslant N_0$;
$c \leftarrow c + 1$.

$c \geqslant 1$: for each simulation step $c = 1, ..., C$,

$i \leftarrow 1$
while $i \leqslant N_c$ **do**
  Sample $\widetilde{\boldsymbol{\theta}}$ from $\{\boldsymbol{\theta}_{c-1}^i\}_{1\leqslant i\leqslant N_{c-1}}$ with probability $\{w_{c-1}^i\}_{1\leqslant i\leqslant N_{c-1}}$;
  Perturb $\widetilde{\boldsymbol{\theta}}$:
  Draw v from a discrete uniform distribution $[1, 2, ..., V]$;
  Draw m_v from a discrete uniform distribution $[1, 2, ..., M]$;
  Let $\widetilde{\boldsymbol{\theta}}^* = \widetilde{\boldsymbol{\theta}} = \{\widetilde{\boldsymbol{\theta}}_1^\top, ...\widetilde{\boldsymbol{\theta}}_V^\top\}$;
  Replace $\widetilde{\boldsymbol{\theta}}_v^\top$ in $\widetilde{\boldsymbol{\theta}}^*$ by the m_vth MCMC sample of vth sub-model.
  Simulate a dataset $y^* \sim \mathcal{M}(\widetilde{\boldsymbol{\theta}}^*)$;
  Calculate the distance vector $\widetilde{\boldsymbol{S}}^*$ between y^* and y_{obs};
  Calculate $\widetilde{d}^*$ (for inter-mediate particle $\widetilde{\boldsymbol{\theta}}^*$), based on $\boldsymbol{\sigma}_{c-1}$ and $\widetilde{\boldsymbol{S}}^*$;
  if $\widetilde{d}^* \leqslant \delta_{c-1}$ **then**
    Accept the particle $\widetilde{\boldsymbol{\theta}}^*$, set $\boldsymbol{\theta}_c^i = \widetilde{\boldsymbol{\theta}}^*$ and $\boldsymbol{S}_c^i = \widetilde{\boldsymbol{S}}^*$;
    Calculate the weight $\omega_c^i \propto \frac{1}{\sum_{j=1}^{N_{c-1}} w_{c-1}^j \mathbb{1}_{\{\boldsymbol{\theta}_{c-1}^j \to \boldsymbol{\theta}_c^i\}}}$;
    $i \leftarrow i + 1$
  else
    Reject $\widetilde{\boldsymbol{\theta}}^*$ and store the distance vector $\widetilde{\boldsymbol{S}}^*$;
    $i \leftarrow i$
  end
end
Normalise weights to have $\{w_c^i\}$ for $1 \leqslant i \leqslant N_c$;
if $c = C$ **then**
  Return the final particle population $\{\boldsymbol{\theta}_C^i\}_{1\leqslant i\leqslant N_C}$ with the corresponding normalised weights $\{w_C^i\}_{1\leqslant i\leqslant N_C}$.
else
  Calculate new MADs $\boldsymbol{\sigma}_c$ based on all the particles (including those rejected);
  Calculate the 'universal' distance for each particle $\{d_c^i\}_{1\leqslant i\leqslant N_c}$ based on $\boldsymbol{\sigma}_c$ and $\{\boldsymbol{S}_c^i\}_{1\leqslant i\leqslant N_c}$;
  Calculate the quantile-based acceptance threshold δ_c;
  $c \leftarrow c + 1$
end

3.5 Detailed explanations for the ABC-PMC Modifications

This section describes the motivations and thought process behind the three modifications made to our developed ABC-PMC. We aim to share why and how we decided to make these adjustments in our case, which may be helpful for others looking to apply or develop these methods in their studies.

1. We utilise a tailor-designed prior distribution $\pi(\boldsymbol{\theta})$ and perturbation kernel $\mathcal{P}(\cdot|\cdot)$ in our implementation of ABC-PMC to better suit our specific case of combined models with highly flexible sub-models, like GP models.

   The problem:

   The practical application of GP models has long been hindered by the computational complexity involved (Cheng and Boots 2017; Schulz et al. 2018). In our case of implementing ABC, obtaining a value for GP hyperparameters is (nearly) cost-free since a value can be simply drawn from the proposal distribution rather than being estimated. However, simulating populations using those specified hyperparameter values remains very expensive. The computational complexity is dominated by computing the GP predictive distributions, which require a complexity of $\mathcal{O}(N^3)$ when solving or approximating inference with N training data points through the Cholesky decomposition.

   In our experiments of IPMs, the entire process — manually feeding an assigned value of $\boldsymbol{\theta}$ into five GP sub-models for the five vital rates (processed in serial) and preparing them ready for the simulations — took about 3 minutes on average on a machine equipped with the 10-core Apple M1 Max processor. In this way, it would take at least 20 days to achieve 10,000 attempts. It is worth mentioning that in most ABC-PMC applications, 10,000 *acceptances* are not sufficient for one iteration, let alone for just 10,000 *attempts*.

   The situation can become even worse when $\boldsymbol{\theta}$ is of high dimension in combined models, as interactions between sub-models can amplify the effects of any

inconsistency. Considerable workloads on computing resources can be exerted by the choice of inefficient perturbation kernels during the sampling process (Filippi et al. 2013). In our scenario, with the commonly adopted multivariate normal perturbation kernel (e.g. Beaumont, Cornuet, et al. 2009), a dramatically high rejection rate was observed. The proposal movements, even if slightly inconsistent or inappropriate, could generate very unrealistic vital rates, resulting in simulated populations that often did not resemble the observations. The acceptance rate was less than 0.1% — this was the case even when the movements were proposed based on a population of very promising particles and with a very gentle acceptance threshold determined by background knowledge.

In a high dimensional problem, designing a perturbation kernel that can efficiently move across all dimensions of $\boldsymbol{\theta}$ simultaneously is challenging (Everitt and Rowińska 2021). Given such a high rejection rate, one can imagine the immense computational time required as the process of computing approximations in GP models is repeated millions and billions of times in the conventional ABC-PMC. Even with parallel computing on a powerful 64-core server, completing the ABC model fitting within a reasonable time frame may still be unfeasible.

The solution:

Recall that, in combined models, the most commonly adopted model fitting process comprises two key steps. Firstly, each sub-model is independently parameterized from the others. Secondly, these multiple pieces of individual models are integrated to construct a comprehensive model. This procedure reminds us of a major distinction between the combined models and some other complicated models with analytically intractable likelihoods. Being a combined model, it can therefore also be decomposed back into those simpler baseline models, each pertaining to individual sub-model. For those sub-models, particularly the GP models in our study, a plenty of robust tools have been developed to efficiently draw samples from the respective individual target posterior. Why not follow the initial step of the standard combined model

fitting procedure to gain a rough idea about the posterior distribution of the parameters, for example, by obtaining MCMC samples? Then, we can re-use these MCMC samples for each individual vital rate (serving as our 'background knowledge' about $\boldsymbol{\theta}$) to construct a more efficient ABC-PMC for exploring their joint behaviours.

In other words, instead of using non-informative priors, like a high-dimensional uniform distribution with wide boundaries $\mathcal{U}(\boldsymbol{\theta})$, we can specify our prior for the hyperparameter vector $\boldsymbol{\theta}$ to be confined by a set of points $\pi(\boldsymbol{\theta}) \propto \mathcal{U}(\boldsymbol{\theta})\mathbb{1}(\boldsymbol{\theta} \in \mathcal{X})$. Here, the hyper-set $\mathcal{X} \in \mathbb{R}^{dim(\boldsymbol{\theta})}$ includes all combinations of different groups of MCMC samples and represents our belief in a broad region where the target $\pi_{ABC}(\boldsymbol{\theta}|\boldsymbol{s}^{obs})$ is likely to be found. After obtaining, for example, M different MCMC samples for each of V individual sub-models, hyper-set $\mathcal{X}$ contains M^V possible points.

To avoid a dramatically high rejection rate resulting from inappropriate proposals, it is also necessary to re-design the perturbation kernel $\mathcal{P}(\cdot|\cdot)$. Generally, the perturbation kernel in ABC serves two primary purposes: (i) guiding particles towards regions of high posterior probability, and (ii) enhancing particle diversity (thereby facilitating a more extensive exploration of the parameter space) (Filippi et al. 2013; S. Sisson and Fan 2018). In our special case, we also require a perturbation kernel from which new particles can be easily and readily sampled, and, meanwhile, computing the transition density between any two particles should not impose great additional computational load.

Drawing upon the MCMC samples, we formulate a straightforward component-wise perturbation kernel $\mathcal{P}(\cdot|\cdot)$. Given a particle accepted in the previous iteration, $\boldsymbol{\theta}^i_{c-1}$, a new movement is proposed by randomly selecting one of the V sub-models and replacing all corresponding parameters with a randomly chosen sample from the M MCMC samples (Line 15 in Algorithm 3). Accordingly, the resulted proposal distribution becomes to $q_c(\boldsymbol{\theta}^*) = \sum_j w^j_{c-1}\mathbb{1}_{\{\boldsymbol{\theta}^j_{c-1} \to \boldsymbol{\theta}^*\}}$ for iteration $c = 1, ..., C$, where w^j_{c-1} is the corresponding weight for particle $\boldsymbol{\theta}^j_{c-1}$

accepted in the previous iteration. $\boldsymbol{\theta}_{c-1}^{j} \rightarrow \boldsymbol{\theta}^{*}$ is a statement about whether it is possible to transit from point $\boldsymbol{\theta}_{c-1}^{j}$ to point $\boldsymbol{\theta}^{*}$ through the perturbation kernel (Line 21 in Algorithm 3).

In each perturbation, the kernel $\mathcal{P}(\cdot|\cdot)$ is designed to make only small adjustments to increase the acceptance probability. It can explore around the current particles more thoroughly, accepting only those new particles that can produce a simulation similar to the observed one. After obtaining the required number of acceptances, the proposal distribution q_c (in Equation 3.3) is automatically updated by incorporating these more promising particles, so that, it will be able to propose values that are more likely to lie in high posterior regions for the next iteration. Through this iterative process, q_c based on the proposed $\mathcal{P}(\cdot|\cdot)$ is expected to continually improve and to ultimately evolve into a highly efficient proposal distribution for the final round.

When using ABC-PMC with this configuration, the computational overhead associated with the approximations in GP models is no longer a big concern. The parameters for these approximations have already been calculated during MCMC simulations, so all that is needed is additional storage space to cache the MCMC results locally and retrieve them as needed. Similarly, for the Cholesky decomposition required to compute predictive distributions, due to the use of a fixed set of MCMC samples for each vital rate, predictions with the same MCMC sample occur much more frequently. In such cases, pre-doing the calculations that are independent of the test points and reusing them between predictions can substantially alleviate the computational burden. Additionally, there is also no need for additional resources to calculate transition probabilities between particles, as they will be a constant of $1/(VM)$.

2. As a toy example demonstrated in Section 3.2, **an adaptive weighting strategy** is employed to combine selected summary statistics into a single 'universal' distance, measuring the difference between observed and simulated datasets.

   The problem:

As detailed in Section 3.1, several distinct summary statistics should be chosen to represent various aspects of the dataset: if only one type of the summary statistics is employed in ABC-PMC, the sampler may be excellent for estimating the parameters for a particular sub-model, but can be very poor and less-constrained for the others (Hamilton et al. 2005). Subsequently, for parameterising multiple sub-models simultaneously, incorporating all the selected summary statistics as a multi-dimensional vector into the ABC sampler is essential, ensuring a comprehensive representation of the observed dataset.

Typically, ABC samplers use a “rectangular" acceptance region when dealing with multi-dimensional summary statistics (Pritchard et al. 1999; Scranton et al. 2014). This extends the smoothing kernel K_{h_c} in Equation 3.3 to be a product of indicator functions, i.e. $K_{h_c} = \prod_{i=1}^{\mathscr{d}_s} K_{h_c}^i = \prod_{i=1}^{\mathscr{d}_s} \mathbb{1}_{{}^i\widetilde{S}^* \leqslant h_c^i}$, where $\widetilde{\boldsymbol{S}}^* = ({}^1\widetilde{S}^*, ..., {}^{\mathscr{d}_s}\widetilde{S}^*)$ and h_c^i is the bandwidth for ith summary statistic ${}^i\widetilde{S}^*$ in cth iteration in the ABC-PMC. The method assumes equal contributions from all summary statistics to the estimation of the target posterior when accepted.

However, when summary statistics are constrained to be interpretable in reality, they may carry overlapping information, causing some to be less informative in the presence of others. This complicates the selection of appropriate thresholds $\{h_c^i\}_{i=1,...,\mathscr{d}_s}$ to prevent less-informative statistics from over-affecting the parameter estimations and ensure that all statistics contribute equally. Additionally, applying $K_{h_c} = \prod_{i=1}^{\mathscr{d}_s} K_{h_c}^i$ in our study may suffer from the curse of dimensionality. Despite our efforts to keep the number of summary statistics as small as possible, having a dimension such as $\mathscr{d}_s = 25$ is still quite large (one summary statistic per vital rate over five years). With such a high-dimensional vector of summary statistics, the ABC algorithm inevitably faces a very high rejection rate in the “rectangular" acceptance region, as it only accepts particles that meet all conditions simultaneously. These challenges further complicate the estimation of parameter values.

To alleviate the computational overhead due to the high rejection probability in the "rectangle" , Beaumont, W. Zhang, et al. 2002 combined the summary statistics through weighted L^2 norm, $\widetilde{d}^* = \left[\sum_{i=1}^{\mathscr{d}_s} \varpi_i({}^i\widetilde{S}^*)^2)\right]^{1/2}$, transforming the region into a "sphere" to increase the acceptance rate for the samples. To make the normalisation in a more appropriate way, the weight ϖ_i of the ith summary statistic is usually taken as the inverse of its estimated dispersion measure calculated from the prior predictive distribution, such as the empirical standard deviation (Beaumont, W. Zhang, et al. 2002), for other examples, see Hamilton et al. 2005; Luciani et al. 2009; Erhardt and S. A. Sisson 2016; Prangle 2017; Rodrigues et al. 2018.

During our studies on IPMs, we found that certain hyperparameter combinations could yield biologically unrealistic vital rates, like an extremely high survival rate for the most reproductive group (which is not typically observed in reality). This discrepancy resulted in huge divergence between simulated and the observed populations. To address this issue, we normalise the summary statistics using the median absolute deviation (MAD) as suggested by Csilléry, François, et al. 2012. Normalising through MADs is less sensitive to those large discrepancies (i.e. outliers) and can also intuitively reduce the impact of the summary statistics that are running on different scales, tending to stop the acceptance rules being dominated by the most variable summary statistic (Prangle 2017; Schälte and Hasenauer 2022).

However, given the iterative nature of ABC-PMC, detailed in Section 3.1.2, it appears inappropriate to employ the standard scaling strategy of utilising consistent MADs (which are estimated from the *initial* prior predictive distribution) throughout the *entire* ABC-PMC process. This is because the objective function will actually keep changing in each iteration, when, for example, a decreasing sequence of thresholds $\{h_c\}_{c=0,\dots,C}$ is employed. In this case, later rounds of ABC-PMC may yield realisations originating from distributions vastly different from the initial prior predictive distribution, making the normalising strategy meaningless. To prevent this problem in our version of ABC-PMC, we adopt an

adaptive strategy to update the MADs in each iteration according to Prangle 2017.

The solution:

The idea behind this modification is straightforward: updating the MADs based on simulations generated in the previous iteration to achieve more sensible normalisation in each round. Specifically, in Algorithm 3, we incorporate all particles in the previous iteration (including those rejected), to calculate the MADs. The reason for including rejected particles is to prevent bias when the target distributions between consecutive iterations are very different. If we rely only on accepted particles, the MADs might primarily reflect the criteria used in the previous iteration, which could no longer match the current target distribution. Including all particles would instead be less restrictive and can provide a more comprehensive picture of the samples generated based on the intermediate proposals. This helps MADs to be more representative of the broader sample set and not overly influenced by past criteria (Prangle 2017).

Note that this modification may still be inadequate if the targets distributions in two successive iterations differ dramatically. In such cases, an alternative approach proposed by Prangle 2017 could be considered. Generally speaking, this alternation involves treating the accepted particles in Algorithm 3 as 'intermediate' particles, and using them to estimate the MADs for normalizing summary statistics generated by an additional layer of simulations. However, this approach requires a large enough set of 'inter-mediate' particles to produce reliable MAD estimations, which can impose a substantial computational burden.

In our study, the rejection rates were already extremely high in the latter rounds of Algorithm 3 (see Table B.7 in the latter section about real case study). With a fixed computation budget, the additional cost required by introducing 'intermediate' particles prevented us from using small values in acceptance thresholds $\{h_c\}_{c=0,\ldots,C}$. This might be the reason why this method performed even worse than Algorithm 3 in our experiment (not shown). Ultimately, we chose to stick

with Algorithm 3 for the latter work, as a trade-off for the limited computational resources.

In summary, the 'universal' distance between the observed and a simulated dataset at cth iteration is defined by

$$\widetilde{d}^* = \left[\sum_{i=1}^{d_s} \left(\frac{{}^i\widetilde{S}^*}{{}^i\sigma_{c-1}}\right)^2\right]^{1/2},$$

where ${}^i\sigma_{c-1}$ is the estimated median absolute deviation (MAD) (based on $c-1$th generation of particles) for the ith summary statistic ${}^i\widetilde{S}^*$.

3. Additionally, the last modification concerns the quantile used to decide the **acceptance region**. Employing a constant quantile α in Algorithm 3 brings us a potential risk: the determined acceptance regions might not originate from subsets in a nested sequence converging to $\widetilde{\boldsymbol{S}}^* = 0$ (Prangle 2017; Schälte, Alamoudi, et al. 2021). To make particles from each successive iteration less diffuse, we explore various schedules and ultimately choose decreasing sequence of quantiles $\{\alpha_i\}_{0\leqslant i\leqslant C-1}$ — to ensure that $\delta_i < \delta_{i+1}$ for $0 \leqslant i < C-1$ and, therefore, expect that the populations of accepted particles gradually converge to the final target distribution in the limit (Toni et al. 2009).

   Under this quantile sequence, rejection rates could increase remarkably as α_i decreased. However, we only have a fixed computational budget that can be allocated at each simulation step. To balance computational time, we have to make an additional modification by gradually reducing the total number of accepted particles required ($\{N_i\}_{0\leqslant i\leqslant C}$). Under a reasonable computational time, we should turn up the N_0 as large as possible. We then progressively reduce N_i for the subsequent iterations in a controlled manner to ensure that the decrease is not too rapid. This is important because a rapid decrease in $\{N_i\}_{0\leqslant i\leqslant C}$ would not provide enough time to gradually reduce α_i, which is necessary to maintain a sufficient number of accepted perturbation (Marin et al. 2012). By managing the reduction, we are able to obtain a final generation of accepted particles that is still of a reasonable size. In addition, we would also ensure that the decrease

is not too slow. This is because, given the accepted particles in the previous iteration, the total number of possible states of $\boldsymbol{\theta}$ is fixed under our perturbation kernel (see Table 4.3 for more details). A proper reduction in the number of acceptances allows us to propagate only the most promising particles, those with a high chance of $\widetilde{d}^* \leqslant \delta_{c-1}$ in step 19 of Algorithm 3, providing stability for the future samples. Please check Table 4.3 and B.7 for specific values of $\{\alpha_i\}$ and $\{N_i\}$ employed in the practical case studies.

4

ABC_GP IPM with Case study 1: *Cryptantha Flava*

Contents

4.1 ABC_GP IPM 57
4.2 The study species 58
4.3 Simulation case study of the enhanced IPM 60
4.3.1 Simulation setup 61
4.3.2 Evaluation metrics 62
4.3.3 Results 63
4.4 Application of the enhanced IPM with real datasets . 73
4.4.1 Results 74
4.5 Discussion 80

In this chapter, we begin by introducing our enhanced version of IPM, incorporating GP models and the developed ABC-PMC sampler (Section 4.1). Next, we provide background information about our case study species, *Cryptantha flava* (Section 4.2). We review its growth-reproduction trade-off and the influence of size and age on its vital rates, laying the groundwork for further modeling tasks. To explore the application of ABC_GP in IPMs, Section 4.3 presents a comparative evaluation of traditional IPMs and our enhancements through simulated datasets with known 'truth'. This validation step is crucial for assessing the reliability of our methods under controlled manners. Subsequently, with the pronounced

insights gained from the simulation analyses, we extend our methodology to real-world datasets of *C. flava* in Section 4.4, demonstrating its practical viability and effectiveness in forecasting population trends. This chapter concludes in Section 4.5.

4.1 ABC_GP IPM

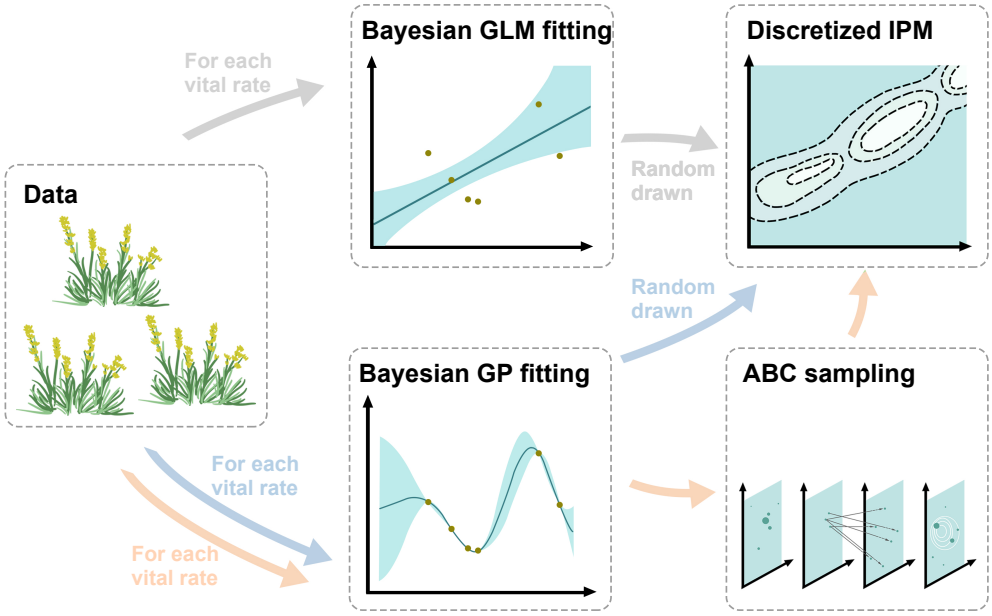


Figure 4.1: Workflow chart illustrating the enhanced IPM construction process: This diagram demonstrates the integration of GP models and ABC in the development of IPMs. Grey arrows depict the steps involved in constructing a traditional Bayesian IPM. Blue arrows highlight the replacement of standard GLMs with GP models. Orange arrows illustrate the innovative application of the ABC methods to synthesise GP IPMs. Please note that all data points and content in this chart are for illustrative purposes only and do not represent any actual data values or practical implementations.

In general, constructing a standard Bayesian IPM involves two main steps (before discretizing it into matrices): (i) fitting each individual vital rate independently using Bayesian methods to obtain MCMC samples; (ii) randomly drawing samples from these fitted models, and combining them to form an IPM kernel based on the life cycle of the target species. This construction process is illustrated by the grey arrows in the Figure 4.1. In response to the three identified limitations in Chapter 1, we propose enhancing IPMs by leveraging the power of GP and ABC.

Our methods innovates upon the traditional Bayesian IPM construction process, illustrated by the blue and orange arrows in Figure 4.1.

1. **Integration of GP models:** We begin by replacing traditional base models, such as GLMs, with GP models. This method, indicated by the blue arrows, involves fitting GP models to each of the vital rates independently under a Bayesian framework, using individual-level data similar to traditional IPM practices.

2. **Advanced synthesis using ABC-PMC:** Moving beyond typical Bayesian IPM approaches, which construct IPMs by assembling randomly drawn MCMC samples of each vital rate, we have further developed our strategy by utilising the ABC-PMC sampler. The model construction process is illustrated by the orange arrows in Figure 4.1.

The inclusion of GP models introduces flexibility and provides probabilistic outputs, handling inherent nonlinearities and variabilities in natural systems. These models mitigate structural sensitivity issues (Wood 2001) and are particularly beneficial when data is limited. Furthermore, the synthesis of the ABC method crafts IPMs by retaining only combinations of vital rates that reproduce observed characteristics. By incorporating population-level information, our approach recalibrates the individually fitted vital rate models to produce joint behaviour that better reflects observed population dynamics.

4.2 The study species

Cryptantha flava is a yellow-flowered perennial subshrub growing in the sandy soils of the Canyonlands region and Colorado Plateau, southwestern of USA (Fagan 2012). The studied population was sampled by Lucas et al. 2008 and Salguero-Gomez et al. 2012, near the Red Fleet State Park in northeastern of Utah state, USA. *C. flava* was selected due to its extensive coverage in prior research with well-documented life history traits (e.g., B. B. Casper 1988; Lucas et al. 2008; Salguero-Gomez et al. 2012; González et al. 2016; Evers et al. 2021), providing a solid foundation for

our simulations and real data analyses. The dataset is comprehensive, spanning a considerable temporal range from 2003 to 2012, and includes detailed tracking of over 1,000 individuals. These attributes make *C. flava* an ideal candidate for testing and comparing the robustness and flexibility of our IPM methodology.

The plant adjusts its size through shrinkage or growth — shrinkage by losing existing rosettes from either flowering or mortality; growth by producing new rosettes from axillary meristems (Salguero-Gomez 2011). At the study site, vegetative growth for the plant begins in late March. A newborn rosette can continue to develop foliage for several years until it produces an erect flowering stalk (Peek and Forseth 2003; B. B. Casper et al. 2006). The inflorescence is initiated in spring and begin to bloom from mid-May. The flowering stalk and the corresponding flowering rosette (but not the whole plant) would senesce and die after seeds ripen around July (B. B. Casper 1996). This naturally non-independent relationship between growth and fertility in *C. flava* — shrinkage happening due to the reproduction — underscores the necessity of distinct growth models for reproducers and non-reproducers.

As a desert plant, *C. flava* is frequently studied for its response to drought. Observations indicate that plants at different stages of age and size respond differently under drought conditions. During drought years, smaller individuals tend to continue developing new leaf rosettes, while larger ones often experience die-back (B. B. Casper 1996). This may be due to structural differences of hydraulic system between juveniles and adults (Salguero-Gómez and B. B. Casper 2011). For larger or older individuals, natural selection during dry years favours those with higher water-use efficiency, but no evidence is found to support this pattern among juveniles (B. Casper et al. 2005; Donovan et al. 2007). Additionally, shrinkage typically occurs for several years before a *C. flava* dies, which might suggest an age-related decline in physiological function (Partridge and Barton 1993; B. B. Casper et al. 2006). Compared to similar-sized individuals without prior die-back, those with shrinkage history show reduced flowering and growth (B. B. Casper et al. 2006), possibly indicating the influence of age on vital rates (as older individuals would have more chances to undergo shrinkage).

Despite the insights into age- and size-dependent living strategies in *C. flava* and other species (Cavender-Bares and Bazzaz 2000; Roach 2012; Quarles and Roach 2019), there are surprisingly less works implemented to study plant population by considering both of age and size effect (but see Childs et al. 2003; Chu and Adler 2014; Edelfeldt et al. 2019; Baden et al. 2021). This may be due to several reasons. Firstly, it is expensive; long-term individual-level tracings are essential to grasp information about age. Because the plants fluctuate between growth and shrinkage independently of age, it is almost impossible to estimate age directly from development (Salguero-Gómez, Shefferson, et al. 2013). Secondly, it could also be a case of 'not necessary'. On the one hand, studies found that size has a more important effect on vital rates than age for some species (Mencuccini et al. 2005; Lauenroth and Adler 2008). On the other hand, the model structures for joint effects are usually more complex than those for a marginal effect; considerable effort is also required in coding the model system. Therefore, the size effect usually has priority over the age effect, if can only pick one of two. Studying population dynamic by only considering size effect is popular in current biology studies.

The biological insights provided above guide our decision to develop an age-size structured IPM for *C. flava*, with separate growth functions for breeders and non-breeders. While age is not included in the simulation studies for simplicity, it is incorporated into the real case study to better capture the combined effects of age and size on vital rates.

4.3 Simulation case study of the enhanced IPM

To demonstrate the application of ABC and GP in IPM, we conducted a case study using simulated datasets for *C. flava*. In addition, we expanded our analysis by incorporating GLMs as well, allowing us to compare our GP-based models with IPMs that are grounded in traditional methodologies.

Leveraging the advantage of simulated data, where the "truth" is known, we evaluated the effects of misrepresenting non-linear relationships as linear and vice versa, to assess the adaptability and effectiveness of different modeling strategies.

We simulated two datasets based on linear and non-linear assumptions and fitted each with both model types to compare performance. We analysed the resulting IPMs, enabling a comprehensive comparison of the three modelling strategies shown in Figure 4.1:

1. **GLM:** fitting vital rate through Bayesian GLMs, constructing IPMs through randomly drawn MCMC samples;
2. **GP:** fitting vital rate through Bayesian GPs, constructing IPMs through randomly drawn MCMC samples;
3. **ABC_GP:** fitting vital rate through Bayesian GPs, constructing IPMs through ABC-PMC.

Furthermore, we examined the efficiency of fitting ABC_GP IPMs through the proposed ABC-PMC algorithm, revealing a pattern of increasing accuracy across iterations. This provides key insights into why our method produced noticeably improved results.

4.3.1 Simulation setup

We initiated our experiment by applying GP models and GLMs within a frequentist framework to fit the real-world population data (aggregated from the three years from 2008 to 2011). Using these models, denoted by M_{gp} and M_{glm}, we conducted IBM simulations to project population dynamics over a decade, creating two simulated datasets: D_{gp} and D_{glm}. These datasets were treated as surrogate 'truths' for subsequent analysis. Table 4.1 summarizes these models.

Using D_{gp} and D_{glm}, we constructed three types of IPMs (GLM, GP, and ABC_GP) to explore the implications of inaccurately modeling non-linear relationships as linear, and vice versa. In the ABC_GP model fitting, we did not borrow the prior knowledge from M_{gp} and M_{glm} to compute sufficient summary statistics, instead following our proposed workflow (Workflow 3.1) to emulate a more generalised modelling process in real practice. Detailed implementation can be found in Appendix B.1.2.

Demographic process	**Simulated variable**	**Model**	**Parameter estimates**
Probability of survival	$Surv \sim Bern(s)$	GLM	$\text{logit}(s) = -0.024 + 0.456z$;
		GP	$\text{logit}(s) = f_{GP}$, $r^2 = 0.399, l = 0.978$.
Probability of reproduction	$Repr \sim Bern(p_b)$	GLM	$\text{logit}(p_b) = -5.340 + 2.501z$;
		GP	$\text{logit}(p_b) = f_{GP}$, $r^2 = 16.87, l = 4.10$.
Number of flowering stalks	$Flow \sim Poi(n_f)$	GLM	$\ln(n_f) = -2.232 + 1.115z$;
		GP	$\ln(n_f) = f_{GP}$, $r^2 = 7.037; l = 2.628$.
Growth for reproducers	$Z' \sim Norm(\mu_r, \sigma_r^2)$	LM	$\mu_r = 0.656 + 0.767z$, $\sigma_r^2 = 0.3853$;
		GPR	$\mu_r = f_{GP}, r^2 = 17.68$, $l = 7.093, \sigma_r^2 = 0.3872$.
Growth for non-reproducers	$Z' \sim Norm(\mu_n, \sigma_n^2)$	LM	$\mu_n = 1.284 + 0.528z$, $\sigma_n^2 = 0.3450$;
		GPR	$\mu_n = f_{GP}, r^2 = 8.117$, $l = 6.354, \sigma_n^2 = 0.3450$.

Table 4.1: Estimated demographic functions in GLMs/LMs and GPs/GPRs (regressions) for *C. Flava*. Here, z represents the log-transformed size; f_{gp} follows GP with Gaussian Kernel measuring input differences, $k_{Gau}(z_i, z_j) = r^2 \exp(-\frac{(z_i - z_j)^2}{2l^2})$ and zero mean functions. Recognising the common scenario of lost information about mother plants in real-world data, offspring size r_{size} and establishment probability p_{est} were modelled using a size-independent distribution, independent of whether a GLM or GP approach was used. We assumed r_{size} and p_{est} were known and fixed for all fitted IPMs to maintain a consistent baseline across models. A Gamma distribution ($\alpha = 0.706, \beta = 1.329$) was used, replacing the traditional Normal distribution, to fit the size of new recruits, r_{size}, due to the absence of negative sizes. The probability of offspring establishment ($p_{est} = 0.210$) was calculated by dividing the number of newborns by the total number of flowering stalks, ignoring seed loss processes.

4.3.2 Evaluation metrics

Our simulation analysis focused on evaluating the performance of the fitted IPMs in characterising the entire population dynamics. We calculated key population metrics based on *stable population growth theory* (Rees and Rose 2002; Ellner and Rees 2006), such as population growth rate (λ), stable state distribution ($\bar{\boldsymbol{\omega}}$), stable size distribution for reproductives ($\bar{\boldsymbol{\omega}}_{\boldsymbol{rep}}$) and relative reproductive value ($\boldsymbol{\gamma}$). These metrics are widely employed measurements and critical indicators of population

dynamics in biological research (e.g. Caswell 2000; Morris and Doak 2002; Rees, Childs, and Ellner 2014). To assess the performance of our models, we compared the population metrics from the discretized 'true' IPMs (generated from M_{gp} and M_{glm}) with those from the estimated IPMs derived from the three modeling methods.

The precision of λ was measured using Mean Squared Error (MSE), calculated as MSE = bias^2 + variance. $\bar{\boldsymbol{\omega}}$ and $\bar{\boldsymbol{\omega}}_{\boldsymbol{rep}}$, being probabilistic in nature, were assessed using the Kullback-Leibler (K-L) distance. For $\boldsymbol{\gamma}$, which reflects relative contributions, instead of assessing absolute values, we employed Cosine similarity to measure the angle between vectors. Table 4.2 summarizes the selected population dynamics measurements and their corresponding evaluation metrics.

	Formula	**Meaning**	**Distance metric**
λ	-	Asymptotic population growth rate	MSE
$\bar{\boldsymbol{\omega}}$	$\boldsymbol{\omega}/\sum_1^{N_m}\omega_i$	Asymptotic population stable state distribution	K-L distance
$\boldsymbol{\gamma}$	-	(Relatively) asymptotic reproductive value	Cosine similarities
$\bar{\boldsymbol{\omega}}_{\boldsymbol{rep}}$	$(\boldsymbol{\omega}\circ\boldsymbol{p_b})$ / $(\boldsymbol{\omega}\cdot\boldsymbol{p_b})$	Asymptotic stable distribution for reproductives	K-L distance

Table 4.2: For an IPM transition kernel $\mathcal{T}$ discretized and evaluated at mesh points $(m_1, ..., m_{N_m})$, we denote the resulting discretization matrix as $\mathcal{K}_{IPM}$. The matrix $\mathcal{K}_{IPM}$ can characterised by its dominant eigenvalue λ, alongside its dominant right eigenvector $\boldsymbol{\omega} = (\omega_1, ..., \omega_{N_m})^T$ and dominant left eigenvector $\boldsymbol{\gamma}$. Additionally, let $\boldsymbol{p_b}$ represent the vector of probabilities of reproduction evaluated at the mesh points. The symbols $\circ$ and $\cdot$ denote the Hadamard and dot product between two vectors, respectively. For a detailed mathematical explanation of the distance metrics employed, please refer to Appendix B.1.1.

4.3.3 Results

Predictive Performance

Figure 4.2 and Figure 4.3 provide model fitting results, with detailed numerical results available in Table B.3 and B.4 in Appendix B.1.2. In evaluating the predictive performance of the three model fitting methods on datasets D_{glm} and D_{gp}, the

ABC_GP method showed noticeable accuracy in capturing the true population dynamics. For both datasets, ABC_GP aligned its predictions closely with the actual growth rate λ, as indicated by the density histograms' peaks clustering around the actual values (Figure 4.2). In particular, for datasets D_{gp}, ABC_GP was the only method to provide a point estimate capturing a true λ value of less than 1, whereas other methods incorrectly suggested values greater than 1.

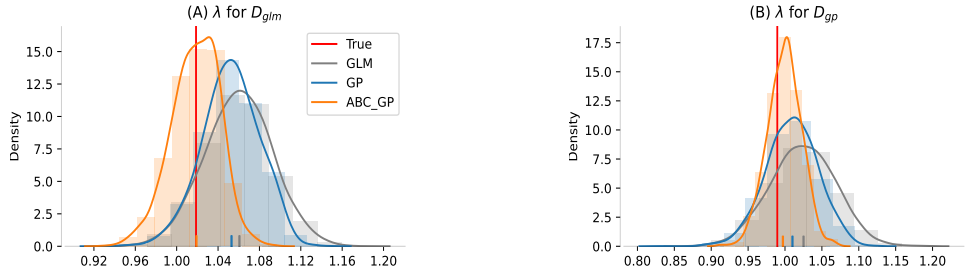


Figure 4.2: Predictions based on datasets D_{glm} and D_{gp}. Predicted λs, along with their corresponding density histograms (shown as shaded areas), were generated using 5,000 samples draw from the estimated posterior distribution of the three fitting methods. The red line represents the λ computed from the true discretized IPM. The pipe ('|') in the bottom of each plot marks the corresponding mean.

The ABC_GP method demonstrated reliability not only in predicting λ, but also in estimating the stable state distribution ($\bar{\boldsymbol{\omega}}$) and stable distribution for reproductives ($\bar{\boldsymbol{\omega}}_{\boldsymbol{rep}}$), with minimal mean K-L distance from the true distributions in both datasets (Figure 4.3). While ABC_GP did not reach the highest similarity scores in the reproductive value ($\boldsymbol{\gamma}$), its cosine similarity exceeded 0.99, confirming its strength in capturing population dynamics.

The GLM method demonstrated reasonable accuracy for dataset D_{glm}, particularly in estimating γ, reflected by the highest cosine similarity scores. However, a noticeable deviation was observed in estimating λ, along with inconsistencies in other demographic indicators. These limitations were more obvious in D_{gp}, suggesting that GLMs may struggle with the complexity embedded within such datasets.

It is worth noting that while the sample size of D_{glm} (562 individuals) is typical for real-world applications, it may not be large enough and remains subject to stochastic effects. To investigate this, a larger dataset D_{glm2} was generated, comprising information of 4,551 individuals. The newly estimated λ from D_{glm2}

demonstrated improved accuracy, closely approximating the true λ (Figure B.1 in Appendix B.1.2) and highlighting the critical role of sample size in GLMs.

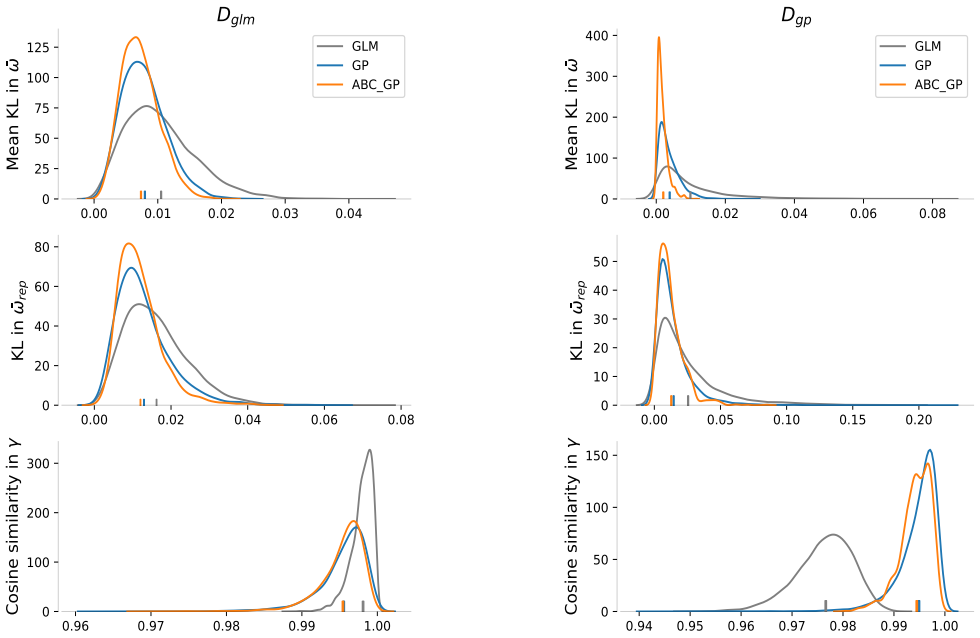


Figure 4.3: Predictive performance evaluations for the three model fitting methods applied to datasets D_{glm} (**left**) and D_{gp} (**right**). Different colours represent various settings of the growth model. The pipe ('|') in the bottom of each plot marks the corresponding mean.

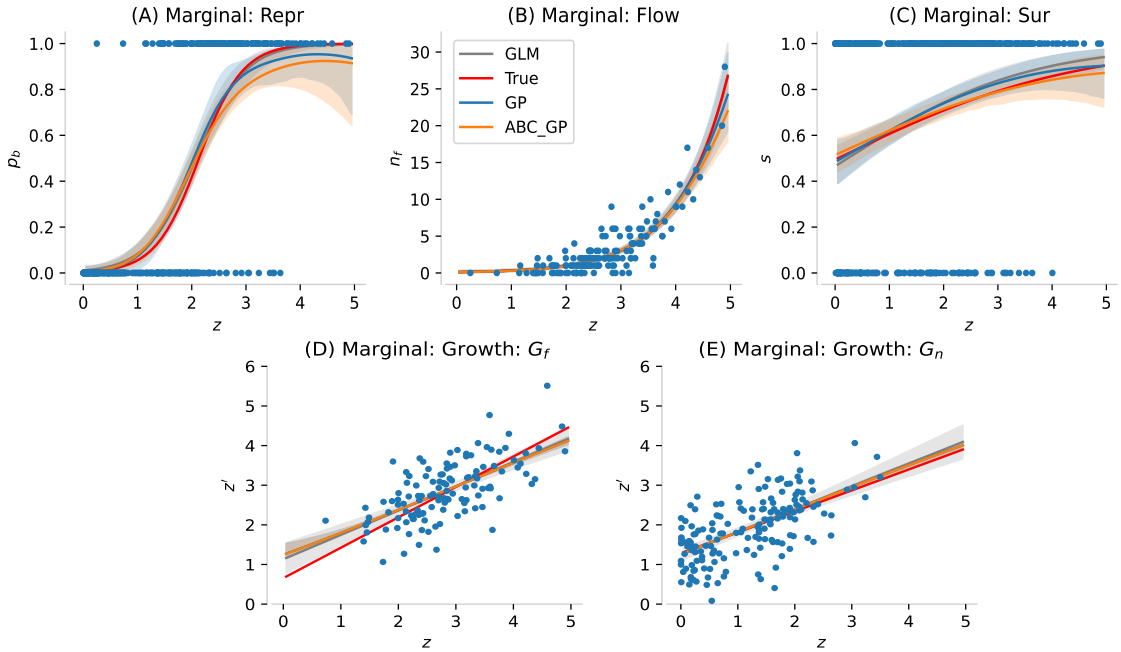


Figure 4.4: Model fitting results for D_{glm}. The true and estimated marginal distributions for the five vital rates, complete with their 95% confidence intervals in matching colours and the corresponding scatters data points in blue.

While ABC_GP generally performed well, it exhibited some deviations in marginal distributions for the vital rate (Figure 4.4). We will revisit this in the Discussion section (Section 4.5) to explain the cause of these deviations and provide guidance on their proper interpretations.

From Figure 4.2 Left, in D_{glm}, we saw that the GP and ABC_GP produced lower estimates of the population growth rate λ compared to the GLM. This difference may be partially explained by the marginal distribution shown in Figure 4.4 (A), where larger individuals were assigned lower reproduction rates in the GP and ABC_GP. To further explore the difference in predicted λ, we compared the pairwise joint behaviour among all vital rates (see Figure B.3 in Appendix B.2). In the main text, we focused on survival and reproduction rates, as they showed more visible differences across fitting methods in Figure 4.4.

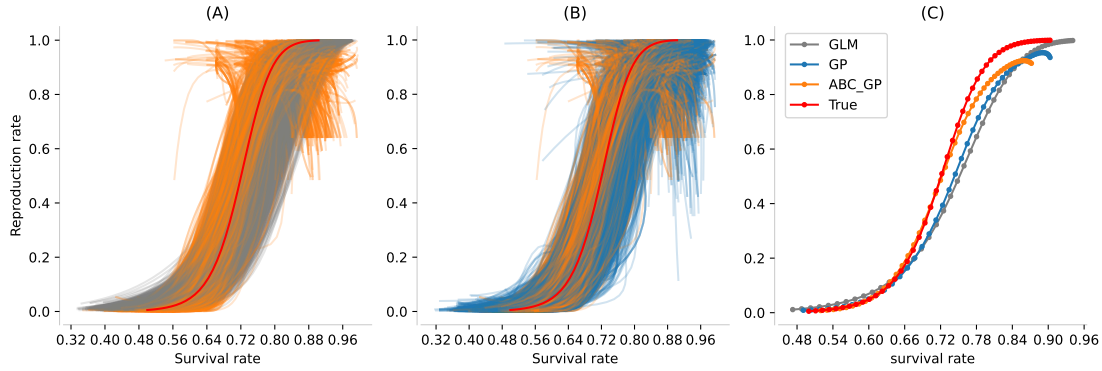


Figure 4.5: Using the 5,000 posterior samples that generated estimated λ in Figure 4.2 left, we extracted the corresponding survival and reproduction models and evaluated them on 50 mesh points (representing z from 0 to 5). The survival–reproduction pairs shown in (A) and (B) were evaluated jointly within each posterior sample at each mesh point, so the joint structure between vital rates was preserved. **(A)** Survival–reproduction pairs from GLM and ABC_GP (each line corresponds to a posterior sample evaluated over the mesh points). **(B)** Survival–reproduction pairs from GP and ABC-GP. **(C)** Sample means of survival and reproduction rates at each mesh point, shown for GLM, GP, ABC-GP, and the true model.

Figure 4.5 (A) compares the survival–reproduction rates across posterior samples from GLM and ABC_GP. Since GLM fitted each vital rate independently and deterministically, the implied relationship remained relatively consistent across samples. In contrast, ABC_GP captured a range of non-monotonic interactions

between survival and reproduction that were not present in GLM, especially in the high-value ranges. As survival increased, reproduction generally rose at first, but in higher survival regions, some samples showed reproduction decreasing, while others continued to increase. Similarly, at high reproduction levels, survival could either decrease or increase across samples.

Figure 4.5 (B) shows the same comparison between GP and ABC_GP. The ABC step introduced a noticeable shift in the joint patterns. For similar reproduction rates, ABC_GP placed more weight on lower survival probabilities. This may help explain why the ABC_GP-inferred λ is lower in Figure 4.2 left.

Figure 4.5 (C) compares the mean survival and reproduction rates at the 50 mesh points across posterior samples. While all models captured the general shapes of the survival and reproduction curves, notable differences appeared in certain regions. For example, both the GLM and GP began to diverge from the truth when survival probabilities exceeded approximately 0.65. In this range, for fixed levels of reproduction, both models tended to overestimate survival rates.

Introducing ABC methods to GP appears to adjust survival-reproduction interaction. However, around a survival probability of 0.72, the ABC_GP method also began to deviate from the truth. In this case, again, at fixed levels of reproduction, all models predicted higher survival than the true model.

Interestingly, although the ABC_GP model predicts higher survival in these regions, this does not lead to higher overall λ values compared to the true model (Figure 4.2 left). One possible reason may be that samples with both very high survival and very high reproduction are relatively rare in the ABC_GP outputs. As shown in the top-right corner of Figure 4.5(C), the average survival and reproduction rates tended to remain around 0.87 and 0.9, respectively. While the GP results allowed for such joint extremes, these combinations may be filtered out during the ABC step if they fail to reproduce the observed features — for instance, if simultaneous high survival and reproduction are not well supported by the summary statistics.

While such joint extremes are present in the true model, they may be under-represented in the observed data. In the dataset, only 12 out of 562 individuals had size $z \geqslant 4$. This imbalance is also visually reflected in Figure 4.4 (D), where only a few reproducers fall in $z \geqslant 4$. Together with the positive association between size and survival in Figure 4.4 (C), this suggests that the limited presence of such large-sized reproducers may make it difficult for the models to learn or retain these joint extremes in survival-reproduction.

Efficiency of the enhanced ABC-PMC

Table 4.3 provides a detailed summary of algorithm efficiency for the developed ABC-PMC. As expected, with the modified sequence of quantiles $\{\alpha_i\}_{0\leqslant i\leqslant 10}$, the acceptance threshold δ_i gradually decreases over the entire simulation process. Since computations were executed in parallel, the actual number of accepted particles (N_c) sometimes slightly exceeded the proposed value N_c^{proposed}. To maintain consistency with Algorithm 3, we use N_c to denote the true number of accepted particles.

Proposal Space Exploration

A key observation is that the number of unique accepted particles remains close to the total number of accepted particles ($N_c^{\text{unique}} \approx N_c$ for $c = 0, ..., 10$). This may suggest that the algorithm does not frequently propose the same particles, avoiding redundant sampling. Other observations from Table 4.3 further support this interpretation

In our ABC-PMC framework, the proposal space at iteration c depends on the set of accepted particles from the previous iteration. Each accepted particle can generate up to $5M$ possible states for the next iteration. However, not all accepted particles contribute equally to exploration in the next iteration. Some particles tend to generate highly similar or even identical proposals, due to the employed perturbation kernel that preserve local structure. To describe this, we define N_{c-1}^{neibo} as the number of accepted particles that can transition to states that are already well-covered in the proposal space. These particles do not add diversity

	$c=0$	$c=1$	$c=2$	$c=3$	$c=4$
α_{c-1}	1	0.50	0.50	0.45	0.40
δ_{c-1}	∞	10.82	9.70	9.00	8.41
N_c^{proposed}	1,000,000	500,000	300,000	150,000	100,000
N_c	1,000,060	501,875	300,176	150,650	100,435
N_c^{unique}	1,000,060	501,869	300,173	150,650	100,435
$n(\mathcal{X}_c)$	5000^5	$> 2.5 \times 10^{10}$	$> 1.1 \times 10^{10}$	$> 6.5 \times 10^9$	$> 3.3 \times 10^9$
$n(\mathcal{X}_c)^{lb}_{ESS}$	5000^5	$> 2.5 \times 10^{10}$	$> 1.1 \times 10^{10}$	$> 6.5 \times 10^9$	$> 3.2 \times 10^9$
N_c^{trials}	1,000,060	1,000,000	1,070,000	1,040,000	1,500,000
P_c^{accep}	1	0.5019	0.2805	0.1449	0.0670
P_c^{prior}	1	0.4999	0.2502	0.1101	0.0422
t_c	53.15	75.17	77.78	74.55	107.97

	$c=5$	$c=6$	$c=7$	$c=8$	$c=9$
α_{c-1}	0.35	0.30	0.20	0.15	0.10
δ_{c-1}	7.92	7.38	6.78	6.19	5.56
N_c^{proposed}	60,000	30,000	15,000	7,000	1,000
N_c	60,040	30,036	15,026	7,003	1,000
N_c^{unique}	60,039	30,036	15,026	7,002	1,000
$n(\mathcal{X}_c)$	$> 2.1 \times 10^9$	$> 1.2 \times 10^9$	$> 6.4 \times 10^8$	$> 3.2 \times 10^8$	$> 1.5 \times 10^8$
$n(\mathcal{X}_c)^{lb}_{ESS}$	$> 2.0 \times 10^9$	$> 1.2 \times 10^9$	$> 6.0 \times 10^8$	$> 2.9 \times 10^8$	$> 1.3 \times 10^8$
N_c^{trials}	2,190,000	3,120,000	6,370,000	16,740,000	19,980,000
P_c^{accep}	0.0274	0.0096	0.0024	4.183×10^{-4}	5.005×10^{-5}
P_c^{prior}	0.0139	0.0038	0.0007	8.600×10^{-5}	8.300×10^{-6}
t_c	158.12	225.1	453.22	1182.57	1433.43

Table 4.3: Results of applying ABC-PMC to IPM in a simulated dataset. **Notations** : $\boldsymbol{\delta_{c-1}}$: the quantile-based acceptance threshold used at step c, which was determined based on $\boldsymbol{\alpha_{c-1}}$ quantile at step $c-1$ (so denoted by δ_{c-1}). δ_{-1} and α_{-1} are for the initialisation. $\mathbf{N_c^{\text{proposed}}}$: the proposed number of accepted particles required for the simulation step c. $\mathbf{N_c}$: the true number of accepted particles for the simulation step c. $\mathbf{N_c^{\text{unique}}}$: the number of unique particles among N_c acceptances. $\mathbf{n(\mathcal{X}_c)}$: the total number of possible states of $\boldsymbol{\theta}$ at simulation step c. Apart from $\mathbf{n(\mathcal{X}_0)}$, the others were estimated through $n(\mathcal{X}_c)^{lb}$. $\mathbf{n(\mathcal{X}_c)^{lb}_{ESS}}$: a lower bound for $n(\mathcal{X}_c)$ but estimated based of ESS. $\mathbf{N_c^{trials}}$: total number of trials to achieve the acceptance. $\mathbf{P_c^{\text{accep}} = N_c/N_c^{trials}}$: the acceptance rate for step c. $\mathbf{P_c^{\text{prior}}}$: the acceptance rate for a single-iterated ABC-PMC, where the samples were drawn directly from the priors made up by MCMC samples. To improve reliability with P_c^{accep}, P_c^{prior}s were computed based on 10,000,290 particles drawn from $\pi(\boldsymbol{\theta})$. To ensure comparability, P_c^{prior}s must be computed under the exactly same setting that produced P_c^{accep}s. That is, for each c in $\{0, 1, ..., C\}$, the 10,000,290 particles were normalised with MADs $\boldsymbol{\sigma}_{c-1}$ first, the threshold δ_{c-1} then employed to compute P_c^{prior}. It took approximately 530.5 minutes to complete this single-iterated algorithm with 10,000,290 particles. $\mathbf{t_c}$: the total amount of time, measured in minutes, used to complete the cth iteration.

to the new proposals. To approximate the number of genuinely new proposal states, $n(\mathcal{X}_c)$, we estimate a lower bound for it as:

$$n(\mathcal{X}_c) > n(\mathcal{X}_c)^{lb} = 5M \times (N_{c-1}^{\text{unique}} - N_{c-1}^{\text{neibo}}),$$

where N_{c-1}^{neibo} is estimated by the number of accepted particles within N_{c-1}^{unique} unique particles that share identical states in $V-1$ out of V sub-models with another accepted unique particle.

Additionally, since ABC-PMC operates as an iterative ABC importance sampling, another lower bound can be roughly estimated using the effective sample size (ESS):

$$n(\mathcal{X}_c) > n(\mathcal{X}_c)_{ESS}^{lb} = 5M \times (ESS_{c-1} - N_{c-1}^{\text{neibo}}),$$

with $ESS_c = 1/\sum_{i=1}^{N_c}(w_c^i)^2$.

From Table 4.3, we observe that the total number of trials required to obtain the desired number of acceptances is much smaller than the estimated size of the proposal space, i.e., $N_c^{trials} \ll n(\mathcal{X}_c)_{ESS}^{lb} < n(\mathcal{X}_c)^{lb} < n(\mathcal{X}_c)$ for $c = 0, ..., 10$. Together with the previous observation that $N_c^{\text{unique}} \approx N_c$ for $c = 0, ..., 10$, there is not much space left for the algorithm to repeatedly propose the same particles (which lie in low posterior probability regions) multiple times and yield only a few acceptances among the repeated proposals. Most particles are likely to be accepted, as they have higher chance of meeting the conditions.

Proposal Quality Assessment

If the distribution of the 'universal' distance remained unchanged throughout the ABC-PMC model fitting process, then running ABC-PMC with a sequence of decreasing acceptance thresholds $\{\delta_c\}_{0 \leqslant c \leqslant C}$ is nothing better than conducting a single iteration with the strictest threshold δ_C. To evaluate the effectiveness of the proposed ABC-PMC, we examine its ability to improve proposal quality over iterations.

To check if there is an improvement, we compare the true acceptance rate P_c^{accp} with P_c^{prior}, the acceptance rate of a single-iteration ABC-PMC using threshold δ_{c-1}. To ensure comparability, P_c^{prior} is computed under exactly the same setting

used to generate P_c^{accp} (see details in the caption of Table 4.3). In this case, if the proposals offer no advantage over the prior, then we expect $P_c^{\text{prior}} \approx P_c^{accp}$ or even $P_c^{\text{prior}} > P_c^{accp}$.

The results from Table 4.3 and Figure 4.6 (Left) indicate that the proposal distribution, q_c evolved gradually to be much more efficient throughout the ABC-PMC. Initially, the ratio $P_c^{accp}/P_c^{\text{prior}}$ is close to 1, but it rapidly increases as the iterations proceed. By the end of the simulation, the true acceptance rate, P_c^{accp}, exceeds P_c^{prior} by a factor of more than 6.

However, a higher acceptance rate in the proposal distribution does not necessarily imply that the ABC-PMC is more efficient. Before we get to the final round of proposals, we have incurred the computational cost of all the previous iterations that refined the distribution, so we need to sum up the time spent on the entire implementation.

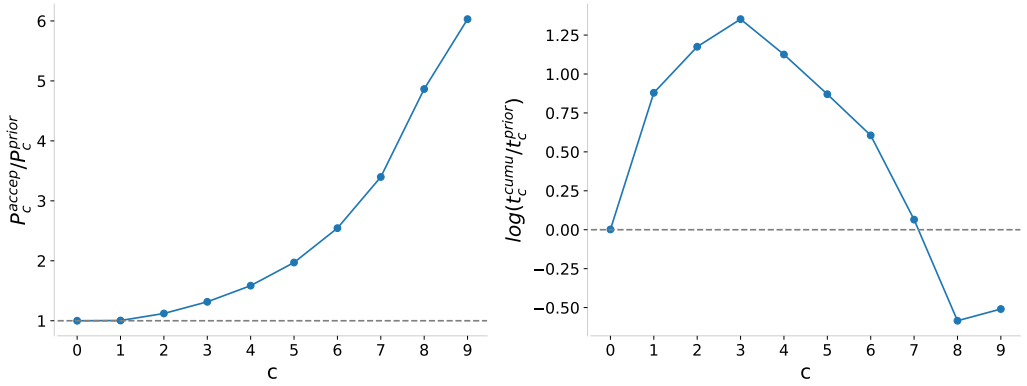


Figure 4.6: Left: The ratio of P_c^{accep} and P_c^{prior} over simulation time c. The grey horizontal line cross the point representing $P_c^{accep}/P_c^{prior} = 1$. **Right:** The logarithmic ratio between the cumulative time spent on ABC-PMC, represented as t_c^{cumu}, and the expected implementation time, estimated as t_c^{prior}. The t_c^{cumu} is obtained by adding up individual times for each single iteration t_c until the current iteration c, $t_c^{cumu} = \sum_{i=0}^{c} t_i$. Instead of timing in real practices, t_c^{prior} was estimated using the formula $t_c^{prior} = (N_c/(P_c^{prior} * 10000290)) * 530.5$ based on the 10,000,290 particles (see Table 4.3).

This total cost is displayed on the right-hand side of Figure 4.6, plotting the log ratio between the cumulative runtime for ABC-PMC, t_c^{cumu}, and the estimated runtime for a single iteration, t_c^{prior}, where the samples are drawn directly from the priors under the identical acceptance thresholds. Although the particles generated

by ABC-PMC always exhibit higher acceptance probabilities, obtaining the same number of acceptances can initially require much more computational time. For example, at $c = 3$, ABC-PMC takes nearly four times longer than a single-iteration approach using MCMC priors (as $t_3^{\text{cumu}}/t_3^{\text{prior}} = \exp(1.352) \approx 3.8666$). However, after $c = 7$, the proposed ABC-PMC becomes increasingly efficient. By the final iteration, the total runtime of ABC-PMC is only around 60% of the estimated runtime required for the single-iterated algorithm using just MCMC priors ($t_9^{\text{cumu}}/t_9^{\text{prior}} = \exp(-0.5092) \approx 0.6010$ times).

Trends in Summary Statistics

Figure 4.7 illustrates the variations in the summary statistics throughout the ABC-PMC process. Looking at the orange lines, we see that the means (dashed) and medians (solid) of the 5 selected summary statistics gradually decrease as the acceptance threshold δ_c becomes more restrictive. This observation indicates the effectiveness of the adaptive weighting strategy, as the imposed barrier on the 'universal distance' can validly constrain each of the 5 summary statistics. Since the orange lines only reflect accepted particles, their decreasing pattern is expected, aligning with the setup of $\{\alpha_c\}$.

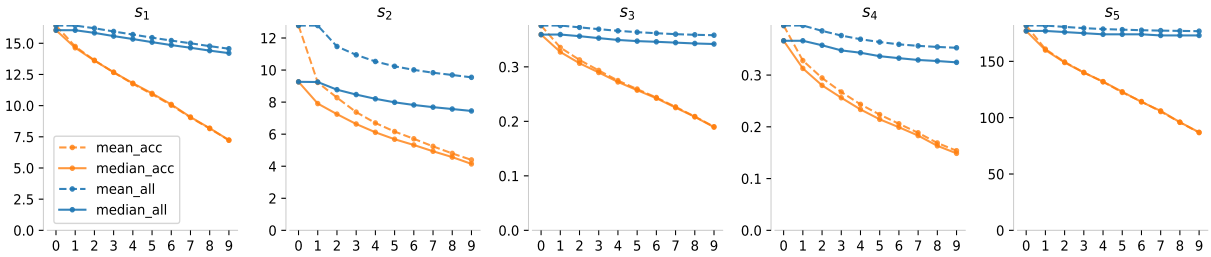


Figure 4.7: The mean (dashed line) and median (solid line) for the 5 selected summary statistics, s_1 to s_5, over the ABC-PMC simulation time c ($x-$axis). The orange lines describing the information of summary statistics for accepted particles only, while the blues stand for all particles including those rejected.

Conversely, the blue lines reveal a different perspective. They represent the summary statistics for all proposed particles (before any decisions to accept or reject are made). These particles can reflect the distance between the posterior $\pi(\boldsymbol{\theta}|y_{obs})$

and the current proposal q_c. If Figure 4.6 is not clear in demonstrating that the proposals q_c are continually enhanced throughout the algorithm, the descending trend of the blue lines in Figure 4.7 offers direct evidence of this improvement.

Given this understanding, it is not surprising to see in Figure 4.8 that, the predicted distribution of population growth rate (λ) was corrected step by step through a sequence of slowly varying intermediate distributions in each ABC-PMC iteration — even though we do not directly target λ.

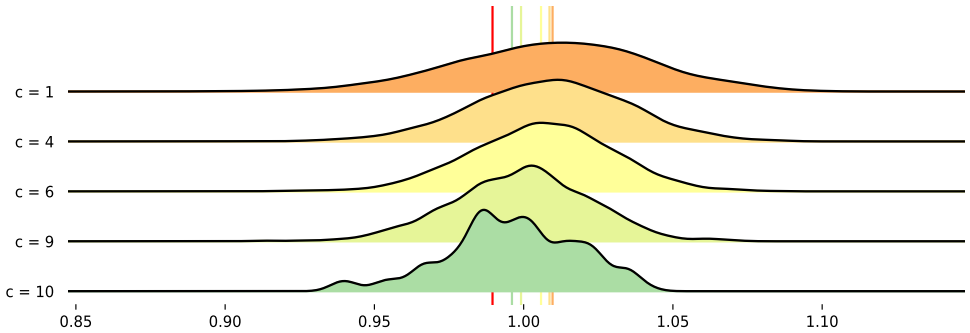


Figure 4.8: Predicted distribution of λ for each ABC-PMC iteration. Vertical lines indicate the true population growth rate, λ, (in red) and corresponding predicted mean for each iteration. Note that, at step $c = 10$, where we applied the most restrict threshold, leading to a extreme high rejection rate and a limited yield of only 120 samples within a feasible time frame. Due to these constraints, the results from this step were excluded from our evaluations of predictive performance and algorithm efficiency. We present the findings about $c = 10$ here to illustrate the potential of the proposed method for achieving even better results.

4.4 Application of the enhanced IPM with real datasets

Building on the insights from the previous simulated analyses, we applied the ABC_GP IPM to *C. flava* using real datasets. According to the background knowledge in Section 4.2, we implemented distinct growth models for reproducers and non-reproducers, and the state variable in the IPMs were considered as 2-dimensional, standing for size and age.

Additionally, to better model population dynamics in interaction with the natural system in the real case study, we integrated environmental factors from a nearby

weather station (USW00094030, NOAA 2013). Specifically, we incorporated three annual weather variables to summarise inter-annual environmental variation: the mean daily maximum temperature, the mean daily minimum temperature, and the mean daily precipitation, each averaged across the year (D. R. Easterling et al. 2000; Zou et al. 2009; Huxman et al. 2004). These variables were included as covariates in our vital rate models, following common practice in IPMs involving environmental stochasticity (Dahlgren, Bengtsson, et al. 2016; Ellner, Childs, et al. 2016; Félix-Burruel et al. 2024). Model training used data from 2003 to 2008, with the last four years reserved for testing. Implementation details were provided in Appendix B.2.

We evaluated predictive performance by simulating populations using IBMs based on three methods, focusing on both long-term and 1-step forecasting of demographic trends. Predictive accuracy was assessed through point estimates and 95% prediction intervals, compared to actual observations. Note that, according to the National Oceanic and Atmospheric Administration (NOAA 2013), the year 2012 was the warmest year in the historical record from 1895 to 2012 for the continental United States. We would not expect the proposed methods to effectively handle such extreme weather conditions.

4.4.1 Results

Algorithm efficiency

A similar ABC-PMC efficiency exploration in this real case study can be found in Appendix B.3. The efficiency of the ABC-PMC is more pronounced now. The summary statistics for all particles are keeping reduced during iterations; meanwhile, by the end, the true acceptance rate (P_c^{accp}) exceeds P_c^{prior} by over 50 times, and the total time for ABC-PMC amounted to only about 9% of the estimated implementation time for the single-iterated algorithm using just MCMC priors ($t_{10}^{cumu}/t_{10}^{prior} = \exp(-2.379) \approx 0.0926$ times).

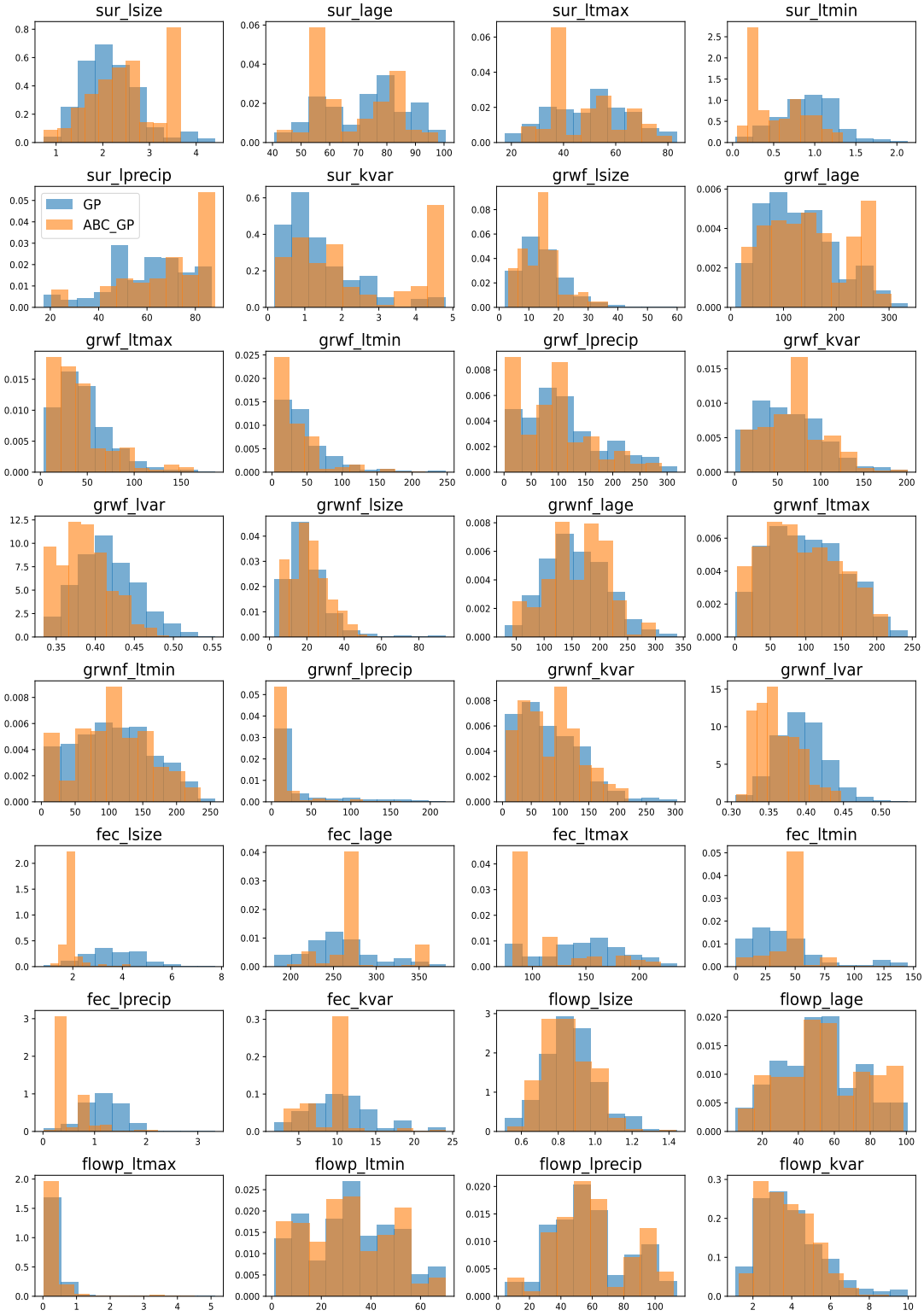


Figure 4.9: Density histograms of estimated marginal posterior distributions for all hyperparameters under methods GP and ABC_GP.

Predictive Performance

Figure 4.9 shows marginal posterior approximations for the hyper-parameters of five vital rate for ABC_GP. The hyper-parameters are subject to varying degrees of restrictions based on the chosen summary statistics. For the probability of reproduction (**fec**), all of the six hyper-parameters appeared concentrated around a single mode, contrasting with their flat MCMC priors. However, for the Poisson model regarding the number of flowering stalks (**flowp**), the hyper-parameters appeared to be less distinguishable, with no noticeable difference between the MCMC and ABC_GP posterior samples in the marginal histograms. Other vital rates also show modifications in MCMC samples due to ABC constraints. The additional information provided by the ABC summary statistics appeared to assign higher weights to certain regions previously having low posterior probability in the MCMC samples, which made the corresponding ABC posterior approximations to be multi-mode, such as the kernel variance for the survival model (**sur_kvar**) and the non-breeders' growth model (**grwnf_kvar**), as well as the lengthscale for size and age in the survival model (**sur_lsize**, **sur_lage**) and the lengthscale for precipitation in the breeders' growth model (**grwf_lprecip**).

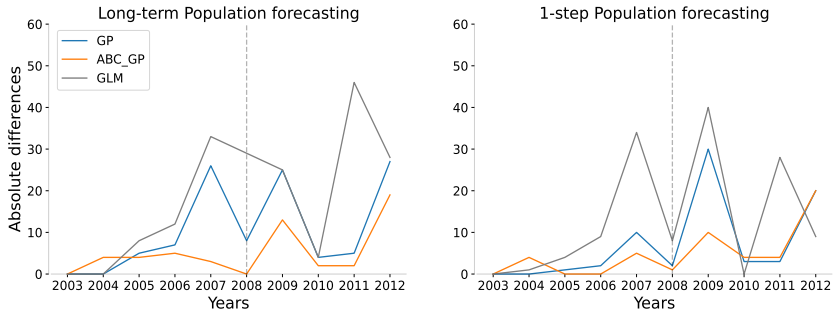


Figure 4.10: Absolute differences between the observed and predicted population size (medians of the predictions). The models were trained using five datasets from years 2003-2004 to 2007-2008 (before the grey dashed vertical line). For **long-term** forecasting, the model used the initial population structure from Year 2003 as the starting point and generated predictions based on simulated populations from the previous year. For example, the predicted population size for 2005 was based on the simulated population for 2004 (which was generated using the true observed population for 2003). For **1-step** forecasting, the model used the true observed population structure from the previous year to generate predictions. That is, for example, to make a prediction for 2005, the true structure for 2004 was plugged into the IBM.

The impact of ABC restrictions on the marginal posteriors (Figure 4.9) becomes more evident when integrating individual vital rates into the joint population model (Figure 4.10 and 4.11). Median values were used as point estimates to reduce outlier impacts in IBM simulations. A side-by-side comparison with GLM and GP methods revealed that, GLM-based models tended to produce larger absolute errors in predicted population size (see Figure 4.10). In certain cases, such as the 1-step forecast in Figure 4.11, GLM predictions showed notable deviations, with trends often contradictory to actual observations. While GP methods also showed some deviations (particularly for 2009), switching to ABC_GP led to improvements in accuracy for both long-term and 1-step forecasts.

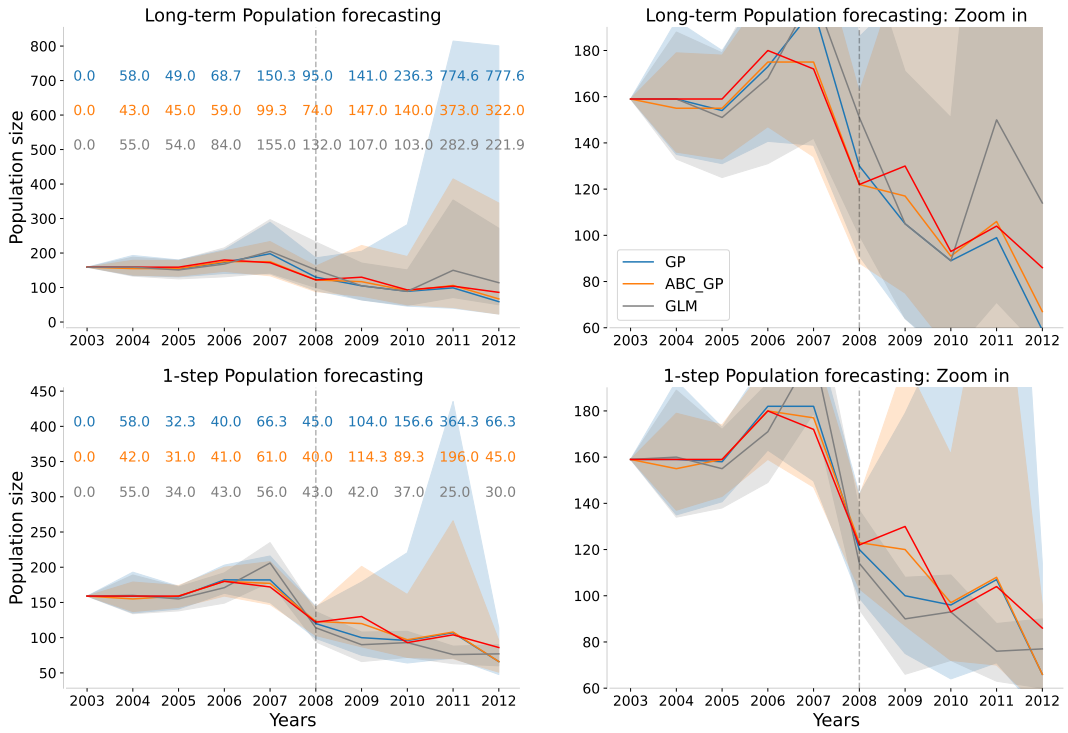


Figure 4.11: Two types of population size predictions, along with their corresponding 95% PI (shaded areas), were generated using GP, ABC_GP and GLM samples through 5,000 IBM simulations. The blue predictions were based on MCMC samples, while the orange and grey ones were based on ABC_GP and GLM posterior samples, respectively. The widths of the corresponding 95% PI are indicated by their respective colours on the left-hand side of the figure. The point estimations are represented by solid lines in their corresponding colours. The red line represents the true observed population size.

For long-term population forecasting, prediction intervals (PIs) of all three model types remain narrow during training years and widen in subsequent periods,

generally covering the observed population sizes. However, GP models exhibit broader PIs compared to the other models.

More details can be illustrated by PIs for 1-step predictions, which reduce the influence of cumulative demographic randomness present in long-term forecasts. GP models show a similar pattern as in long-term predictions, but the width of the PI generated by GLMs remains consistently narrow, regardless of whether applied to training or testing datasets.

Although GLMs exhibited narrow PIs, their reliability appears limited. As detailed in Figure 4.12, focusing on the annual population growth rate, we observed that the actual growth rates frequently fell outside the GLM's 95% PIs, indicating a mismatch even for the training year 2006-2007. In contrast, the ABC_GP method not only covers the true growth rates but also adjusts its estimates to align with observed changes. Among the three models considered, ABC_GP provides more consistent point estimates, while maintaining reasonably narrower PIs that cover the observations.

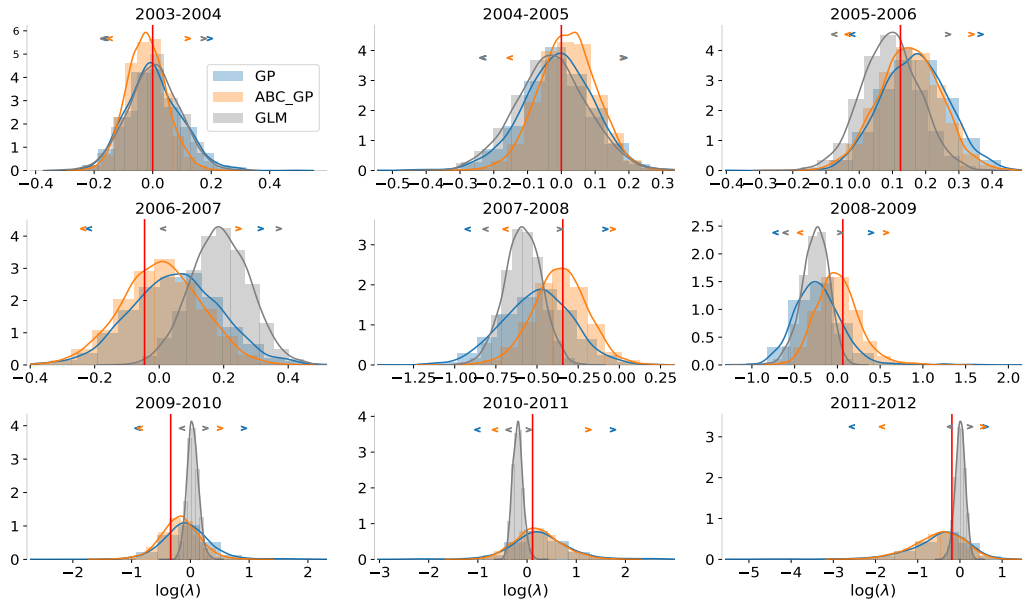


Figure 4.12: Histograms represent the density of logarithmic simulated population growth rates generated through **1-step** forecasting. The associated 95% prediction intervals are denoted by angle brackets, each in its corresponding colour, positioned at the top of each histogram. The red vertical line represents the true observed population growth rate.

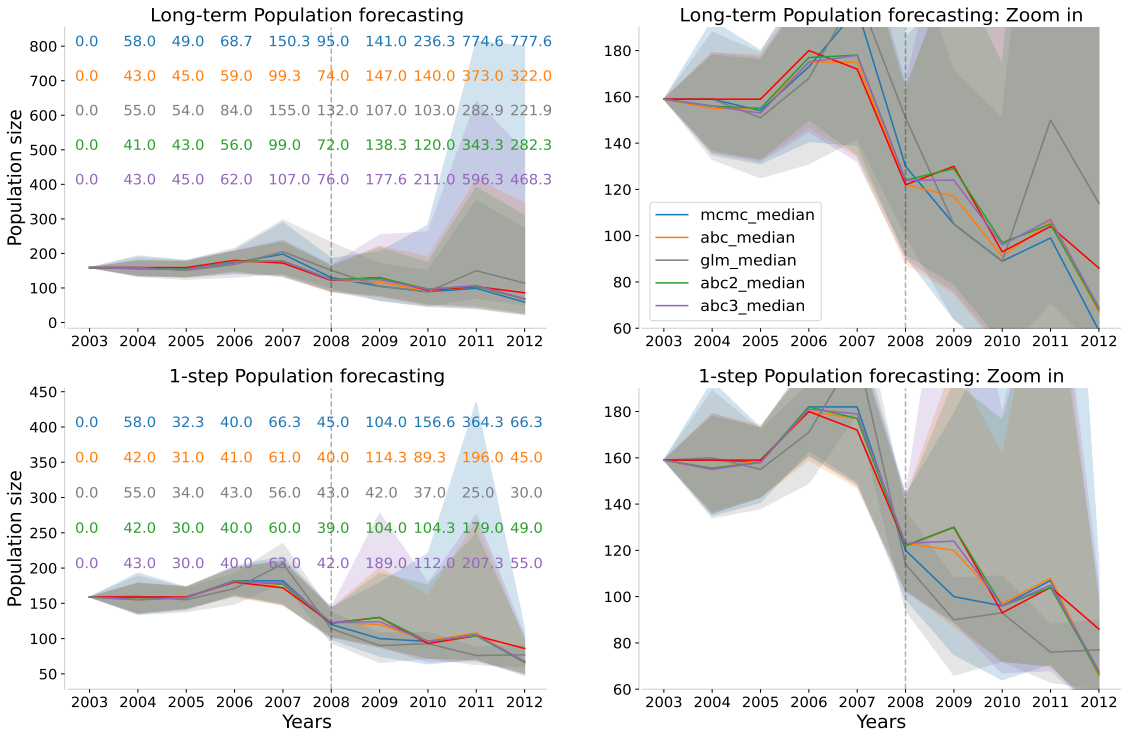


Figure 4.13: Exactly the same setting as Figure 4.11, but additional simulations generated using two other sets of ABC posterior samples are depicted here. The new simulations are visually represented in green and purple.

Appendix B.3.1 presents the estimated marginal posteriors for vital rates under the three modelling methods. The goal of this analysis is not to interpret biological mechanisms, but to illustrate how the method behaves when applied to real data. These help to explore differences in prediction across models — why GLM predictions differ from those of GP and ABC_GP, and how ABC_GP further narrows posterior uncertainty relative to GP through population-level constraints. The analysis in Appendix B.3.1 offer similar insights to those from the simulation study (Figure 4.5).

The stochastic nature of the ABC algorithm, influenced by sampling randomness and adaptive threshold adjustments, leads to variability in posterior samples and corresponding PI, which can be narrower or wider. Figure 4.13 additionally presents forecasts from other two ABC-PMC implementations, illustrating varied uncertainties (shown as shadow areas) but consistently accurate point estimates (solid lines). Variations in PI generated by different replications of our proposed ABC-PMC algorithm could raise concerns about the convergence of ABC posterior

samples. These concerns will be discussed in detail in Section 4.5.

4.5 Discussion

Why is the overall performance Good, but the marginal distribution Poor?

The adoption of ABC method shifts our focus from solely maximising likelihood to, additionally, minimising the discrepancy between model predictions and observed population data, using carefully selected summary statistics. This dual focus, while broadening the scope of analysis, introduces inherent biases, particularly when using insufficient summary statistics or non-zero thresholds (which is common in practical applications). Additionally, stochasticity at the observed population might bring additional biases. In our study of simulated datasets, these biases can be observed when deconstructing the comprehensive population model into individual vital rates (Figure 4.4). However, this does not necessarily imply that ABC performs poorly in higher dimensions (population-level) compared to traditional approaches (Figure 4.2)).

In IPMs, interactions among rates are inevitable, even when modeling each vital rate independently using simpler methods like GLMs, as demonstrated in our simulation case study. For instance, all vital rates are modelled as depending on the same state variables (Fung et al. 2022). The ABC framework, by accounting for these interdependencies, potentially offers a more nuanced representation of population dynamics. This suggests that while ABC might introduce biases at the level of individual vital rate predictions, its ability to incorporate rate interactions enhances its effectiveness in capturing broader population dynamics.

A natural question then arises: if our primary interest lies in individual vital rates, would not such biased marginal results be misleading? The answer fundamentally depends on the research objectives. ABC shifts the focus to the interactions among various vital rates, by adding layers of complexity at the population level. Consequently, using a model calibrated for population-level insights to infer details about marginal distributions might not yield meaningful results in this context.

On one hand, not all population-level insights may be relevant to that specific target vital rate. On the other hand, in our simulations, where life cycle components are clearly defined, we are able to accurately project the high-dimensional population model onto the 'correct' dimensions for individual vital rates. However, this level of clarity is often lacking in real-world situations, where life cycle components might be obscured or unknown. In such scenarios, projecting a complex ecological model onto dimensions that do not align with the true life cycle result in less-meaningful interpretations. This situation is akin to viewing a cylindrical object from different angles, where each angle offers a distinct 2D projection, leading to varying interpretations of the original 3D object. Therefore, again, it is crucial to tailor the methodological approach to the specific research objectives: traditional methods might suffice for analysing individual vital rates marginally, but they may fail to capture the complex interplay necessary to accurately reflect true population dynamics.

If ABC results converged?

Figure 4.13 could raise concerns about the convergence of our ABC posterior samples. If the processes for obtaining the three sets of ABC posterior samples differ only by being three separate replications, does the variability in their PI suggest that the results from our proposed ABC-PMC algorithm have not yet converged? The answer is: 'Yes, they might indicate non-convergence. It would be nearly impossible to attain perfectly converged results through the proposed ABC-PMC.' The most immediate reason for this can be that the threshold for acceptance in our ABC-PMC can never go to zero in real practice.

On the one hand, we face limitations due to a lack of sufficient computational resources. In the three repetitions of ABC-PMC implementations, the thresholds employed in the final iteration are already quite low, approximately 9.104, 9.135, and 9.148, but still far from zero. The acceptance rates hover at approximately $P_C^{accep} \approx 2.043 \times 10^{-5}$, with a total implementation time of 11761.89 minutes, approximately 8 days (see Table 4.3). These constraints prevent us from enforcing

stricter ABC acceptance conditions while keeping the total implementation time reasonably manageable. Notice that these threshold (9.104/9.135/9.148) are not directly comparable, as difference values in MAD were employed for normalising.

On the other hand, the prior and perturbation kernel used in ABC-PMC also pose challenges to reaching a very small threshold. The prior is quite restrictive, assuming the true ABC target must be contained within a finite, discretized set constructed by MCMC samples, while the set might not include samples that satisfy the desired small threshold. The perturbation kernel, being conservative, allows only one vital rate to be perturbed at a time, which can inevitably eliminate potential good samples in the early stages of ABC-PMC. As the space of posterior samples for the final iteration depends on those from earlier iterations, the sample space for the final iteration is merely a subset of the finite, discretized set (constructed by MCMC samples).

However, the final posterior samples we obtained do not appear to be overly concerning. Looking at the estimated marginal posterior distributions (Figure B.6 in Appendix B.3), there are some mode shifts in `sur_ltmax` and `sur_ltmin`, but for the most of other hyperparameters, results generated from the three repetitions seem to be in agreement. In addition, the point estimated population sizes from the three ABC repetitions are quite stable and highly overlap each other (Figure 4.13). These might be the best outcomes we can achieve with the current ABC-PMC algorithm. With limited computational resources being an objective constraint we cannot overcome, but the current ABC-PMC sampler could potentially be further developed with a less restrictive prior, enabling an exploration of the entire parametric space (Section 6.2.2).

5

A Python package: ABC_GP_IPM

Contents

5.1 Object-oriented programming in Python 86
5.2 Case study 2: *Ovis aries* 87
5.2.1 The study site 87
5.2.2 Interplay between Weather, Population density, and Vegetation resilience 88
5.2.3 Data collection and the Study dataset 89
5.2.4 Data analysis and Model settings 90
5.3 Getting start with ABC_GP_IPM 93
5.3.1 Initialisation 93
5.3.2 SS selection 94
5.3.3 Cache 96
5.3.4 ABC sampling 97
5.3.5 Forecasting 99
5.4 Case study 2: Results 100
5.5 Case study 2: Discussion and Future works 104
5.6 Discussion 104

Population dynamics research frequently confronts the challenge of modeling complex biological systems with inherent variability and uncertainty. Our previous work improves IPM by integrating GP models via a custom ABC sampler, enhancing their adaptability and the ability to reflect vital rate interactions (Chapter 4). However, the practical implementation of this innovative approach is non-trivial, requiring considerable computational expertise, which can be a barrier for researchers

without a strong programming background.

To address this, we developed a dedicated `Python` (Van Rossum and Drake Jr 1995) package, `ABC_GP_IPM`, simplifying the execution of our proposed ABC_GP IPM. This tool integrate the selection of ABC summary statistics (Section 3.3) and the ABC sampling process (Section 3.4), providing a user-friendly interface that allows researchers to concentrate on their biological inquiries rather than the complexities of computational implementation.

A key feature of our package is its respect for the highly context-specific nature of demographic studies. It allows users to customise functions, like those for IBM simulations, according to the actual life cycle of the species under study. This customisation is not confined to fixed forms but requires adherence to certain rules defined within the package, including specific conventions for naming, input, and output formats. Such structured flexibility ensures that, while the package is adaptable to various ecological contexts, it maintains a consistent framework that facilitates reliable and reproducible research.

The decision to develop this package in `Python` was intentional, primarily to leverage the capabilities of `GPflow` (Matthews et al. 2017) — an advanced library for building GP models. `GPflow` provides a powerful and flexible framework for specifying and solving GP models with cutting-edge techniques, including integrating of `TensorFlow` (Martín Abadi et al. 2015) for faster computation and GPU acceleration. By developing in `Python`, our package also benefits from the extensive scientific computing ecosystem available in `Python`, which includes libraries such as `NumPy`, `SciPy`, and `matplotlib` for numerical computations and data visualisation (Harris et al. 2020; Virtanen et al. 2020; Hunter 2007). Furthermore, Python's widely use in data science and machine learning makes it accessible to a broader audience, potentially facilitating interdisciplinary research.

Our package has targeted functionality, specifically designed to support the initial stages of demographic modeling (see the workflow in Figure 5.1). This includes a systematic workflow to select ABC summary statistics (Section 3.3) and the implementation of the tailored ABC sampling process (Section 3.4). While our tool

provides robust support for these critical preliminary steps, it deliberately does not include the construction of IPM matrices (Step 5 in Figure 5.1). This domain of IPMs is already well-serviced by a number of robust and mature packages, particularly within the `R` ecosystem (R Core Team 2016), such as `IPMpack` (C. J. E. Metcalf et al. 2013, `ipmr` (Levin et al. 2021). Recognising this, we have chosen not to replicate these functionalities but instead focus on enhancing the efficiency of the stages leading up to IPM construction. For researchers needing to construct IPM matrices, our case studies includes `Python` code examples. For those wishing to utilise existing R-based IPM functions or tools directly, we recommend exploring well-established interoperability tools such as `rpy2`. These tools support executing R code within `Python`, offering a easy integration for users who wish to combine the strengths of both programming environments.

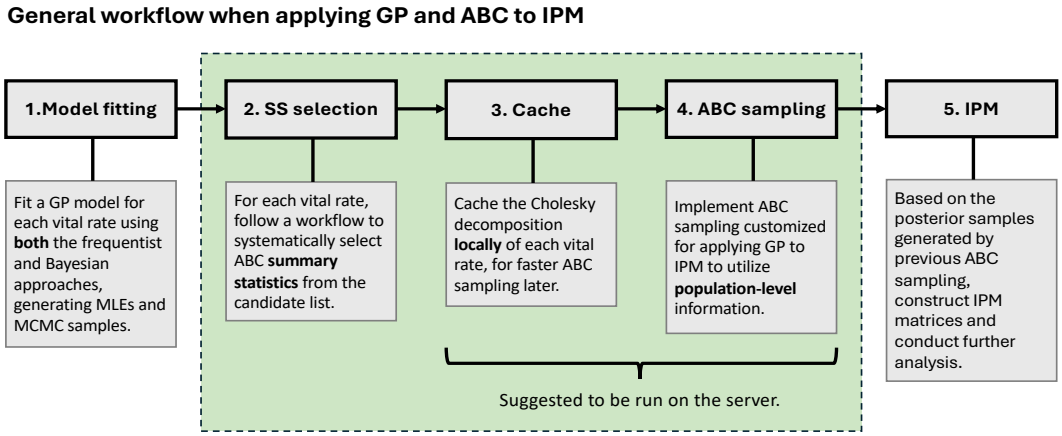


Figure 5.1: A general workflow when applying GP and ABC to IPM. The processes within the green box have been encoded in our package.

This chapter starts by an introduction to the foundational background of our study, including the use of object-oriented programming in `Python` (Section 5.1) and details about the study species (Section 5.2). We then demonstrate the utility and performance of our package through a case study involving real-world biological data (Section 5.3), showcasing how our tool simplifies complex modeling processes and data management in IPMs. The results analysis and discussion

for the case study are presented in Section 5.4 and 5.5, followed by a broader discussion of the package in Section 5.6.

5.1 Object-oriented programming in Python

The `ABC_GP_IPM` package is implemented following the object-oriented programming (OOP) paradigm with `Python`, which improves code readability, reusability and ease of maintenance. In OOP, a class serves as a blueprint for creating objects, defining their behaviour through methods and setting their initial attributes. An instance refers to an actual object created from the class. For illustrative purposes, consider the initialisation process in `R`:

```
human <- list(name = "Marc", age = 29);
```

compare to `Python`:

```
human = Human(name="Marc", age=29).
```

In this `Python` example, '`Human`' represents the class (blueprint), and '`human`' is the instance (object) constructed with defined attributes '`name`' and '`age`'.

Python classes encapsulate both data (attributes) and functionalities (methods). For example, the '`human`' instance can perform actions like '`eat`' or '`sleep`' defined within the '`Human`' class, without needing to externally pass attributes like '`name`' and '`age`' each time when these methods are invoked. These attributes are bound to the instance and consistently accessible. This encapsulation allows each instance to operate independently with its own data, promoting a clear and organised code structure.

Additionally, OOP ensures that methods are specific to their relevant classes. For example, both the '`Human`' and '`Sheep`' classes might have methods for behaviours like '`eat`' or '`sleep`', but each class implements these actions according to its own context. This ensures that methods are relevant and specific to each class.

In the `ABC_GP_IPM` package, we define three classes to handle different use cases, organised hierarchically. For example, the '`Human`' class could be a parent class, with '`Female`' and '`Male`' as sub-classes that inherit common behaviours (methods and attributes). This hierarchy (also called inheritance) allows us to

create specific instances of '`Female`' or '`Male`' without duplicating code, making the overall structure more maintainable and efficient.

5.2 Case study 2: *Ovis aries*

We use the Soay sheep (*Ovis aries*) to demonstrate detailed implementations of `ABC_GP_IPM` package.

5.2.1 The study site

The case study focuses on the population of Soay sheep residing in Village Bay within the Island of Hirta, part of Scotland's St.Kilda archipelago. The study species, which are small, horned, and almost deerlike sheep, have thrived in St.Kilda for millennia, with historical records dating back between 3,000 and 4,000 years (T. H. Clutton-Brock and J. M. Pemberton 2004). They retain characteristics that are intermediate between modern domestic sheep and their wild ancestors, making them a valuable subject for evolutionary studies (Boyd and Jewell 1974).

Hirta, the largest island of the St.Kilda archipelago, was inhabited continuously until 1930, after which it was designated primarily as a nature reserve (National Trust for Scotland 2024). The catchment area of Village Bay, historically the principal settlement area on Hirta, encompasses a range of habitats, including grasslands, cliffs, and sand beach, providing a diverse environment for the Soay sheep (T. Clutton-Brock, J. Pemberton, et al. 2004).

In 1932, 107 Soay sheep were translocated to Hirta, with the population expanding to 1,114 by 1952 (Grubb et al. 1974). During the study period, approximately 300 of these sheep inhabit the 175-hectare catchment area of Village Bay (Simmonds and Coulson 2015). The population has been closely monitored since 1985, providing extensive datasets for analysis (T. Clutton-Brock, Illius, et al. 1997). The Soay sheep of Village Bay live in a predator-free environment with minimal interspecific competition and human intervention, closely mirroring the living conditions of their wild ancestors (Catchpole et al. 2000; Simmonds and Coulson

2015). This unique setting can offer a relatively simplified model for studying natural selection and their population dynamics in a near-pristine environment.

5.2.2 Interplay between Weather, Population density, and Vegetation resilience

The dynamics of the Soay sheep population at the study site are closely linked to food availability, which is influenced by both population density and weather factors (Crawley et al. 2004).

Weather conditions play a critical role in determining food accessibility and, consequently, the survival rates of these sheep ((Catchpole et al. 2000; Clutton–Brock and Coulson 2002; Ozgul et al. 2009; Coulson, Tuljapurkar, et al. 2010)). During harsh, wet, and windy winters, wild ungulates experience increased heat loss and reduced foraging opportunities (Forchhammer et al. 1998; Post and Stenseth 1998; Post and Forchhammer 2002), resulting in higher mortality rates of the study species ((Catchpole et al. 2000; Clutton–Brock and Coulson 2002)). Although spring vegetation biomass can be relatively high following such winters (Crawley et al. 2004), this does not buffer the population against the direct impacts of severe winter weather on survival (T. Clutton-Brock, Grenfell, et al. 2004). Summer conditions, in contrast, are generally favourable due to the abundance of food, resulting in no significant impact on sheep survival during this period (Crawley et al. 2004).

Population density can exert a substantial influence on food availability through increased intraspecific competition (T. Clutton-Brock, Illius, et al. 1997). During winters, high sheep densities intensify competition for limited food, often resulting in substantial mortality (T. Clutton-Brock, Stevenson, et al. 1996), particularly among juveniles (T. Clutton-Brock, Grenfell, et al. 2004). In contrast, periods of low population density are associated with reduced competition, which in turn promotes relatively better individual health and higher survival rates (T. Clutton-Brock, Grenfell, et al. 2004).

The resilience of the vegetation on Hirta is also pivotal in balancing sheep population density and food supply. Winter conditions variably affect vegetation

biomass on Hirta, with some years experiencing exceptionally high grazing pressures. Despite these fluctuations, the vegetation consistently demonstrates remarkable resilience — recovering rapidly by the subsequent spring and summer, and by August, achieving biomass levels that are comparable to those observed in typical years (Mysterud 2006; Crawley et al. 2004). This robust regrowth is critical to support surviving sheep, thereby facilitating their population recovery during the summer months. Although there is significant variability in vegetation biomass in the beginning of spring (March), the biomass at the onset of winter (August) exhibits considerably less variation (Crawley et al. 2004), indicating a relatively stable food supply during winter. This stability suggests that the primary challenge to winter survival are more likely due to reduced foraging opportunities from harsh weather and intraspecific competition, rather than the food reserves (Jones 2003; Crawley et al. 2004).

5.2.3 Data collection and the Study dataset

Since 1985, the Soay Sheep Research Project has been using individual tags to track each sheep from birth, through breeding attempts until death, building a robust database of over 8,000 sheep in Village Bay (Team 2024). Data collection is conducted by the field assistant visiting the island three times per year. Each visit involves comprehensive censuses and searches for deceased sheep, aiming to maintain accurate records of population density and mortality (T. H. Clutton-Brock and J. M. Pemberton 2004). The research expeditions are structured around three seasonal focuses:

- **Late Winter and Spring:** Daily searches to monitor winter mortality; newborn lambs are tagged, weighed, and blood sampled shortly after birth.
- **Summer:** Capture 50-70% of the population using netting traps for health assessments. Conduct a comprehensive sheep count to gather data about population density.

- **Autumn:** Monitor the rut, determine the oestrus status of ewes, and capture rams for genetic sampling to confirm paternity and understand reproductive success.

Our study dataset is focused exclusively on *females*, providing detailed individual information spanning from 1987 to 1996. The dataset includes identifiers, age, survival status, body weight for the current and next year, reproductive data (numbers and sizes of *female* newborns), and annual population sizes (which also count *males*) (Coulson 2012). Our model training process will utilise information from the first seven years (1987 to 1993), while the remaining three will be reserved for model testing.

5.2.4 Data analysis and Model settings

Data entries that were inconsistent with predefined conditions have been systematically removed prior to modeling. For example, each row in the data file is expected to represent a female sheep known to be alive by August census of each year (Coulson 2012). Entries with missing survival statuses are therefore removed to align with established criteria for living individuals

We further assumed that the observed data for individual sheep each year are representative enough for the unmeasured segments of the population. This assumption enables us to circumvent the complexities associated with an artificially predefined capture process when conducting IBM simulations in ABC sampling. More specifically, for example, unlike deer, Soay sheep show less hiding behaviour to their offsprings; juvenile and yearling females typically follow their mothers closely from birth (Grubb 1974; T. Clutton-Brock, Grenfell, et al. 2004). This behavioural trait suggests that the capture of a mother would logically imply the capture of her (newborn) lamb. However, a great number of missing values in newborns for observed mothers can be found in the study dataset. This discrepancy indicates that the capture process may be influenced by factors not fully understood or captured in our dataset. Therefore, by simplifying the capturing process, we facilitate a direct comparison between the simulated populations and the observed

datasets, acknowledging the potential nuances and limitations of this setting. In this case with ABC sampling procedure, since we only observed partial information, we applied a bootstrap method to supplement the missing individuals in the initial population structure. For instance, if the population size was 200, but only 130 were observed, we randomly sampled with replacement from these 130 individuals to reach the full population size of 200.

In the study dataset of the Soay sheep population, the occurrence of twin births happens in less than 1.9% of cases. We follow Simmonds and Coulson 2015; Coulson 2012 to exclude it to simplify the modeling process. In this case, we assume that, in the rare event of twins being born, only one female offspring survives to the next census. Our IPMs simplify the complexity of individual variability by representing each sheep solely by its measured size z. The population with size z' at the next time step is described by:

$$n(z', t+1) = \int (s(z|\boldsymbol{\omega})G(z'|z, \boldsymbol{\omega}) + p_f(z|\boldsymbol{\omega})r_{size}(z'|z, \boldsymbol{\omega}))n(z, t)dz,$$

where $\boldsymbol{\omega}$ represents living condition factors influencing vital rates. Specifically, the model describes that a female individual of size z has a survival probability $s(z|\boldsymbol{\omega})$ and follows a growth trajectory determined by $G(z'|z, \boldsymbol{\omega})$, over an annual cycle ending each August. For reproduction, it starts with considering the probability of recruitment, $p_f(z|\boldsymbol{\omega})$. This is the likelihood that a female of size z will become pregnant, then give birth to a *female* newborn, and eventually have that newborn survive to the next year. Conditional on the such recruitment, the female newborn is expected to grow to size z' with $r_{size}(z'|z, \boldsymbol{\omega})$ by the next August, influenced by her mother's size z. Since the data accounts for female newborns only, there is *no* need to set up an additional sex rate for recruitment in our IPM; such a rate is already built into the observed datasets.

Living condition factors encapsulated by $\boldsymbol{\omega}$ include winter weather conditions and initial population densities each August. Given the constraints imposed by the limited seven-year span of our training dataset, strategic modifications about this 2-D vector, $\boldsymbol{\omega}$, were essential. These adjustments address the challenges posed by

sparse data points across its continuous variable range. Although such modifications inevitably result in some information loss, they have been carefully crafted to retain critical insights from previous research on the dynamics of the Soay sheep population.

- Winter weather conditions of St.Kilda are largely influenced by the North Atlantic Oscillation (NAO), a climatic phenomenon known to affect many marine and terrestrial ecosystems (Forchhammer et al. 1998; Belgrano et al. 1999). The NAO index, reflecting large-scale atmospheric patterns over the North Atlantic (Rogers 1984), can lead to winters on St.Kilda that are either windy and wet or colder and drier (T. Clutton-Brock, J. Pemberton, et al. 2004). Particularly, stormy winters, indicated by high NAO indices, limit ungulate movement and increase thermal stress in the wild (Forchhammer et al. 1998; Post and Stenseth 1998; Post and Forchhammer 2002)), which significantly in turn adverse consequences for the sheep population detailed in T. Clutton-Brock, Grenfell, et al. 2004.

  To address the marked influence of *high* NAO indices, $\boldsymbol{\omega}$ first consider an averaged NAO index from December to March (Simmonds and Coulson 2015); then, a floor function is employed to adjust the averaged NAO index, eliminating negative values to focus exclusively on the significant impacts of high positive NAO values on sheep dynamics.

- In addition to the impact of winter weather, our analysis also considers the influence of population density on the population dynamics. While we previously acknowledged that population growth is significantly affected by population density, we observe that this relationship is not linear. Analysis in T. Clutton-Brock, Grenfell, et al. 2004 reveals a pronounced threshold effect on St.Kilda island: below a population size of approximately 1,100 to 1,200 animals, the population tends to increase by an average of 1.27 per animal per year; however, exceeding this threshold triggers a decline at an average rate of about 0.2 per animal per year. Additionally, a strong positive correlation

exists between the populations on St.Kilda island and those in our study site, the Bay area.

Building on these insights, we have adopted a threshold value for population density in our models, estimated at $exp(5.9)$ according to graphical results presented in T. Clutton-Brock, Grenfell, et al. 2004. Consequently, in our models, population density is treated as a binary variable: it is coded as 1 if initial population densities in August surpass $exp(5.9)$; otherwise, it is coded as 0.

GP models were employed for modelling vital rates (see Appendix C for implementation details).

5.3 Getting start with ABC_GP_IPM

To enable a easier code reusability and maintenance through the package, the first step is to create an instance of a class that manages data and IPM vital rate models related to the target species.

5.3.1 Initialisation

The initiation of the package involves creating an instance of the '`GP_IPM`' class, encapsulating population data and GP models for various vital rates. In the example of Box 1, the constructed instance, '`sheep_gp_ipm`', encapsulates the population data in two key formats: (1) The '`popu_data`' attribute represents the consolidated dataset across all training years, used for model fitting in Step 1 of the workflow in Figure 5.1. (2) The '`POPUdata_dict`' is the segmented version of '`popu_data`', broken into smaller datasets sorted by consecutive years. It is used later for comparisons with simulated time series datasets. Furthermore, constructing the instance '`sheep_gp_ipm`' requires the fitted GP models for all vital rates, including MLEs ('`GPmodel_mle`'), Bayesian GPs with priors ('`GPmodel`'), and the corresponding MCMC samples ('`MCMC_samples`').

Box 1: Initialisation

```
# Initiate: build an instance of GP_IPM
sheep_gp_ipm = GP_IPM(popu_data=train_data, POPUdata_dict=list_popu,
                      GPmodel_mle=gp_mle_models,
                      GPmodel=GP_models, MCMC_samples=mcmc_samples)

# Incorporate custom functions
sheep_gp_ipm.add_XY_compu(XY_m_grw_compu)
sheep_gp_ipm.add_model_build(build_new_m_grw)
```

The package supports the integration of custom functions to adapt to specific data structures and modeling needs. The method '`add_XY_compu`' allows users to add custom functions for extracting data specific to vital rates. For example, as demonstrated in Box 1, the function '`XY_m_grw_compu`' — which processes '`popu_data`' to generate independent and dependent variables for the growth vital rate — has been added to the instance. Similarly, the '`add_model_build`' method allows the user to specify how the GP model for a given vital rate, such as growth, should be initialised via a custom function like '`build_new_m_grw`'.

A systematic naming convention is utilised throughout the package to ensure clarity and maintainability. The naming convention employs a prefix "m_" followed by the vital rate name to clearly delineate related methods and variables. For example, '`GPmodel_mle`', '`GPmodel`', and '`MCMC_samples`' are `Python` dictionaries with keys such as "m_grw", reflecting models and data pertaining to growth vital rates. This consistent use of naming conventions extends to custom functions, such as '`build_new_m_grw`' and '`XY_m_grw_compu`', ensuring the instance correctly associates each method with the appropriate vital rate. This naming convention will be strictly enforced during implementation, with input validation checks in place to verify adherence.

With the initialisation complete, we make the next three sections correspond to the steps of the workflow shown in Figure 5.1, providing a clear structure for readers to follow.

5.3.2 SS selection

To select summary statistics for further ABC sampling, the package adopts a systematic workflow based on Negative Log Posterior Probability (NLPP) per-

Box 2: SS selection

```
# Build an instance to select summary statistics for the growth: m_grw
sheep_grw_pert = Perted_IPM(sheep_gp_ipm, target='m_grw', opt_percentage=6,
                            fun_IBM=IBM_1step_gp,
                            name_ss=name_ss, fun_ss=list_comparisons)
```

```
# (1) Optimum NLPP
sheep_grw_lp = sheep_grw_pert.log_posterior_density()
# (2) Simulations at the Optimum
summary_opt = sheep_grw_pert.ss_optVSopt(log_posterior_density=sheep_grw_lp,
                                         test_data=test_dataset, repetition=1000, num_cores=8)
# (3) NLPP Perturbations & Re-simulations:
summary_around_opt = sheep_grw_pert.ss_optVSmcmc(log_posterior_density=sheep_grw_lp,
                                                 test_data=test_dataset, repetition=1000, num_cores=8)
# (4) Summary statistics Selection:
sheep_grw_pert.top_ss(summary_opt, summary_around_opt, threshold=0.7, num_top_returned=10)
```

turbation (detailed in Section 3.3). This approach selects summary statistics individually for each vital rate.

To determine which summary statistics are more sensitive to shifts within the model, we first need to introduce what aspects of the model are being perturbed and how these perturbations are applied. The '`Perted_IPM`' class, a specialised subclass of '`GP_IPM`', is designed to manage these perturbations. In the provided example (see Box 2), '`sheep_gp_ipm`', an instance of '`GP_IPM`', is perturbed specifically for the growth vital rate ('`m_grw`') to generate a perturbed IPM object, '`sheep_grw_pert`'. Only the first 6% of MCMC samples closest to the optimal NLPP are retained, representing the perturbation results (more details latter).

Additionally, to handle diverse scenarios, three user-defined attributes tailored to specific study species are necessary. The function '`fun_IBM`' is used to perform 1-step forward IBM simulations and generate simulated datasets. The '`fun_ss`' function is then employed, computing a list of potential summary statistics and assessing the distance between datasets. These candidate statistics are named as specified by '`name_ss`'.

The subsequent steps strictly adhere to the workflow proposed in Section 3.3. **(1)** Optimal NLPP: through the constructed perturbed object '`sheep_grw_pert`' with method '`log_posterior_density`', we compute NLPP scores for all MCMC samples to identify which are with or near the optimal NLPP. **(2)** Simulations at the Optimum: we estimate the distance between two datasets simulated under the same hyperparameters associated with the Optimum NLPP through method

'ss_optVSopt'. The argument 'test_dataset' is used as the initial population structure in a 1-step IBM simulation function to generate 'repetition=1000' pairs of datasets. This process is set up to use 'num_cores=8' cores parallel computing to optimise efficiency. **(3)** NLPP Perturbations & Re-simulations: subsequently, though method 'ss_optVSmcmc', a similar process is followed for the first 6% of MCMC samples closest to the Optimum NLPP (6% was stated when constructing 'sheep_grw_pert'). For each of these samples, we estimate the distance between 1,000 pairs of datasets: one set generated using the hyperparameters at the Optimum NLPP, and the other with hyperparameters from within the 6% closest to the Optimum. **(4)** Summary Statistics Selection: eventually, to evaluate the sensitivity of summary statistics to small deviations from the optimal settings, the area under the receiver operating characteristic curve (AUC) is calculated. An AUC threshold of 0.7 is set to determine sensitivity, and ultimately, the top 10 most responsive summary statistics are reported based on their performance by method 'top_ss'. Users can thereby manually select summary statistics based on the returned results.

5.3.3 Cache

To streamline computational efficiency, especially in the context of large-scale simulations in ABC sampling, caching mechanisms are employed. The caching process involves storing the results of computationally intensive tasks, like the Cholesky decomposition needed for GP models, which are frequently used throughout the sampling. With the package, a single line of code activates caching for all vital rates defined within the instance (Box 3). This command stores results in the 'Lm' directory under the 'true_popu' parent directory, dynamically adapting to the current script's execution environment through command 'os.getcwd()'.

Box 3: Cache

```
# Cache
sheep_gp_ipm.calculate_cache(file_path=os.getcwd()+f"/true_popu/Lm")
```

Once the cache are set, the package automatically randomly draws and loads MCMC samples for each vital rate in the background. For each sample, predictive

distributions are then calculated in two ways: (1) using the locally stored cache results just computed; (2) recalculating from scratch without the cached data. The package then compares these two sets of results to confirm their consistency, ensuring that the cached computations are accurate and reliable.

Since caching is intended to expedite ABC sampling processes, these caches need to be stored in a location that allows for rapid access by the ABC sampling mechanism. Additionally, due to the large storage demands of caching, it is strongly recommended to execute this program directly on same server used for ABC sampling, rather than execute it locally and then transferring to that server. This way helps to avoid potential delays and resource constraints during ABC sampling..

5.3.4 ABC sampling

We are now ready to proceed to the core phase: ABC sampling. Executing a ABC sampler needs assistance of '`ABC_GP_IPM`' class. At this stage, it is assumed that all necessary preprocessing for ABC have already been completed. Therefore, see Box 4, the attribute function '`fun_ss`' required by '`ABC_GP_IPM`' class is now configured to compute only the list of selected summary statistics. These final listed statistics are named as specified by '`name_ss`'. In the sheep example, one summary statistic is chosen for each of the four vital rates.

For IBM simulations, instances of class '`ABC_GP_IPM`' use the '`fun_IBM_cache`' method, which leverages pre-computed cached data for sampling efficiency. To switch from the standard '`fun_IBM`' method to '`fun_IBM_cache`', simply replace the call to '`predict_y`' method with '`predict_y_loaded_cache`' for each of GP models involved in '`fun_IBM`'. The method '`predict_y_loaded_cache`' is well-established to speed up sampling, and is readily available for import from our package.

The '`fun_IBM_cache`' conducts IBM simulations using the observed initial independent variables ('`ini_popu`'). The resultant dataset of this method, representing simulated one-year data, is subsequently processed by the '`fun_structure`' function. This user-defined function is responsible for generating the independent variables needed to initiate the next year's simulation. Additional parameters for

Box 4: ABC sampling

```
test_abc_gp_ipm = ABC_GP_IPM.ABC_GP_IPM(GP_IPM_instance=sheep_gp_ipm,
                                        fun_ss=list_comparisons_interested, name_ss=['2c', '4e', '6f', '8g'],
                                        ini_popu=ini_popu, fun_IBM_cache=IBM_1step_gp_cache, fun_structure=popu_structure,
                                        fun_structure_kwargs={'popsize_list':popsize, 'nao_list':nao})
                                        # fun_capture, fun_capture_kwargs

# Implement ABC sampling
test_abc_gp_ipm.ABC_PMC(quantiles=np.array([0.55, 0.5, 0.45, 0.4, 0.34, 0.28, 0.18, 0.13, 0.1, 0.09]),
                        n_particles=np.array([400000, 250000, 150000, 80000, 50000, 25000, 11000, 7000, 4000, 1000]),
                        num_cores=40, file_path2store=os.getcwd() + f"/ABC_details")
```

'`fun_structure`' can be supplied through the '`fun_structure_kwargs`' argument. For instance, in our sheep model, '`fun_structure`' also require the observed population density ('`popsize_list`') and NAO index ('`nao_list`') for the subsequent simulation year (Section 5.2).

In our Soay sheep case study, as mentioned in Section 5.2, we assumed that the observed individuals each year are representative enough of the unmeasured population segments. Based on this assumption, we have ignored the capturing process to avoid introducing unnecessary complexity into the simulation. However, users with specific insights or concerns about the capture process can still implement a custom capture function to better reflect their data structure. In this package, the custom capture function can be implemented via the '`fun_capture`' argument. By providing this capture function, the constructed instance is enabled to transform the simulation output of the entire population into a format comparable to the observed datasets during the ABC sampling process. The function should take the simulation output and the simulation year as key inputs, with additional arguments passed through '`fun_capture_kwargs`'.

We have finally reached the last step: performing the ABC sampling using the '`ABC_PMC`' method. The algorithm proposed by Section 3.4 is all set up and can be easily implemented by specifying the quantiles and the number of particles ('`n_particles`'). The sampling process in Box 4 is executed with parallel computing of 40 cores. The resulting output files will be stored in the directory specified by the '`file_path2store`' attribute.

5.3.5 Forecasting

The 'ABC_PMC' method of 'ABC_GP_IPM' class ultimately results in a set of parameter estimates that represent the approximate posterior distribution, ready for further analysis or predictions. Our package also provide method 'simulate' for instances belonging to class 'ABC_GP_IPM', enabling users to easily utilise the results returned from 'ABC_PMC' to perform population dynamic simulations.

'abc_index' and 'abc_weight' are output files of 'ABC_PMC'. The 'abc_index' is an array containing indices that reference specific MCMC samples. Each index corresponds to a particular parameter configuration from the estimated ABC posterior distribution. The 'abc_weight' array holds the associated weights, reflecting the relative importance or likelihood of each sample. During simulation in 'simulate', the method randomly selects an index from 'abc_index' based on the probabilities in 'abc_weight'; the corresponding parameter configuration is then used in the simulation of population dynamics.

Box 5: Forecasting

```
simulation = test_abc_gp_ipm.simulate(abc_index=abc_index, abc_weight=abc_weight, num_samples=1000,
                                      simu_length=9, fun_IBM=IBM_1step_gp, fun_pop_metric=pop_metric)
```

The method 'simulate' allows the user to specify the number of samples to generate ('num_samples'), the length of the simulation ('simu_length'), and two customisable key functions: 'fun_pop_metric', which calculates population metrics of interest during the simulation (such as population size or growth rate, etc.), and 'fun_IBM', which performs user self-defined one-step forward IBM simulation. In our example in Box 5, we are running this method locally, so we used 'IBM_1step_gp'. However, if the method is executed in an environment that allows for rapid access to cached information, it can be also replaced with, for example 'IBM_1step_gp_cache', to improve efficiency.

The 'simulate' function may not be an essential part of a general workflow when applying GP and ABC to IPM. However, we include it here not only for users who may find it useful, but also to provide an example for understanding how the

results returned from '`ABC_PMC`' should be processed. By inspecting the source code of '`simulate`', users can gain insights on how to handle the output and may even develop further analyses or other types of simulations tailored to their own datasets.

5.4 Case study 2: Results

We generate ABC samples using the model setup discussed in Section 5.2 and the code provided in Section 5.3 via the developed `ABC_GP_IPM` package. These samples are then used to evaluate the ABC_GP models' predictive performance and compared with those from MCMC and GLM methods, as demonstrated in the previous *C. flava* case study.

For the **long-term** forecasting, we initialise the population structure in 1987, using a bootstrap method to account for missing individuals. Predictions are then generated year by year, based on the population simulated from the previous year's data. For the **1-step** forecasting, we use the observed (incomplete) population structure from the previous year, supplement it with the same bootstrap method, and then make only a single year predictions, keeping the remaining procedures consistent with the approach detailed earlier of *C. flava*.

From long-term population forecasting on the Figure 5.2 left, all models show a general increasing trend in population size over time, with GP and ABC_GP closely aligned until 1993, after which ABC_GP predicts slightly higher values with larger uncertainty. GLM, while following a similar trend, deviates more from the other two models after 1993, showing wider uncertainty bands. Moreover, all models exhibit increased uncertainty in the later years suggesting the challenges of making long-term predictions over time.

Compared to long-term forecasting, the 1-step predictions on the right-hand side have narrower uncertainties (as they use more immediate observed data). In 1-step population forecasting, all models display sharper fluctuations, especially in the early years. Here, GP and ABC_GP again closely track each other, with ABC_GP predicting slightly higher population sizes in the later years. GLM remains the most variable model, with larger deviations and uncertainty.

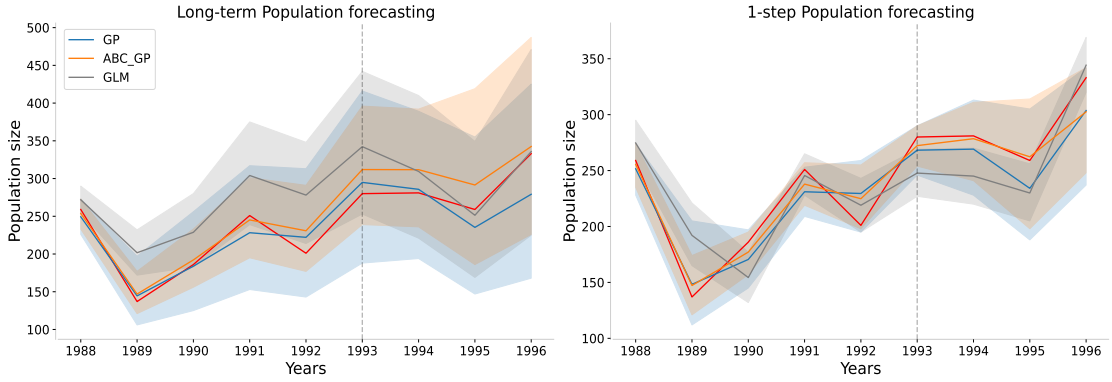


Figure 5.2: Two types of population size predictions, along with their corresponding 95% P.I. (shaded areas), were generated using GP, ABC_GP and GLM samples through 1,000 IBM simulations. The blue predictions were based on MCMC samples, while the orange and grey ones were based on ABC_GP and GLM posterior samples, respectively. The point estimations (means) are represented by solid lines in their corresponding colours. The red line represents the true observed population size. The model training process utilised data from 1987 to 1993 (before the grey dashed vertical line). The red line represents the true observed population size.

Let us take a closer look at their point estimations. We use absolute relative differences between the true and predicted population size (Figure 5.3) to assess each model's overall performance, without being affected by whether over- or under-estimated. Normalising these differences relative to the true population size ensures that errors are comparable across different population sizes and time points.

Figure 5.3 left shows the absolute relative differences in long-term forecasting, where both GP and ABC_GP models perform well, with the differences staying below 0.2. In contrast, the GLM model shows much larger deviations, peaking around 0.4 in years like 1989 and 1992, indicating relatively less predictive accuracy. The right plot in Figure 5.3, focused on 1-step forecasting, shows smaller errors across all models compared to long-term forecasting. Both GP and ABC_GP maintain strong short-term accuracy, with differences remaining under 0.1 for most years, while GLM again shows more variability.

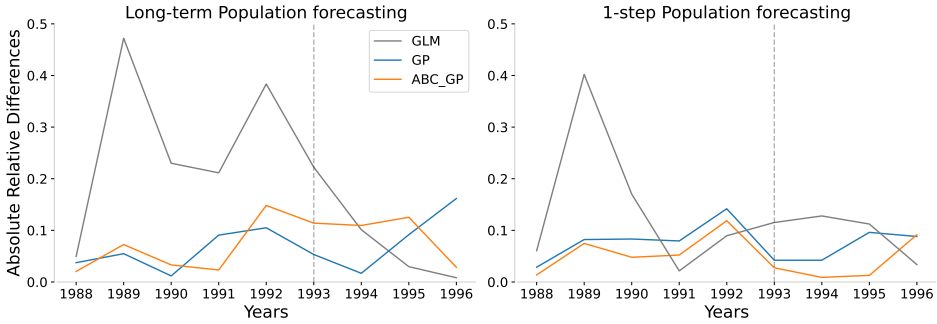


Figure 5.3: Absolute differences between the observed and predicted population size (means of the predictions) relative to the true observed population size.

An interesting point emerges with GLM, particularly in long-term forecasting, where it shows large errors during the training years but surprisingly smaller errors in the test years. However, this may not be a positive sign. The seemingly improved performance in the test period might not reflect GLM's true predictive power, but rather chance alignment with the test data, raising concerns about the model's reliability for future predictions.

On the other hand, the GP model performs well during training but shows (slightly) increasing errors in the test years, which might suggest potential over-fitting — the model may struggle to generalise to unseen data.

In contrast, ABC_GP stands out for its consistency across both training and test years. While its errors are not always the smallest, they remain stably acceptable across both phases. This stability makes ABC_GP, while not perfect, a balanced option for both short- and long-term forecasting.

Figure 5.4 offers another perspective on the 1-step forecasting results by focusing on the estimated distribution of population growth rates (the one with the similar pattern for long-term is shown in Appendix C). During the training years, the prediction intervals (P.I.) across all models appear fairly comparable in width. However, in the test years, the GP and ABC_GP models tend to have wider P.I. compared to GLM.

Consistent with earlier observations of *C. Flava*, GLM's narrower intervals likely reflect a more constrained or rigid model structure. This does not necessarily translate to better accuracy. In fact, GLM often fails to include the true population

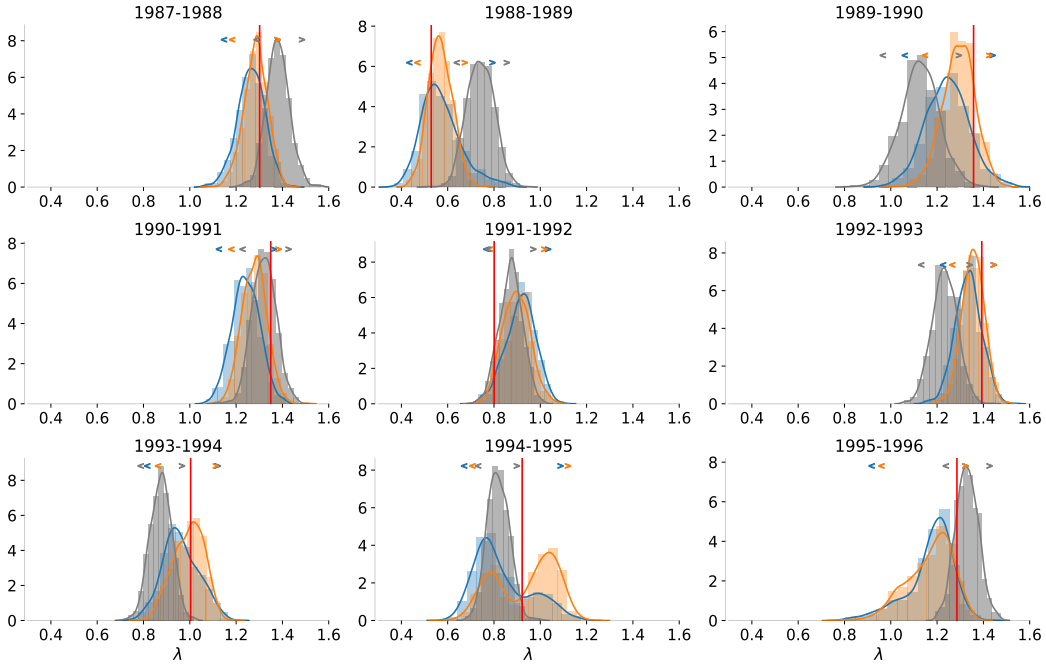


Figure 5.4: Histograms represent the density of simulated population growth rates generated through **1-step** forecasting. The associated 95% prediction intervals are denoted by angle brackets, each in its corresponding colour, positioned at the top of each histogram. The red vertical line represents the true observed population growth rate.

growth rate within its P.I., particularly during the years 1988-1989-1990 and 1992-1993-1994-1995, revealing its limitations in capturing the underlying population dynamics. On the other hand, while GP and ABC_GP exhibit wider P.I., they are generally better at covering the true population growth rate, reflecting a better balance between uncertainty and accuracy.

A notable feature of ABC_GP is the bimodal distribution in 1994-1995, where two distinct peaks reflect different possible growth patterns contradicting to each other — one indicating population growth and the other suggesting shrinkage. The true population growth rate lies between these peaks, making both the mean and median predictions from ABC_GP rather good for this year. This bimodal result may suggest that, constrained by the current ABC summary statistics, multiple plausible outcomes exist. It may also indicate a limitation of the proposed ABC sampler, as the fixed MCMC sample could prevent ABC from exploring a broader parameter space — potentially leading to fewer results concentrated between the

peaks, where the true value is located (discussed in **Employing MCMC samples as ABC priors** in Section 6.1).

5.5 Case study 2: Discussion and Future works

In this case study, building on previous research, we introduced novel covariates by modifying the NAO index and population density to reflect environmental factors affecting population dynamics. While these factors have been considered in past studies, our modifications represent a new contribution. These enhancements helped models to capture the general population size trends in line with the true observations. Additionally, incorporating ABC and GP methods provided more accurate point estimates and more reasonable prediction intervals, further improving performance compared to traditional approaches like GLMs.

However, during the ABC modeling, one key assumption was that the observed data for individual sheep each year were representative enough of the unmeasured population segments. Given the complexities observed in the capture process — such as the discrepancy between observed mothers and missing newborns (Section 5.2) — future research should incorporate domain-specific knowledge to propose a more appropriate and biologically meaningful capturing process. This could help bridge the gap between the simulated and observed datasets, improving their comparability during ABC sampling.

Additionally, to simplify this example as a demonstration of the `Python` package, we did not include age as a state variable, despite previous studies highlighting its noted impact on Soay sheep dynamics (Coulson, Catchpole, et al. 2001; King et al. 2006; Ravindran et al. 2022). Incorporating age in future models may hold potential of improving the accuracy of the predictions.

5.6 Discussion

This `Python` package provides a streamlined and user-friendly interface for implementing the ABC_GP IPM approach. By integrating the selection of ABC

summary statistics and the ABC sampling process, it simplifies complex workflows into several lines of `Python` code, making the method more accessible and allowing researchers to focus on their biological inquiries.

One of the key benefits of the package is its flexibility, allowing users to customise functions, such as those for IBMs. This adaptability allows it to accommodate specie-specific life cycles, making it suitable for various ecological contexts. However, users must follow specific conventions for inputs and outputs, promoting flexibility, reliability, and reproducibility.

While the package supports the early stages of demographic modeling, it does not include tools for constructing IPMs, as established R packages like `IPMpack` and `ipmr` have addressed this need. For those integrating `Python` with `R`, tools like `rpy2` can facilitate interoperability. However, a primary limitation is the use of `GPflow` for GP modeling, which is deeply integrated with `TensorFlow` in `Python`. Currently, there is no straightforward way to transfer fitted GP models directly to `R`, meaning predictions and outputs must be completed in `Python` before transferring results to `R` for further analysis.

Additionally, most established IPM tools in `R` are designed to accept direct model structures, such as formulae describing GLMs, and are not equipped to incorporate predictions from external sources, such as predictions about meshpoints generated by `Python`-based models. While the online guide of our package offers `Python` scripts to assist in constructing IPM matrices from these predictions, this solution requires researchers to either adapt their existing `R` workflows to include `Python` components or manage separate workflows in both programming languages. Users who primarily work with `R` may find the approach challenging.

6

Discussion and Future works

Contents

6.1 Algorithm discussion 107
6.1.1 ABC's role in revealing interactions detected among vital rates 107
6.1.2 Managing background knowledge integrated in ABC GP IPM 108
6.1.3 Employing MCMC samples as ABC priors 109
6.1.4 Lacking sufficiency in non-parametric modelling 111
6.2 Future works 112
6.2.1 A new sampler with a more appropriate structure . . . 112
6.2.2 A new sampler with a less-restricted prior 118

In this thesis, we apply GP models and ABC methods to a combined model: IPM. Understanding how GP models and ABC interact is crucial for comprehending the proposed ABC_GP IPM. GP models, as Bayesian non-parametric models, could theoretically encompass an infinite number of parameters. However, with real-world finite datasets, they only utilise a finite subset of these parameters, with the remainder being effectively marginalised out (Orbanz and Teh 2010). Applying ABC to such models introduces selected summary statistics as additional data points, thereby expanding the existing dataset. As a result, as more data points

become available, the GP models can adaptively evolve to become more complex, reflecting the increased volume of information.

Under this understanding, the ABC_GP IPM increases model complexity without fundamentally altering the modeling approach, making it a data-driven method that requires no additional assumptions beyond trust in our observations. Despite its data-driven nature, the approach maintains clarity and interpretability. In addition, ABC_GP IPM delivers these improved outcomes using identical datasets as standard IPMs, making it more accessible for studies previously using traditional IPMs. Its adaptability also broadens the potential for IPMs to serve as a versatile approach in context-dependent demographic research.

We conclude with some final remarks regarding the proper interpretation of our approach and potential concerns.

6.1 Algorithm discussion

6.1.1 ABC's role in revealing interactions detected among vital rates

Our methodology integrates ABC with IPMs, enabling the detection and incorporation of interactions among vital rates. By leveraging carefully selected summary statistics that encapsulate population-level information, it captures patterns, for example, the 'trade-off' between high survival rates and low fecundity.

However, it is important to clarify that our method's capacity to detect such interactions should be viewed as a bonus from employing ABC in IPMs, rather than a dedicated tool for studying the joint behaviours. This feature arises from the inherent design of ABC, which retains only those vital rate combinations that reproduce observed population dynamics. Here, the developed ABC-PMC sampler leverages population-level information to narrow down under-determined vital models, retaining only the joint distributions that satisfy specific criteria.

In this sense, ABC_GP is not intended to infer causal or mechanistic relationships, but rather to recalibrate the ensemble of sub-models such that their combined output aligns more closely with empirical population-level dynamics.

The interactions observed in posterior samples should therefore be interpreted as statistical alignment resulting from this recalibration process, rather than as evidence of explicitly modelled biological dependence.

As these inherent interactions are not systematically formulated, they might be tricky to harness in studies where the sensitivity of these interactions can substantially impact uncertainty of model predictions (Fieberg and Ellner 2001) (since varying these interactions can be difficult (Ellner, Childs, et al. 2016)).

While our approach can not directly explain how interactions shape life-history evolution, it equips IPMs to better 'mirror' observations via ABC. This enhancement could open doors to studying interactions from a fresh perspective. By successfully replicating observed behaviours and generating predictions, these models can guide researchers in designing new experiments or identifying the type of data required to further test these mechanisms (Farrell et al. 2021).

6.1.2 Managing background knowledge integrated in ABC GP IPM

In the Introduction (Chapter 1), we proposed that our model offers scientists greater flexibility in integrating their expertise with IPMs. This claim, however, may spark discussion, as prior ABC research suggests that unsuitable summary statistics can bias inferences (Prangle 2017). This point is also exemplified in Section 3.2 where an unreliable inference result was generated due to our over-reliance on an unsuitable summary statistic in a Poisson model. This conclusion was drawn on the premise that the true underlying model was Poisson.

However, in real-world practice, it is critical to ask: "who genuinely knows the true model that generated the observed data?" If our background knowledge suggests greater confidence in the chosen summary statistic over the Poisson assumption, it implies that the Poisson model, not the summary statistic, may be the one 'ill-suited'. In this case, the derived ABC posterior approximation can be therefore treated as a set of parametric values most likely to align with our intended results under this (potentially) unsuitable Poisson model. In reality, where the true model

is unknown, an observed summary statistic should indeed hold potential to serve as a valid summary statistic in ABC, especially when aligned with expert domain knowledge affirming its relevance.

The debate, concerning the appropriateness of the model or summary statistics, may provide another reason for applying ABC to non-parametric models, like in our case with GP models. Unlike parametric models, GPs do not assume a fixed form for underlying probability functions. They can potentially have an infinite number of parameters — the model size can (adaptively) grow alongside with the data size (Gershman and Blei 2012; Y. Li et al. 2019), challenging the typical fixed-dimension assumption of parametric models.

While our methods offer a flexible way to integrate users' own background knowledge into IPMs, it is crucial to control the volume of content added to avoid falling into the curse of dimensionality. We suggest constructing a candidate list comprising all aspects of interest and then employing a systematic approach to pick out the final statistics, as demonstrated in our study. Some statistics in the candidate list may bring overlapping information, causing certain statistics to become less informative when others are present. Techniques like Principal Component Analysis can be employed to reduce dimensionality, but the resultant statistics might not be as interpretable as their original counterparts.

6.1.3 Employing MCMC samples as ABC priors

Recall that, for a finite subset of random variables from a GP with hyperparameter $\theta \in \mathcal{P}$, this corresponding realisation of the GP can be expressed as

$$f_{GP}|\theta = L_\theta z, \quad z \sim \mathcal{N}(0, I), \tag{6.1}$$

where I is an identity matrix and L_θ is the Cholesky decomposition of the GP kernel matrix $K_\theta = L_\theta L_\theta^\top$. Inference and prediction using GP models require L_θ to compute matrix inversions and determinants, both of which exhibit a complexity of $\mathcal{O}(N^3)$ with data size. As a result, the application of IPMs in conjunction with GP models through fully Bayesian inference becomes highly computationally demanding.

Our proposed ABC sampler specifies the prior to be constrained by a collection of points, denoted as $\pi(\boldsymbol{\theta}) \propto \mathcal{U}(\boldsymbol{\theta})\mathbb{1}(\boldsymbol{\theta} \in \mathcal{X})$, where the hyper-set $\mathcal{X} \in \mathbb{R}^{dim(\boldsymbol{\theta})}$ includes all possible combinations of distinct groups of MCMC samples. This indicates that our methods do not heavily rely on MCMC algorithms to provide a reliable approximation of marginal distributions, but only to return a hyper-set, representing our belief in an expansive region where the target $\pi_{ABC}(\boldsymbol{\theta}|\boldsymbol{s}^{obs})$ is likely to be found. In our method, L_θ were pre-calculated and stored locally, accordingly the requisite matrices can be accessed directly, thereby obviating additional computation and processing time.

However, this convenience comes at a cost. Our version of ABC-PMC assumes that the ABC target must be contained within $\mathcal{X}$, a discretized subset of $\mathcal{P}$. Therefore, the primary concern that may arise is the possibility of the true high ABC posterior region not being encompassed by the hyper-points constituted by the pre-stored MCMC samples. This is particularly likely to occur in high-dimensional models. Increasing the population size of MCMC samples might help, but it could also introduce additional challenges. First, there is no clear guideline about how big the size of MCMC population is enough. Second, the number of MCMC samples cannot be arbitrarily increased due to local storage constraints; pre-calculated matrices for 5,000 MCMC samples per vital rate already consume approximately 100 GB of storage space. Furthermore, increasing the number of MCMC samples could potentially make the current ABC sampler less efficient. This is because, for each vital rate to be updated, the current employed perturbation kernel only offers a random draw from the corresponding MCMC samples and does not provide the sampler with any guidance to move towards the region of high ABC posterior probability.

From another perspective, simply increasing the number of MCMC samples does not appear to fully address the core issue, especially when the MCMC posterior shape differs considerably from the shape of the ABC target. For example, if the marginal ABC mode for a vital rate resides in the low MCMC posterior area, the pre-stored MCMC samples in this low probability region will be sparse. Consequently,

a significant number of ABC samples may be heavily concentrated on a single or few MCMC samples (from a marginal perspective), resulting in a very low ESS and a small number of unique samples for that vital rate. This was evident in our study in Section 4.4, where, out of 200 samples, only 37 were unique from the marginal aspect of the vital rate `fec`. A potential working direction regarding this concern is proposed in Section 6.2.2 of Future Works.

Another possible approach to mitigate this limitation is to deliberately relax the sub-model posterior during MCMC sampling. This aims to produce more diffuse samples and construct a broader hyper-set $\mathcal{X}$. Specifically, for a sub-model with parameters θ_v and data y_v, instead of sampling from the standard posterior $\pi(\theta_v)p(y_v \mid \theta_v)$, one could use a tempered version:

$$\pi(\theta_v)p(y_v \mid \theta_v)^{\eta}, \quad \text{with } \eta \in (0, 1).$$

This approach, sometimes referred to as *tempered MCMC* or *power posteriors*, produces more diffuse samples (Friel and Pettitt 2008; Neal 1996). The resulting marginal samples cover a wider region in parameter space, potentially improving the overlap between $\mathcal{X}$ and the target ABC posterior $\pi_{ABC}(\boldsymbol{\theta}|\boldsymbol{s}^{obs})$.

However, this comes with two costs: it reduces the influence of the data y_v, so the resulting distributions no longer match the standard posterior implied by the model; and it introduces an additional tuning parameter, η. In our case, since these results are only used as priors for the ABC stage, the first cost may be acceptable. As for the second, its impact depends on how the method is implemented in practice.

6.1.4 Lacking sufficiency in non-parametric modelling

On a different note, when applying ABC and Bayesian non-parametric models into combined models, the concept of sufficient statistics, in the usual sense of the term, is less clear and might not be applicable given our current knowledge. Non-parametric models do not make strong assumptions about the data distribution or the functional form of the model, making it challenging to identify a set of statistics that can summarise all the relevant information in the data. These models are

designed to capture as much information as possible from the data, often utilising the entire dataset for predictions. In this case, the 'statistics' calculated from the data should essentially be the data points themselves. As such, while there might be ways to summarize the data in non-parametric problems, but should not referred to as "sufficient statistics" in the same sense as in parametric models. Additional theoretical work is required to clarify how ABC can be applied to non-parametric models, particularly considering the lack of a clear concept of sufficiency in these models, as compared to their parametric counterparts.

6.2 Future works

6.2.1 A new sampler with a more appropriate structure

Thus far, we have discovered the application of ABC samplers for static problems to perform direct inference on GP hyperparameters, by generating samples from the joint distribution

$$\pi_{ABC}(\boldsymbol{\theta}, \boldsymbol{s}^*|\boldsymbol{s}^{obs}) \propto \mathbb{1}_{\boldsymbol{d}(\boldsymbol{s}^{obs},\boldsymbol{s}^*)<h_C} w_C(\boldsymbol{\theta})\mathcal{M}(\boldsymbol{s}^*|\boldsymbol{\theta})q_C(\boldsymbol{\theta}), \tag{6.2}$$

as stated in Section 3.1. However, here, we identify two primary shortcomings related to the proposed sampling framework. Addressing these issues might greatly improve the performance.

1. Our goal is to approximate the objective distribution $\pi(\boldsymbol{\theta}|\boldsymbol{s}^{obs})$ using the ABC approximation $\pi_{ABC}(\boldsymbol{\theta}|\boldsymbol{s}^{obs})$. As in Equation 6.2, our algorithm's final iteration involves sampling $\boldsymbol{\theta} \sim w_C(\boldsymbol{\theta})q_C(\boldsymbol{\theta})$ and $\boldsymbol{s}^* \sim \mathcal{M}(\cdot|\boldsymbol{\theta})$; we then accept the pair $(\boldsymbol{\theta}, \boldsymbol{s}^*)$ with a probability proportional to $\mathbb{1}_{\boldsymbol{d}(\boldsymbol{s}^{obs},\boldsymbol{s}^*)<h_C}$. Samples $\{\boldsymbol{\theta}^{(i)}\}$ from the ABC approximated posterior $\pi_{ABC}(\boldsymbol{\theta}|\boldsymbol{s}^{obs})$ were obtained by simply discarding the marginal values of $\boldsymbol{s}^*$ in $\{(\boldsymbol{\theta}^{(i)}, \boldsymbol{s}^{*(i)})\}$. There is no need to evaluate the likelihood function in our sampling. However, $\pi_{ABC}(\boldsymbol{\theta}|\boldsymbol{s}^{obs})$ can be also gained by marginalising over the $\boldsymbol{s}^*$, that is $\pi_{ABC}(\boldsymbol{\theta}|\boldsymbol{s}^{obs}) = \int \pi_{ABC}(\boldsymbol{\theta}, \boldsymbol{s}^*|\boldsymbol{s}^{obs})d\boldsymbol{s}^*$. Thus, an alternative estimator for $\pi(\boldsymbol{\theta}|\boldsymbol{s}^{obs})$ can be considered as

$$\hat{\pi}_{ABC}(\boldsymbol{\theta}|\boldsymbol{s}^{obs}) \propto w_C(\boldsymbol{\theta})q_C(\boldsymbol{\theta})\frac{1}{R}\sum_{r=1}^{R} \mathbb{1}_{\boldsymbol{d}(\boldsymbol{s}^{obs},\boldsymbol{s}_r^*)<h_C}, \tag{6.3}$$

where $\{\boldsymbol{s}_r^*\}_{1\leqslant r\leqslant R}$ denotes R independent draw from $\mathcal{M}(\boldsymbol{s}^*|\boldsymbol{\theta})$ for a fixed $\boldsymbol{\theta}$. Due to the unbiasedness (up to proportionality) of this new estimator, the resulting target distribution maintains consistency with the original (S. Sisson and Fan 2018). As such, $\hat{\pi}_{ABC}$ had been used as an alternative to $\pi_{ABC}(\boldsymbol{\theta}|\boldsymbol{s}^{obs})$ in ABC-PMC (Toni et al. 2009) and some other studies (Marjoram, Molitor, et al. 2003; S. A. Sisson, Fan, and Tanaka 2007). Additionally, by setting $w_C(\boldsymbol{\theta})q_C(\boldsymbol{\theta}) \propto \prod_{r=1}^{R} p(\boldsymbol{s}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})$ in Equation 6.3, S. A. Sisson and Fan 2010 conducted a comparison between π_{ABC} and $\hat{\pi}_{ABC}$ in a pseudo-marginal ABC MCMC algorithm, empirically demonstrating the enhancement of chain mixing as R grows (S. Sisson and Fan 2018).

Furthermore, the results of our case study seem also suggest us the need to draw $\boldsymbol{s}^* \sim \mathcal{M}(\boldsymbol{s}^*|\boldsymbol{\theta})$ more than once. After running the proposed ABC-PMC algorithm to generate 200 samples $\{\boldsymbol{\theta}^{(i)}\}_{1\leqslant r\leqslant 200}$ from $\pi_{ABC}(\boldsymbol{\theta}|\boldsymbol{s}^{obs})$, we then simulated 1,000 summary statistic vectors $\{\boldsymbol{s}_r^*\}_{1\leqslant r\leqslant 1000} \sim \mathcal{M}(\boldsymbol{s}^*|\boldsymbol{\theta})$ for each of the 200 samples. The medians of the 200,000 simulated summary statistic vectors, $\boldsymbol{m} = (m_1, ..., m_{25})^\top$, are displayed in the Figure B.7 in Appendix B.3 for comparison with the information we collected during the last iteration of the ABC-PMC sampling. While $\boldsymbol{m}$ falls somewhere between the medians of the simulated summary statistic vectors for only the accepted particles, $\boldsymbol{m}^{acc} = (m_1^{acc}, ..., m_{25}^{acc})^\top$, and that for all particles including rejected ones, $\boldsymbol{m}^{all} = (m_1^{all}, ..., m_{25}^{all})^\top$, across all 25 dimensions, $\boldsymbol{m}$ is only slightly smaller than $\boldsymbol{m}^{all}$ but much greater than $\boldsymbol{m}^{acc}$. This strongly suggests that, with the current ABC algorithm, the particles sampled from $\pi_{ABC}(\boldsymbol{\theta}, \boldsymbol{s}^*|\boldsymbol{s}^{obs})$ are not able to consistently reproduce the observed datasets. However, even for the best parameter values, it is rare to match a fixed target well for a high-dimensional simulation (Prangle et al. 2018). We, therefore, also tracked the frequency of simulated datasets meeting the acceptance threshold used in the final iteration of ABC-PMC across 1,000 simulations for each of $\{\boldsymbol{\theta}^{(i)}\}_{1\leqslant r\leqslant 200}$. The Pearson correlation coefficient between the total number of passes and the corresponding importance weight for $\{\boldsymbol{\theta}^{(i)}\}_{1\leqslant r\leqslant 200}$ was calculated to be

-0.0291, indicating no evidence for detectable correlation. These findings are somewhat expected since the weighting strategy in the current ABC algorithm is solely based on the perturbation kernel and does not consider 'how likely a particle can reproduce the observed dataset'. In this scenario, an estimator like $\hat{\pi}_{ABC}(\boldsymbol{\theta}|\boldsymbol{s}^{obs})$, which evaluates weights by drawing $\boldsymbol{s}^* \sim \mathcal{M}(\boldsymbol{s}^*|\boldsymbol{\theta})$ multiple times, appears more reasonable.

Nevertheless, estimators like $\hat{\pi}_{ABC}$ may exhibit greater variability in acceptances and weights, which can be strongly influenced by the number of draws, R (S. Sisson and Fan 2018). A larger R yields more accurate results but at the cost of increased computational overheads. Bornn et al. 2017 therefore explored the optimal R value for ABC rejection and MCMC algorithms based on $\hat{\pi}_{ABC}$. They found that, when using a uniform kernel $\mathbb{1}_{\boldsymbol{d}(\boldsymbol{s}^{obs},\boldsymbol{s}^*_r)<h_C}$, $R = 1$ is actually the most efficient. This is because, for simplified, the combination of R draws used to accept a single $\boldsymbol{\theta}$ could instead be used to accept up to R different $\boldsymbol{\theta}$, with each using just one draw. Simply increasing R, therefore, may not be an optimum solution,

2. Our study of IPMs is fundamentally dependent on population census datasets. In the context of these datasets, each conducted census, occurring at specific intervals, offers a distinct snapshot of the population along with its respective characteristics at that particular moment. Consequently, an IPM, which is based on this kind of time series data, outlines the evolution of a population over discrete time periods. In essence, both in our empirical observations and in the conceptualisation of IPMs, the population structure at a given time $t+1$ depends on its preceding time point t. As a result, the elements in $\boldsymbol{s}$ are likely to exhibit some degree of correlation with each other, given that they encompass just, in our case study of *C. Flava*, five unique types of summary statistics over a span of five years.

   To provide a clearer representation, let us redefine the elements in $\boldsymbol{s}$ in another way, $\boldsymbol{s} = (s_1, ..., s_{25})^\top = (s_1^1, s_1^2, ..., s_1^5, s_2^1, ..., s_2^5, ..., s_5^5)^\top = (\boldsymbol{s_1}^\top, \boldsymbol{s_2}^\top, ..., \boldsymbol{s_5}^\top)^\top =$

$\boldsymbol{s_{1:5}}$, where $\boldsymbol{s_i}$ represents the five different types of summary statistics for ith year. Therefore, ABC target in Equation 6.2 can be rewritten as

$$\pi_{ABC}(\boldsymbol{\theta}, \boldsymbol{s^*_{1:5}}|\boldsymbol{s^{obs}_{1:5}}) \propto \mathbb{1}_{\boldsymbol{d}(\boldsymbol{s^{obs}_{1:5}}, \boldsymbol{s^*_{1:5}})<h_C} w_C(\boldsymbol{\theta}) \mathcal{M}(\boldsymbol{s}^*|\boldsymbol{\theta}) q_C(\boldsymbol{\theta}). \tag{6.4}$$

ABC targets in Equation 6.2 and 6.4 are unable to capture the dependence among the summary statistics $\boldsymbol{s_{1:5}}$ within a probabilistic structure, as there is no specification available for conditional probability densities, such as $p_\theta(\boldsymbol{s_i}|\boldsymbol{s_{1:i-1}})$. Although we may not be primarily interested in inferring these probabilities for the years we have observed, retaining such a probabilistic structure in our study could be still highly beneficial. One significant advantage is that it would help alleviate the curse of dimensionality in ABC.

When models have a factorised likelihood, utilising conditional probability densities allows us to perform ABC in stages, sequentially sampling from the sequence of target posterior probability densities (Prangle 2018). That is, our ABC algorithm can be redesign to sample from an extended ABC approximation,

$$\tilde{\pi}_{ABC}(\boldsymbol{\theta}, \boldsymbol{s^*_{1:5}}|\boldsymbol{s^{obs}_{1:5}}) \propto (\prod_{i=1}^{5} \mathbb{1}_{\boldsymbol{d}(\boldsymbol{s^{obs}_i}, \boldsymbol{s^*_i})<h} \mathcal{M}(\boldsymbol{s^*_i}|\boldsymbol{s^*_{1:i-1}}, \boldsymbol{s^{obs}_{1:i-1}}, \boldsymbol{\theta}))\pi(\boldsymbol{\theta}). \tag{6.5}$$

This ABC approximation breaks down the global dependence ($\mathbb{1}_{\boldsymbol{d}(\boldsymbol{s^{obs}_{1:5}}, \boldsymbol{s^*_{1:5}})<h_C}$ in Equation 6.4) into a collection of smaller sub-problems (i.e. the factors $\mathbb{1}_{\boldsymbol{d}(\boldsymbol{s^{obs}_i}, \boldsymbol{s^*_i})<h}$), which can be computationally easier to handle (Jasra 2015). Accordingly, the summary statistics used for ABC can be chosen for each year, rather than requiring very high dimensional summaries of the entire model across all the years. We need a new ABC sampler with a more appropriate structure for time-series datasets to retain these dependence, and therefore reduce the number of summary statistics required to increase acceptance rates and sampling efficiency.

In a study by Jasra, Singh, et al. 2012, a sequential Monte Carlo (SMC) algorithm was used to fit and sample from ABC approximations for a Hidden Markov Model

with an intractable likelihood function, performing filtering inference. The authors also showed theoretically that the ABC approximation for filtering is biased but the bias approaches to zero as the bandwidth h goes to zero. Building on this work, Jasra 2015 generalised the results to a class of time-series models. For a fixed value of $\boldsymbol{\theta}$, they used SMC to sample from ABC approximations for the likelihoods of the observed time-series data, such as the one in Equation 6.5. They then, borrowing the idea of pseudo-marginal MCMC (Andrieu and G. O. Roberts 2009), incorporated the approximated likelihood into the MCMC algorithm to conduct posterior inference. A simplified replication of their methods in our study is shown in Algorithm 5 (based on Algorithm 4). It is worth noting that standard SMC can collapse when a small bandwidth h is used, so we used Alive SMC (Amrein and Künsch 2011; Jasra, Lee, et al. 2013) to address this issue.

Algorithm 5 utilise SMC in a manner similar to how it was originally proposed for time-series datasets. Unlike our current ABC-PMC, particles in each iteration in Algorithm 5 are one-step simulated datasets for a fixed $\boldsymbol{\theta}$. Only promising particles will be propagated to the next iteration to provide reasonable initial population structures for the one-step simulations in the next round. Employing ABC samplers like Algorithm 5 can perfectly address the issues we are concerned with. It allows us to retain the probabilistic structure and reduce the number of summary statistics required. It can also help us to more appropriately define the 'weight' for each proposed $\boldsymbol{\theta}$ based on 'how likely it can reproduce the observed dataset'. However, there are also two potential challenges that come with the algorithm.

- Firstly, the computational cost to apply Algorithm 5 may be prohibitively high. As mentioned before, S. Sisson and Fan 2018 found that, by working with pseudo-marginal MCMC, using the computational resources to accept different θ of up to R is more efficient than using the combination of R draws to accept a single θ. Here, instead of just R draws, Algorithm 5 utilise an entire SMC to evaluate a single θ. It is uncertain whether it would be worthwhile to switch from our current ABC-PMC algorithm to a computationally expensive

Algorithm 4: Alive SMC for $\tilde{\pi}_{ABC}$

Input : a specific value of hyperparameter $\boldsymbol{\theta}$; $\boldsymbol{s}_{\mathbf{1:5}}^{obs}$; a big integer $M > 0$, ensuring that there are M samples alive.

Output : an estimated likelihood $p_{h,M}(\boldsymbol{s}_{\mathbf{1:5}}^{obs}|\theta) \propto \prod_{i=1}^{5} \frac{M-1}{m_i-1}$

Sampling :

$\boldsymbol{c = 1}$ **:** initialisation

$k = 0$; $m_0 = 0$;
while $k < M$ **do**
  $m_0 \leftarrow m_0 + 1$;
  Simulate a dataset $y^* \sim \mathcal{M}(\cdot|\boldsymbol{\theta})$;
  Calculate simulated summary statistics $\boldsymbol{s}_{\mathbf{1}}^* = (s_1^{1*}, s_1^{2*}, ..., s_1^{5*})^\top$.
  if $\boldsymbol{d}(\boldsymbol{s}_{\mathbf{1}}^{obs}, \boldsymbol{s}_{\mathbf{1}}^*) < h$ **then**
    $k \leftarrow k + 1$;
    Accept the particle y^*, set $y_{k,1}^* = y^*$
end
$c \leftarrow c + 1$.

$\boldsymbol{c \geqslant 1}$ **:** for each simulation step $c = 2, ..., 5$,

$k = 0$; $m_c = 0$;
while $k < M$ **do**
  $m_c \leftarrow m_c + 1$;
  Sample a trajectory from $\{y_{k,c-1}^*\}_{1 \leqslant k \leqslant M-1}$ with probability $\frac{1}{M-1}$;
  Simulate a dataset $y^* \sim \mathcal{M}(\cdot|y_{k,c-1}^*, \boldsymbol{\theta})$ and calculate $\boldsymbol{s}_{\boldsymbol{c}}^* = (s_c^{1*}, s_c^{2*}, ..., s_c^{5*})^\top$.
  if $\boldsymbol{d}(\boldsymbol{s}_{\boldsymbol{c}}^{obs}, \boldsymbol{s}_{\boldsymbol{c}}^*) < h$ **then**
    $k \leftarrow k + 1$;
    Accept the particle y^*, record the trajectory $y_{k,c}^* = (y^*, y_{k,c-1}^*)$
end
$c \leftarrow c + 1$

Algorithm 5: Particle Marginal Metropolis-Hastings MCMC with Alive SMC

Input : a prior for the hyperparameter $\pi(\boldsymbol{\theta})$; a proposal distribution $\boldsymbol{\theta} \sim q(\cdot|\boldsymbol{\theta}')$;

Output : $\boldsymbol{\theta} \sim \tilde{\pi}_{ABC}$

Sampling :

Draw $\boldsymbol{\theta}' \sim q(\cdot|\boldsymbol{\theta})$;
Plug $\boldsymbol{\theta}'$ into Algorithm 4 to obtain an estimated likelihood $\tilde{p}_{h,M}(\boldsymbol{s}_{\mathbf{1:5}}^{obs}|\theta')$;
Accept $\boldsymbol{\theta}'$ with probability $1 \wedge \frac{\tilde{p}_{h,M}(\boldsymbol{s}_{\mathbf{1:5}}^{obs}|\theta')\pi(\theta')q(\theta|\theta')}{\tilde{p}_{h,M}(\boldsymbol{s}_{\mathbf{1:5}}^{obs}|\theta)\pi(\theta)q(\theta'|\theta)}$.

algorithm like Algorithm5. Whether the computational cost is worth the gain in accuracy remains to be determined.

- Secondly, finding an appropriate proposal distribution $q(\boldsymbol{\theta}|\boldsymbol{\theta}')$ for the 32 hyperparameters is not an easy task. In initial experiments with a multivariate normal perturbation kernel, the rejection rate was dramatically high ($> 99.9\%$). The inappropriate proposal movements generated unrealistic vital rates, resulting in simulated populations that does not resemble the observations. Given the experience of using current ABC-PMC, using a Gibbs-like step (Clarté et al. 2021) to update hyperparameters for a single vital rate at a time and keeping the rest fixed could be a potential solution. Alternatively, relying on MCMC samples for proposal moves may be a viable first step in applying Algorithm 5 to our study. We can directly compare the current ABC-PMC and Algorithm 5 with the only difference being the sampler itself, to determine if this 'seemingly' more appropriate sampling structure can produce a more reasonable inference result.

6.2.2 A new sampler with a less-restricted prior

Recently, a GP approximation method called PriorCVAE relying on deep generative models (Tomczak 2021; Bond-Taylor et al. 2021) was proposed by Semenova, Cairney-Leeming, et al. 2023, and could be potentially useful in our study. In general, random inputs coming from a basic probability distribution (for instance, a standard Normal distribution) are transformed — through a generative model in the form of a neural network with trainable parameters — into samples that align with the desired target probability distribution. That is, a non-linear transformation $\mathcal{T}$ is generated through deep generative modelling methods to approximate $f_{GP}|\theta$ in Equation 6.1 through $f_{PriorCVAE}|\theta$, where

$$f_{GP}|\theta \approx f_{PriorCVAE}|\theta = \mathcal{T}(z, \theta). \tag{6.6}$$

Once the non-linear transformation is well-trained, making inferences can be done instantly. Both the MCMC-based method and the PriorCVAE method rely on

pre-calculations to learn the characteristics of GPs and enable inference through 'drop-in'; however, they employ distinct approaches to achieve this goal.

Our MCMC-based method 'write' the characteristics down explicitly. Consequently, the posterior random variables generated by this method are limited to a countable number of distinct values. In contrast, the PriorCVAE method focuses on uncovering the patterns underlying the characteristics of the GPs, rather than providing explicit definitions. Therefore, this approach, in expectation, enables the random variables in the posterior to have an infinite number of possible values (within a reasonable range near to the training datasets).

Training neural networks in PriorCVAE would involve utilising a collection of representative samples from the target distribution, ultimately enabling the networks to learn the underneath probabilistic distribution instead of just a single representation from the target (Semenova, Xu, et al. 2022). To enable adaptability and accurately approximate GPs with different hyperparameter values in ABC sampler, the target distribution's hyperparameters (GP models in our case) should be encoded along with representative samples. The approximations generated by PriorCVAE have been shown to have reliable results in the case study of Semenova, Cairney-Leeming, et al. 2023. They have also demonstrated an application of PriorCVAE in extrapolation of hyperparameters. To explore the feasibility of employing PriorCVAE in our study, several challenges need to be addressed. The following points should be the focus of our investigation:

- **Training PriorCVAE and Evaluating its performance in high dimensional GP models:** The performance of PriorCVAE in high-dimensional GP models requires further assessments. The original article used a 2D hyperparameter GP model with one fixed dimension (resulting in an actual 1D GP) to evaluate PriorCVAE's performance. Our study, however, requires investigating how PriorCVAE performs when multiple hyperparameters are varied simultaneously. Moreover, to achieve a good GP approximation, we need to determine the optimal method for selecting the hyperparameter values for training PriorCVAE. For example, in their example, a simple 1-D grid-based

approach was used to select training hyperparameters. PriorCVAE in this case has demonstrated the capability to perform extrapolation about GP hyperparameters — maintaining high-quality reconstructed GPs if out-of-sample hyperparameter values are close to in-sample values, though deteriorating as the distance increases. However, it remains uncertain whether a simple grid-based approach for hyperparameters can be employed in high-dimensional scenarios to train PriorCVAE while maintaining satisfactory performance in extrapolation. Since hyperparameters in GP models typically have different scales, and the effects on GP predictions are not linearly correlated with GP hyperparameter perturbations (as discussed in Section 2.2.6), we must consider whether standard Euclidean distance is adequate for defining grid distances or if alternative distance metrics should be explored. We need to establish a more appropriate method for configuring the grid network.

- **Assessing PriorCVAE as a Replacement for true GP:** A more formal or systematic approach is needed to assess whether a trained PriorCVAE is adequate to replace the true GP. This involves considering the potential consequences and precautions needed if our sampler proposes particles that lie outside the well-trained area.

- **Costs of using PriorCVAE:** Our goal is to achieve a high-quality GP approximation while ensuring that the total implementation time for PriorCVAE and the subsequent sampling process remains reasonably manageable. Implementing PriorCVAE comes with significant costs, particularly the substantial upfront expense of training. Once trained, PriorCVAE can be applied to a wide range of situations, with the Bayesian inference MCMC step taking seconds or minutes (Semenova, Cairney-Leeming, et al. 2023). However, the initial training cost may not be easily affordable. As a reference (though not directly comparable), training a similar method, πVAE, may take days or weeks and typically involves heuristic and parameter searches common in applied deep learning for optimal performance (Mishra et al. 2022). Training

time for PriorCVAE appears to be even longer, as additional work is required to train the CVAE based on different hyperparameter values. The upfront cost is closely linked to the two previous bullet points: the number of training points we choose to use and the desired level of approximation quality. Finding the right balance between these factors is crucial.

One of the characteristics of PriorCVAE is that the values in covariates for predictions must be identical to that for the training points. For example, if a fixed grid (for covariates) was used to train the CVAE, predicting off-grid points would not be possible. However, this limitation should not pose a significant concern in our study. We can manually discretize the continuous state variables, and train PriorCVAE by taking all possible discretized state values into account. The IPM will ultimately be discretized into large matrices capable of accurately representing the IPM kernel's topography. To some extent, the discretization would implicitly push the result matrices used for demographic analysis away from the originally proposed IPM kernel. Incorporating discretization earlier in the model-fitting step might help reduce the discrepancy. Nevertheless, this characteristic may also give rise to other drawbacks. As in Equation 6.6, when a standard normal distribution-drawn z is fed into the trained deterministic transformation $\mathcal{T}$, it generates an entire vector $f_{PriorCVAE}|\theta$ with the same length as the covariate values used for training, i.e. all possible discretized state in our case, even if we are only interested in a few grid points. This inevitably results in wasted resources for additional forecasts where they might not be needed. The waste could be substantial when a large number of simulations are required, as in the case of our ABC samplers.

Appendices

A

Appendix for Chapter 2

Plot	Kernel	nll	n_s	Scale r^2	ARD	
					Size	Age
(a)	Gau	54.02	0.3243	8.533	$l_1 = 7.098$	$l_2 = 18.31$
(b)	Gau	53.94	0.3243	8.533	$l_1 = 6.098$	$l_2 = 18.31$
(c)	Gau	54.52	0.2952	4.480	$l_1 = 2.233$	$l_2 = 5.531$
(d)	Gau	56.51	0.2952	4.480	$l_1 = 1.233$	$l_2 = 5.531$

Table A.1: The values of the hyperparameters of two GP regression models in Figure 2.2 and Figure 2.3. Models producing Plot (b) and (d) were produced by perturbing the lengthscale for the log size with the same absolute amount, -1, on (a) and (c) respectively.

Plot	Kernel	nll	n_s	Scale r^2	ARD	
					Size	Age
(a)	Gau	54.02	0.3243	8.533	$l_1 = 7.098$	$l_2 = 18.31$
(b)	Gau	53.94	0.3243	8.533	$l_1 = 5.323$	$l_2 = 18.31$
(c)	Gau	54.52	0.2952	4.480	$l_1 = 2.233$	$l_2 = 5.531$
(d)	Gau	55.20	0.2952	4.480	$l_1 = 1.675$	$l_2 = 5.531$

Table A.2: The values of the hyperparameters of two GP regression models in Figure A.1 and Figure A.2. Models producing Plot (b) and (d) were produced by perturbing the lengthscale for the log size with the same proportional amount, -25%, on (a) and (c) respectively.

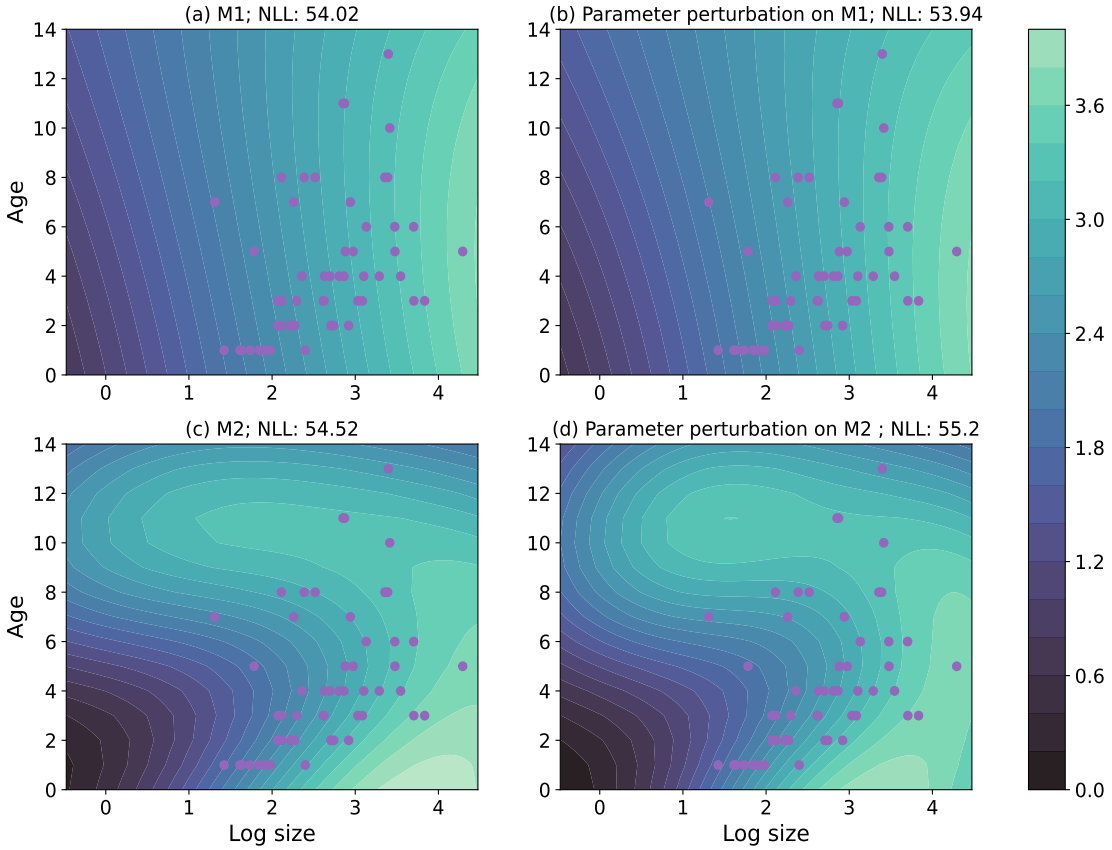


Figure A.1: Given the current age (y-axis) logged size (x-axis), the contour plots represent the predicted mean for logged size in the next year. Predictions were made through Gaussian process models with hyperparameters shown in Table A.2 in Appendix A. Plot (b) and (d) were produced by perturbing the same hyperparameter with the same **proportional** amount in (a) and (c) respectively. Purple dots represent for the values of covariance for the training data points.

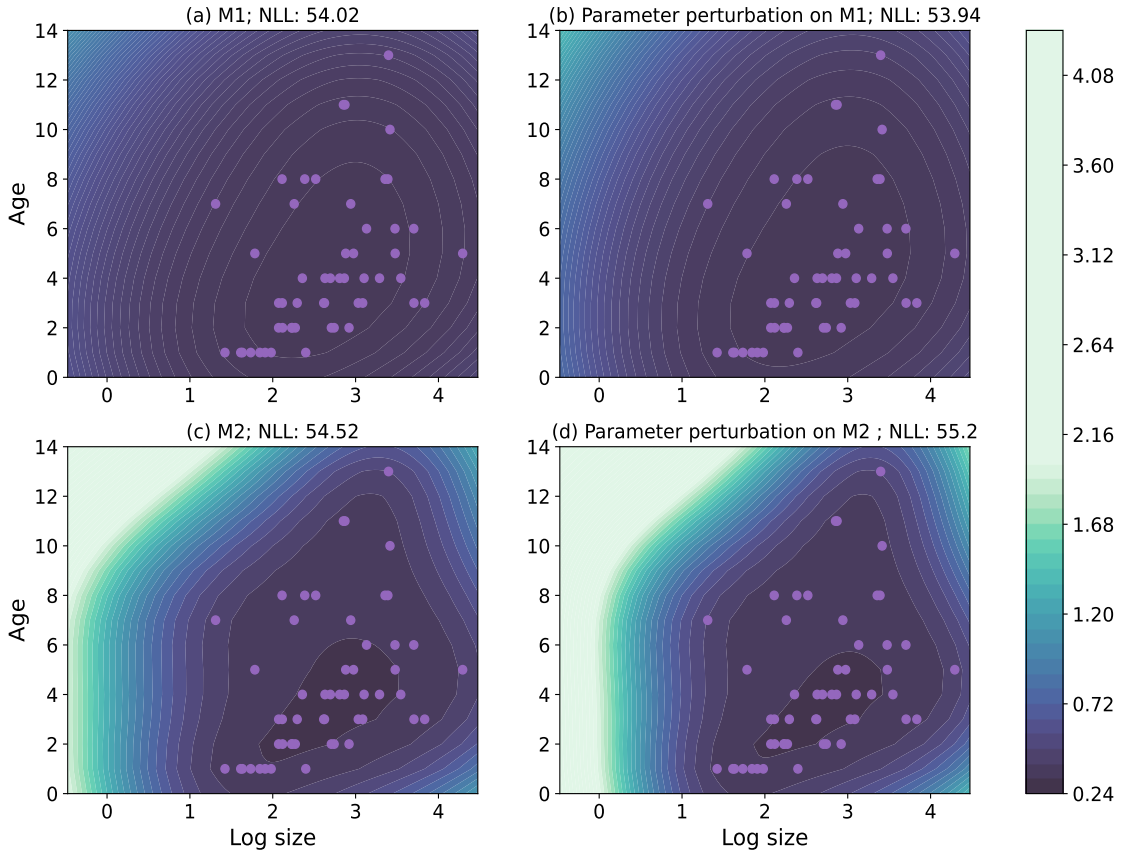


Figure A.2: Contour plots for the predicted variance corresponding to the predicted mean shown in Figure A.1.

Example:

Suppose we have a vector representing current age, denoted as $\boldsymbol{a} = (a_1, ..., a_{13})$ with $a_1 < a_2 < ... < a_{13}$, and a vector of current logarithmic size, denoted as $\boldsymbol{z} = (z_1, ..., z_{14})$ with $z_1 < z_2 < ... < z_{14}$. To create a list of 2-dimensional inputs, we use the `meshgrid` function to obtain pairs of $\boldsymbol{a}$ and $\boldsymbol{z}$. Specifically, this function produce a set of 182 inputs $\{\boldsymbol{x_i}\}_{i=0,...,181} = \{(x_{1,i}, x_{2,i})^\top\}_{i=0,...,181} = \{(z_1, a_1)^\top, (z_1, a_2)^\top, ..., (z_1, a_{13})^\top, (z_2, a_1)^\top, ..., (z_{14}, a_{12})^\top, (z_{14}, a_{13})^\top\}$. Please note that, the input elements of $\{\boldsymbol{x_i}\}_{i=0,...,181}$ are intentionally ordered. This arrangement ensures a periodic variation in their Euclidean distances, and help us easier to virtualise systematic patterns in their covariances.

Let us consider the effect of decreasing l_z (the lengthscale for $\boldsymbol{z}$) while keeping other parameters unchanged. Specifically, we plug $\{\boldsymbol{x_i}\}_{i=0,...,181}$ into the Gaussian kernels used by **M1**, **M2** and their perturbed counterparts, respectively, to compute

the covariances between the data points $\boldsymbol{x_0}$ and the others, $\{\boldsymbol{x_j}\}_{j=1,\ldots,181}$. The results, shown in Figure A.3, align with our earlier discussion.

- **From the perspective for the dimension of age:** reducing l_z decreases the covariance between x_0 and the other points, regardless of their distance. This effect is evident in the pairs of waves in Figure A.3 left, where one such pair is highlighted.
- **From the aspect for the dimension of z:** as shown by the peaks of the waves in Figure A.3 right, when $\boldsymbol{a}$ (age) is fixed, points $\{\boldsymbol{x_j}\}_{j=1,\ldots,181}$ with x_j^2 that are far from x_0^2 become less correlated with $\boldsymbol{x_0}$. In this case, points closer to x_0^2 thus have a relatively higher impact on it.

From Figure A.3, we can see that the perturbation affects both models in a similar way. However, the differences in hyperparameter values between **M1** and **M2** amplify or diminish the impact to varying degrees. This highlights that, in GP models, the effect of perturbing a single hyperparameter depends heavily on the values of the other hyperparameters. Furthermore, from Figure A.3, it seems the elasticity analysis (proportional perturbation) is more suitable for GP models, but, again, this still depends on the values of other hyperparameters.

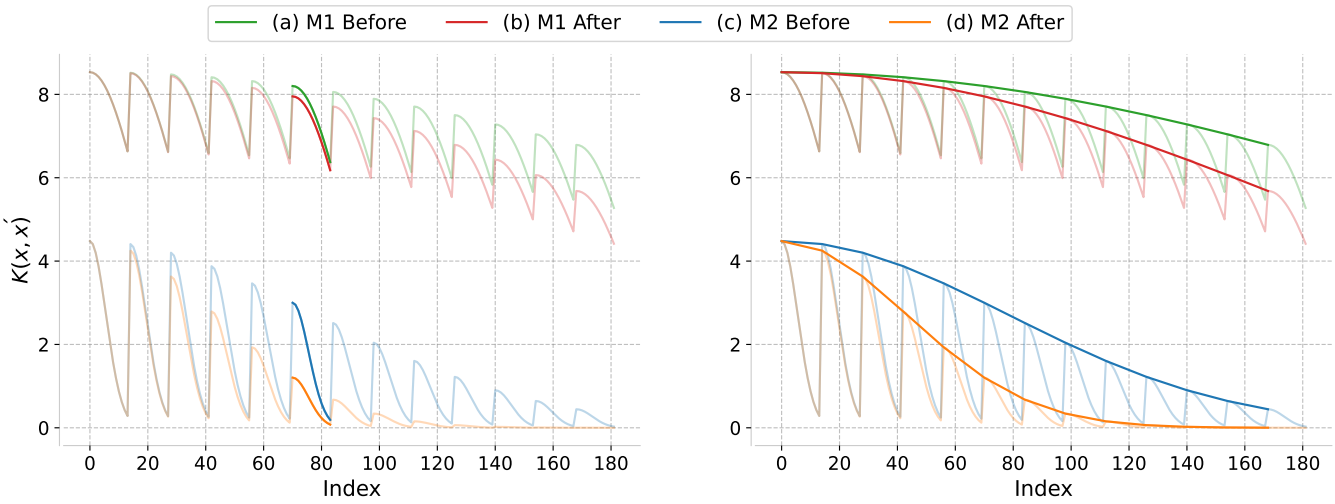


Figure A.3: The GP kernel values for data points $(\boldsymbol{x_0}, \boldsymbol{x_j})$ for $j = 1, ..., 181$. The calculations were performed using Gaussian kernel with four different sets of hyperparameters, which were used in the models that generated the prediction surface depicted in Figure 2.2. The green and red lines represent the kernel values computed before and after perturbation, respectively, for model **M1**, while the blue and orange lines represent those for model **M2**. The set of values $\{\boldsymbol{x_i}\}_{i=0,...,181}$ was obtained through the meshgrid function, resulting in periodic patterns in the plots. The two plots are essentially the same, except that different regions have been highlighted. Notice that, this figure only illustrates how perturbations influence the GP covariance kernels. These changes in covariance also impact the predictive distributions. However, the prediction surface's response depends on the specific dataset. As a non-parametric method, GP models rely on the positions of the data points to evaluate these effects. Using the conditional density formula for a multivariate normal (MVN) distribution, one may calculate how covariance changes influence predictions in GP regression models.

B

Appendix for Chapter 4

B.1 More details about the Simulation case study

B.1.1 A full description for candidate summary statistics and distance metrics

To calculate the majority of candidate summary statistics under consideration, individuals need to be grouped first according to their size. Individuals can be grouped by their current size or by their size in the next year (denoted by sizeNext). Twelve size/sizeNext groups are classified based on their logarithm size/sizeNext (bins: $[0, 0.3, 0.6, 0.9, 1.2, 1.5, 1.8, 2.1, 2.5, 3, 3.5, 4, > 4]$). For most of the candidate summary statistics, meanings behind them have already been fully represented by their names. However, we would like to emphasise the following summary statistics that might be slightly confusing:

- **The empirical population distribution in sizeNext groups:** the counts of individuals in the corresponding groups divided by the total number of individual in the entire population.
- **Average number of flowering stalks/breeders/survivors in size groups:** the counts of flowering stalks/breeders/survivors in the corresponding groups divided by the total number of individuals in the corresponding group.

- **breeders'/non-breeders' sizeNext distribution:** the counts of breeders/non-breeders in the corresponding groups divided by the total number of individual in the entire population.

- **breeders' and non-breeders' survival distribution:** individuals are divided into four groups based on whether they were breeders in the past year and whether they survived in the past year.

Some of the summary statistics are constructed using frequencies, while others are based on distributions. Accordingly, we employed different distance metrics based on their attributes. The complete list of employed distance metrics can be found below. Let P and Q denote two (empirical) discrete distributions, p_i and q_i are the ith element of distributions P and Q, respectively. Let U and V denote two sets of values with ith element denoted by u_i and v_i respectively.

1. **Bhattacharyya distance:** is a measure of similarity between two probability distributions. For discrete probability distributions, the distance is defined as $D_B(P,Q) = -\ln(\sum_i \sqrt{p_i q_i})$.

2. **Hellinger distance:** is another measure of similarity between two probability distributions. It is related to the Bhattacharyya distance and is defined as $D_H(P,Q) = \frac{1}{\sqrt{2}}\sqrt{\sum_i(\sqrt{p_i} - \sqrt{q_i})^2}$.

3. **Earth Mover's Distance (EMD):** (also known as the Wasserstein distance or Kantorovich-Rubinstein norm) is a measure of the distance between two probability distributions, interpreted as the minimum amount of "work" necessary needed to transform one distribution into the other. The EMD is defined as: $D_{EMD}(P,Q) = \min_W \sum_{i,j} W_{ij} d_{ij}$ where, W is a flow matrix representing the optimal transportation plan, and d_ij is the difference between the elements i and j of the distributions. Here, "work" is calculated by multiplying the amount of weight that must be relocated by the distance it must travel.

4. **Kullback-Leibler (K-L) distance:** (also known as relative entropy) is a measure of how one probability distribution is different from a second, reference probability distribution. It is defined as $D_{KL}(P||Q) = \sum_i p_i \ln(p_i/q_i)$, for $p_i, q_i > 0$.

5. **Hilbert projective metric:** a distance function defined on the positive cone of a real Hilbert space. It defined as $D_H(P,Q) = \log(\max_i(p_i/q_i)) + \log(\max_i(q_i/p_i))$, for $p_i, q_i > 0$.

6. **Chi-square (χ^2) distance metric:** The chi-square distance is a measure of dissimilarity between two sets of values and is often used in comparing histograms. It is defined as: $\chi^2(U,V) = \sum_i (u_i - v_i)^2/(v_i)$.

7. **Sum of squared differences (SSD):** a common metric used to measure the dissimilarity between two sets of values. It is defined as: $SSD(U,V) = \sum_i (u_i - v_i)^2$.

8. **Cosine similarities:** a common metric used to measure the dissimilarity between two vectors. It is defined as: cosine similarity$(U,V) = \frac{U \cdot V}{\|U\|\|V\|}$. Here, $U \cdot V$ is the dot product of vectors. $\|U\|$ and $\|V\|$ are the magnitudes (or lengths) of the vectors. The cosine similarity ranges from -1 to 1, where 1 indicates that the vectors are identical, -1 indicates that they are completely opposite, and 0 indicates orthogonality (no similarity).

All of the calculations were implemented by self-developed code in `Python`, with the exception of EMD which was computed using the `wasserstein_distance` function from package `scipy`. Both the K-L distance and the Hilbert projective metric require that p_i and q_i be greater than 0. Our functions for computing these two types of distances were therefore modified to detect if $p_i < 0$ or $q_i < 0$ occurred. If either condition occurred, the functions would automatically merge the neighbouring cells in both P and Q to ensure comparability, until $p_i > 0$ and $q_i > 0$ for all i. A similar strategy was also applied to the χ^2 distance metric to ensure that the frequencies in the expected values are greater than 5.

B.1.2 Implementation details and more results

We initiated our experiment by applying GP models and GLMs within a frequentist framework to fit the real-world population data (aggregated from the three years from 2008 to 2011). In this simulation case study, we ignored the age dependence and environmental stochasticity. The state variable z was defined solely as log-transformed individual size, and z' represents the state in the following year.

In our implementation, each vital rate is treated as a separate sub-model (so 5 sub-models in total). These sub-models are fitted independently using either GLMs or GPs. Specifically, we model:

- survival probabilities $s(z)$ and reproduction probabilities $p_f(z)$, using GLMs with logit links and GPs with Bernoulli likelihoods;
- numbers of flowering stalks given reproduction $n_f(z)$, using Poisson GLMs with log links and GPs with Poisson likelihoods;
- growth of breeders $G_r(z'|z)$ and non-breeders $G_n(z'|z)$, using LMs and GP regressions.

Each of these components is modelled as a function of z (or z'), and fitted separately using the training data. These sub-models are then combined into the IPM transition function $\mathcal{T}(z'|z) = P(z'|z) + F(z'|z)$, where $F(z'|z) = p_f(z)n_f(z)p_{est}r_{size}(z')$ and $P(z'|z) = s(z)p_f(z)G_r(z'|z) + s(z)(1 - p_f(z))G_n(z'|z)$ (see caption of Table 4.1 for details on p_{est} and $r_{size}(z')$).

To implement the IPM numerically, the continuous transition kernel $\mathcal{T}(z' \mid z)$ usually be discretized into a finite-dimensional matrix. This is done by evaluating $\mathcal{T}$ on a grid of mesh points that span the range of the state variable z. To discretize IPM transition kernels $\mathcal{T}$, we employed 50 mesh points to describe the distribution of individual traits, following the default settings in the R package `IPMpack` (C. J. E. Metcalf et al. 2013). The lower and upper limits were fixed at 0 and 5 to encompass and slightly exceed the observed log-transformed sizes of the plants (Rees, Childs, and Ellner 2014). To address eviction - where individuals

with predicted growth probabilities exceed the limits of the IPM matrices are "accidentally" eliminated - we implemented a 'constant' style correction (J. L. Williams et al. 2012; C. J. E. Metcalf et al. 2013).

For each of GP models in this experiment, the mean function was set to zero (as in common practices), and the most widely used kernel function, the Squared Exponential (Gaussian), was applied (C. K. Williams and Rasmussen 2006). Bayesian individual vital rate model fitting was conducted using non-informative priors and Hamiltonian Monte Carlo methods (Hensman et al. 2015). Basic MCMC sample convergence and mixing were assessed through visual inspection during development. Trace plots are not included here, as the primary role of MCMC in our framework is to define a broad support set of hyperparameter configurations for use in ABC, rather than to characterise full posterior distributions. A summary of model-specific sampling settings is provided in Table B.1.

Dataset $\boldsymbol{D}_{\mathbf{glm}}$			
Sub-model	Priors for all parameters	Burn-in	Samples
$p_f(z)$	HalfNormal(scale=100)	5000	5000
$n_f(z)$	HalfNormal(scale=100)	5000	5000
$G_r(z'\|z)$	HalfNormal(scale=100)	10000	5000
$G_n(z'\|z)$	HalfNormal(scale=100)	10000	5000
$s(z)$	HalfNormal(scale=50)	10000	5000
Dataset $\boldsymbol{D}_{\mathbf{gp}}$			
Sub-model	Priors	Burn-in	Samples
$p_f(z)$	HalfNormal(scale=100)	7000	5000
$n_f(z)$	HalfNormal(scale=100)	5000	5000
$G_r(z'\|z)$	HalfNormal(scale=100)	3000	5000
$G_n(z'\|z)$	HalfNormal(scale=100)	5000	5000
$s(z)$	HalfNormal(scale=50)	5000	5000

Table B.1: MCMC configuration for individual GP models.

Frequentist individual model fitting for GP regressions can be computed directly, while that for GP models with non-Gaussian likelihood, Variational inference (Opper and Archambeau 2009) was employed. All the standard GP and GLM model fitting tasks were implemented using the `GPflow`, `statsmodels` and `Bambi` package in

	Selected Summary statistics
D_{glm}	(1) SSD for the total number of breeders in size groups; (2) SSD for the average number of flowering stalks in size groups; (3) the χ^2 distance in total number of individuals in sizeNext groups; (4) the Bhattacharyya distance in non-breeders' sizeNext distribution; (5) SSD for the total number of survivors in size groups.
D_{gp}	(1) the χ^2 distance for the total number of breeders in size groups; (2) SSD for the average number of flowering stalks in size groups; (3) the EMD in breeders' sizeNext distribution; (4) the Hilbert projective metrice in breeders' and non-breeders' survival distribution; (5) SSD for the total number of survivors in size groups.

Table B.2: The list of the Selected Summary statistics employed when applying ABC-PMC to ABC_GP for datasets D_{glm} and D_{gp}.

the `Python` programming language (Van Rossum and Drake Jr 1995; Seabold and Perktold 2010; Matthews et al. 2017; Capretto et al. 2022). Workflow 3.1 was executed for each vital rate model, ensuring that there was at least one summary statistic that performed well for each vital rate. We retained only the most robust summary statistics after implementing Workflow 3.1, resulting in five different types of summary statistics for each datasets (see Table B.2 for details).

When implementing ABC sampler, for D_{gp}, $\{\alpha_i\}_{0\leqslant i\leqslant C-1} = \{0.55, 0.5, 0.45, 0.4, 0.35, 0.3, 0.2, 0.15, 0.1, 0.088\}$; for D_{glm}, $\{\alpha_i\}_{0\leqslant i\leqslant C-1} = \{0.55, 0.5, 0.45, 0.4, 0.35, 0.3, 0.2\}$. For D_{glm}, the sampler stopped early because no obvious difference was found in the posterior marginal distributions (see Figure B.2).

Measure	**GLM**	**GP**	**ABC_GP**
MSE in λ	0.002757	0.001899	0.000547
Bias in λ	0.04122	0.03356	-0.00033
Mean KL in $\bar{\boldsymbol{\omega}}$	0.0104	0.0079	0.0072
Mean KL in $\bar{\boldsymbol{\omega}}_{\boldsymbol{rep}}$	0.0161	0.0128	0.0119
Mean Cosine similarity in γ	0.9980	0.9954	0.9953

Table B.3: Predictive performance evaluations for the 3 model fitting methods for dataset D_{glm}.

Measure	**GLM**	**GP**	**ABC_GP**
MSE in λ	0.003129	0.001626	0.000623
Bias in λ	0.034505	0.019512	0.009555
Mean KL in $\bar{\boldsymbol{\omega}}$	0.0097	0.0037	0.0018
Mean KL in $\bar{\boldsymbol{\omega}}_{\boldsymbol{rep}}$	0.0249	0.0140	0.0122
Mean Cosine similarity in γ	0.9764	0.9947	0.9943

Table B.4: Predictive performance evaluations for the 3 model fitting methods for dataset D_{gp}.

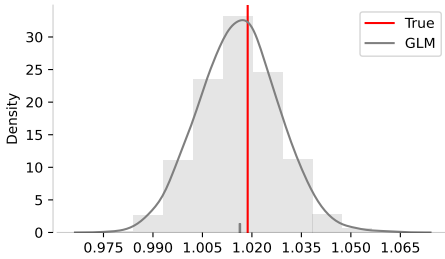


Figure B.1: GLM based IPM predictions with a larger sized training dataset, D_{glm2}. Predicted λs, along with their corresponding density histograms (shown as shaded areas), were generated using 5,000 samples draw from the estimated posterior distribution of GLMs. The red line represents the λ computed from the true discretized IPM. The pipe ('|') in the bottom of each plot marks the corresponding mean.

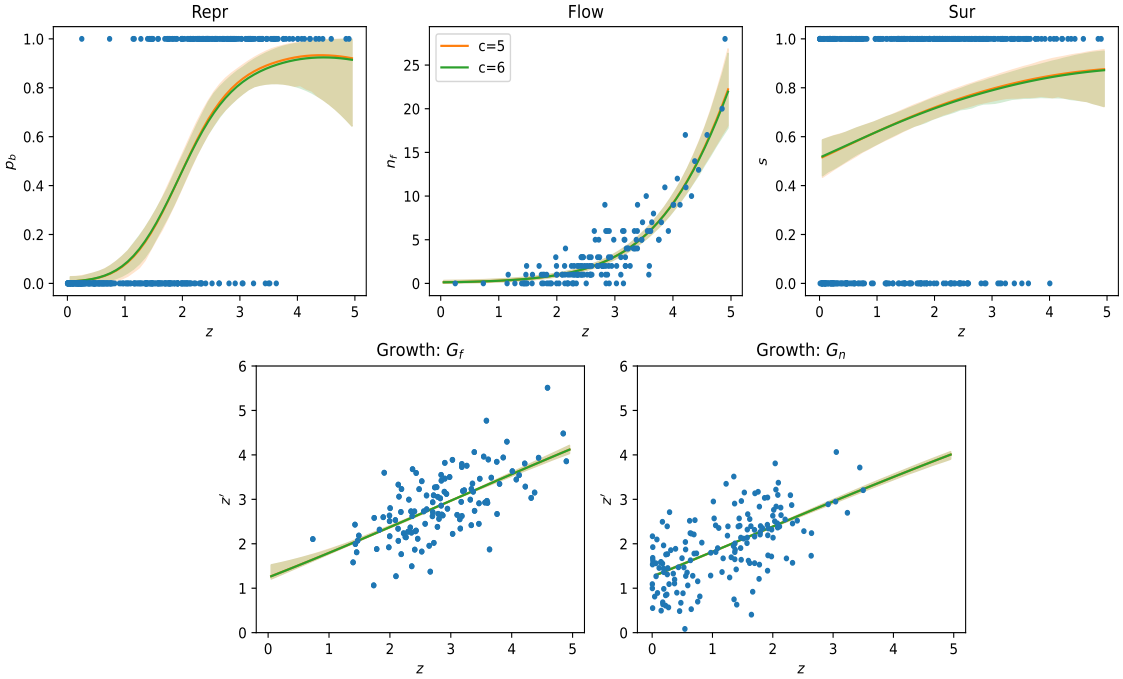


Figure B.2: Model fitting results for D_{glm} at simulation step 5 and 6. The estimated marginal distributions for the five vital rates, complete with their 95% confidence intervals in matching colours and the corresponding scatters data points in blue.

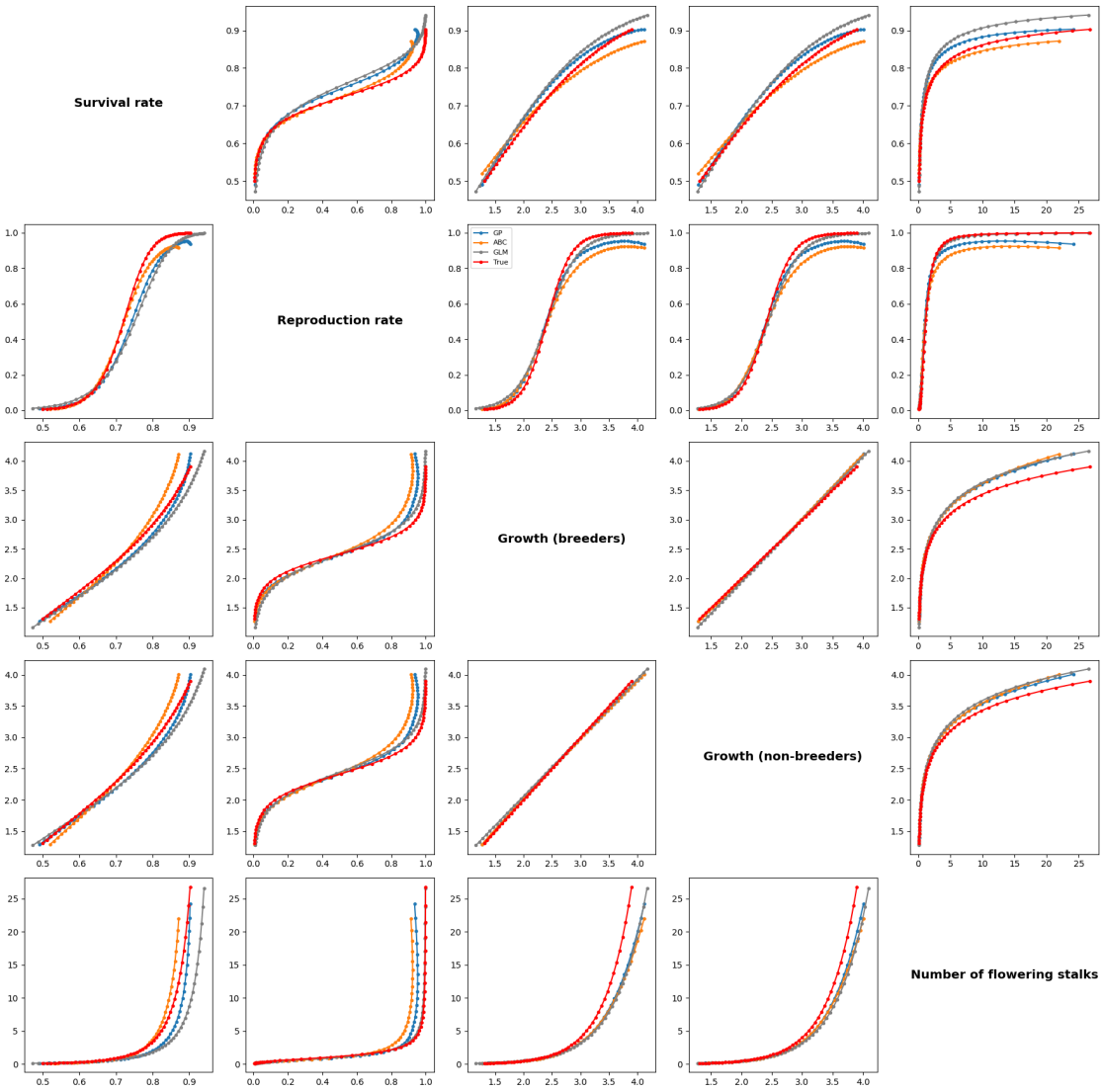


Figure B.3: Using the 5,000 samples that generated predicted λ in Figure 4.3 left, we extracted the corresponding vital rate models and evaluated them on 50 mesh points (representing size from 0 to 5). Sample means of vital rates at each mesh point, shown for GLM, GP, ABC-GP, and the true model.

B.2 More details about the Real case study

B.2.1 Candidate summary statistics

Summary statistics	Distance measures
The empirical population distribution in sizeNext, ageNext, sizeNext-ageNext groups	Bhattacharyya distance; Hellinger distance; EMD; K-L distance; Hilbert projective metric
Total number of individuals in sizeNext, ageNext, sizeNext-ageNext groups; Total number of flowering stalks in size, age, size-age groups; Total number of breeders in size, age, size-age groups; Total number of survivors in size, age, size-age groups;	χ^2 distance metric; Sum of squared differences
Average number of flowering stalks in size, age, size-age groups; Average number of breeders in size, age, size-age groups; Average number of survivors in size, age, size-age groups;	Sum of squared differences
breeders' sizeNext and ageNext distribution; non-breeders' sizeNext and ageNext distribution; breeders' and non-breeders' survival distribution	Bhattacharyya distance; Hellinger distance; EMD; K-L distance; Hilbert projective metric

Table B.5: The list of candidates about the summary statistics and the corresponding distance measures. To calculate the majority of candidate summary statistics under consideration, individuals need to be grouped first according to their size, age, or both. sizeNext and ageNext represent the individuals' size and age in the next year.

To calculate the majority of candidate summary statistics under consideration, individuals need to be grouped first according to their size, age, or both. Individuals can be grouped by their current size or age, and can also be categorised by their size or age in the next year (denoted by sizeNext and ageNext respectively). Five size/sizeNext groups are classified using the floor function based on their logarithm size/sizeNext (from 0 to 4). There are also four categories for age/ageNext: age/ageNext = 1 represents newborns; age/ageNext $\in [2, 5]$ for young individuals; age/ageNext $\in [6, 10]$ for middle-aged individuals; > 10 for white-bearded individuals. With this grouping strategy, for example, the candidate summary statistic 'the empirical population distribution in sizeNext-ageNext groups' produces 20 data points. For most of the candidate summary statistics, meanings behind them have already

been fully represented by their names. However, we would like to emphasise the following summary statistics that might be slightly confusing:

- **The empirical population distribution in sizeNext, ageNext, sizeNext-ageNext groups:** the counts of individuals in the corresponding groups divided by the total number of individual in the entire population.
- **Average number of flowering stalks/breeders/survivors in size, age, size-age groups:** the counts of flowering stalks/breeders/survivors in the corresponding groups divided by the total number of individuals in the corresponding group.
- **breeders'/non-breeders' sizeNext and ageNext distribution:** the counts of

  breeders/non-breeders in the corresponding groups divided by the total number of individual in the entire population.
- **breeders' and non-breeders' survival distribution:** individuals are divided into four groups based on whether they were breeders in the past year and whether they survived in the past year.

B.2.2 Implementation Details

Specifically, in IPMs, when z represents age and size and sub-kernel P and F depend on a vector of environmental factors $\boldsymbol{\omega}$, our IPM describes the population of *C. flava* by

$$n_0(x', t+1) = \sum_{a=0}^{M} \int_L^U F_a(x'|x, \boldsymbol{\omega}) n_a(x, t) dx$$
$$n_a(x', t+1) = \int_L^U P_{a-1}(x'|x, \boldsymbol{\omega}) n_{a-1}(x, t) dx,$$

for age $a = 1, 2, ..., M-1$ and size $x \in [L, U]$. $P_{a-1}(x'|x, \boldsymbol{\omega})$ denotes the probability that an individual with age $a-1$ and size x at current time could survive and grow to $[x', x'+dx]$ in the next year, with weather conditions $\boldsymbol{\omega}$. Similarly, $F_a(x'|x, \boldsymbol{\omega})$ describes the expected number of offspring in $[x', x'+dx]$ at the next time, per

size-x individual at current time, with weather conditions $\boldsymbol{\omega}$. M is the age class for 'greybeard', speaking for individuals with age M or older. This requires an addition formula for the model:

$$n_M(x', t+1) = \int_L^U P_{M-1}(x'|x, \boldsymbol{\omega}) n_{M-1}(x, t) + P_M(x'|x, \boldsymbol{\omega}) n_M(x, t) dx.$$

The sequence of life cycle events in the proposed IPMs align with the actual census process conducted in Salguero-Gomez et al. 2012, reflecting realities more properly. Specifically, with z represents size and age, we defined the reproduction kernel as $F(z'|z, \boldsymbol{\omega}) = p_f(z|\boldsymbol{\omega}) n_f(z|\boldsymbol{\omega}) r_{est|\boldsymbol{\omega}} r_{size}(z'|\boldsymbol{\omega})$ and the survival-growth kernel as $P(z'|z, \boldsymbol{\omega}) = s(z|\boldsymbol{\omega}) p_f(z|\boldsymbol{\omega}) G_f(z'|z, \boldsymbol{\omega}) + s(z|\boldsymbol{\omega})(1 - p_f(z|\boldsymbol{\omega})) G_{nf}(z'|z, \boldsymbol{\omega})$ (see detailed explanations in Section 2.1).

Each vital rate was modelled separately as a function of age, size, and environmental covariates. Specifically:

- survival probabilities $s(z|\boldsymbol{\omega})$ and reproduction probabilities $p_f(z|\boldsymbol{\omega})$, using GLMs with logit links and GPs with Bernoulli likelihoods;
- numbers of flowering stalks given reproduction $n_f(z|\boldsymbol{\omega})$, using Poisson GLMs with log links and GPs with Poisson likelihoods;
- growth of breeders $G_f(z'|z, \boldsymbol{\omega})$ and non-breeders $G_{nf}(z'|z, \boldsymbol{\omega})$, using LMs and GP regressions.

For each of the GP models, the mean function was set to zero, and the most commonly used kernel function, the Squared Exponential, was applied. The Squared Exponential kernel function was employed with the automatic relevance determination setting, allowing it to separately modulate the magnitude of similarity in each input coordinate (e.g. age, size, and various climate factors). Bayesian individual vital rate model fitting was conducted using non-informative priors and Hamiltonian Monte Carlo methods (Hensman et al. 2015). Basic MCMC sample convergence and mixing were assessed through visual inspection during development. Trace plots are not included here, as the primary role of MCMC in our framework

is to define a broad support set of hyperparameter configurations for use in ABC, rather than to characterise full posterior distributions. A summary of model-specific sampling settings is provided in Table B.6.

Sub-model	Priors for all parameters	Burn-in	Samples
$p_f(z)$	HalfNormal(scale=100)	1000	5000
$n_f(z)$	HalfNormal(scale=100)	30000	5000
$G_f(z'\|z)$	HalfNormal(scale=100)	150000	5000
$G_{nf}(z'\|z)$	HalfNormal(scale=100)	150000	5000
$s(z)$	HalfNormal(scale=100)	50000	5000

Table B.6: MCMC configuration for individual GP models.

Frequentist individual model fitting for GP regressions can be computed directly, while that for GP models with non-Gaussian likelihood, Variational inference (Opper and Archambeau 2009) was employed. All the standard GP model fitting tasks were implemented using the `GPflow` package in the `Python` programming language (Van Rossum and Drake Jr 1995; Matthews et al. 2017).

The model training process used information from the first five years (2003 to 2008) with 1075 individuals in total, while the data from the last four years were be reserved for model testing. With the dataset's maximum age being 7 in 2003-2004, we set the absorbing age, M, to 8. Individual size, the total number of rosettes, was log-transformed to make it to be less-skewed (in data exploration session; not shown).

In this real-world case study, recognising that directly accounting for all environmental factors could introduce great complexity and computational burden, we did not apply Workflow 3.1 to the entire dataset to select summary statistics. Instead, we focused exclusively on the final year (2007-2008) of the training datasets to minimise the influence of fluctuating environmental conditions across multiple years. The final year was selected because the age information was most complete compared to earlier years. MLEs of the GP models were estimated for the vital rates derived from the 2007-2008 dataset. A simulated dataset, based on the fitted GP models, was then generated and used as the "true" dataset within

Workflow 3.1, allowing for a controlled assessment of model performance. After this, Workflow 3.1 was executed as usual.

Workflow 3.1 was executed for each vital rate model, ensuring that there was at least one summary statistic that performed well for each vital rate. We retained only the most robust summary statistics after implementing Workflow 3.1, resulting in five different types of summary statistics. These are: (1) the sum of squared differences for the total number of breeders in size groups; (2) the sum of squared differences for the average number of flowering stalks in size groups; (3) the Hilbert projective metric in breeders' sizeNext distribution; (4) the EMD in non-breeders' sizeNext distribution; and (5) the sum of squared differences for the average number of survivors in size groups. We obtained observed datasets for 5 years, yielding a total of 25 summary statistics. For future use, we denote the distances between a simulated and the observed dataset through the list of selected combinations of summary statistics and distance metrics as $\widetilde{\boldsymbol{S}}^* \in \mathbb{R}^{\mathscr{d}_s}$ with $\mathscr{d}_s = 25$. When implementing ABC sampler, $\{\alpha_i\}_{0 \leqslant i \leqslant C-1} = \{0.55, 0.5, 0.45, 0.4, 0.35, 0.3, 0.2, 0.15, 0.1, 0.088\}$; with $\{N_i\}_{0 \leqslant i \leqslant C} = \{400000, 240000, 200000, 140000, 100000, 60000, 30000, 20000, 10000, 1600, 200\}$.

B.3 Efficiency explorations and more results with the real-world dataset

	$c=0$	$c=1$	$c=2$	$c=3$	$c=4$	$c=5$
α_{c-1}	1	0.55	0.50	0.45	0.40	0.35
δ_{c-1}	∞	21.01	18.46	17.05	15.60	14.19
$N_c^{proposed}$	400,000	240,000	200,000	140,000	100,000	60,000
N_c	400,017	242,959	201,541	141,150	100,473	60,241
N_c^{unique}	400,017	242,956	201,538	141,148	100,471	60,241
$n(\mathcal{X}_c)$	5000^5	$> 1 \times 10^{10}$	$> 5.3 \times 10^9$	$> 4.1 \times 10^9$	$> 2.9 \times 10^9$	$> 2.0 \times 10^9$
$n(\mathcal{X}_c)_{ESS}^{lb}$	5000^5	$> 1 \times 10^{10}$	$> 5.3 \times 10^9$	$> 4.1 \times 10^9$	$> 2.8 \times 10^9$	$> 1.9 \times 10^9$
N_c^{trials}	400,017	440,000	670,000	890,000	1,270,000	1,710,000
P_c^{accep}	1	0.5522	0.3008	0.1586	0.0791	0.0352
P_c^{prior}	1	0.5510	0.2746	0.1215	0.0462	0.0152
t_c	96.35	152.85	190.27	236.50	326.55	433.95

	$c=6$	$c=7$	$c=8$	$c=9$	$c=10$
α_{c-1}	0.30	0.20	0.15	0.10	0.088
δ_{c-1}	13.02	11.88	10.91	9.995	9.135
$N_c^{proposed}$	30,000	20,000	10,000	1,600	200
N_c	30,046	20,018	10,012	1,600	200
N_c^{unique}	30,045	20,018	10,010	1,600	200
$n(\mathcal{X}_c)$	$> 1.2 \times 10^9$	$> 6.3 \times 10^8$	$> 3.8 \times 10^8$	$> 1.9 \times 10^8$	$> 4.5 \times 10^7$
$n(\mathcal{X}_c)_{ESS}^{lb}$	$> 1.1 \times 10^9$	$> 5.6 \times 10^8$	$> 3.3 \times 10^8$	$> 1.5 \times 10^8$	$> 2.6 \times 10^7$
N_c^{trials}	2,180,000	5,390,000	11,900,000	11,410,000	9,790,000
P_c^{accep}	0.0138	0.0037	8.413×10^{-4}	1.402×10^{-4}	2.043×10^{-5}
P_c^{prior}	0.0041	0.0007	8.200×10^{-5}	7.700×10^{-6}	4.000×10^{-7}
t_c	550.62	1362.4	3018.8	2903.7	2489.9

Table B.7: Results of applying ABC-PMC to GP-IPM for real case study. **Notations**: $\delta_{\mathbf{c-1}}$: the quantile-based acceptance threshold used at step c, which was determined based on $\alpha_{\mathbf{c-1}}$ quantile at step $c-1$ (so denoted by δ_{c-1}). δ_{-1} and α_{-1} are for the initialisation. $\mathbf{N_c^{proposed}}$: the proposed number of accepted particles required for the simulation step c. $\mathbf{N_c}$: the true number of accepted particles for the simulation step c. $\mathbf{N_c^{unique}}$: the number of unique particles among N_c acceptances. $\mathbf{n}(\mathcal{X}_\mathbf{c})$: the total number of possible states of $\boldsymbol{\theta}$ at simulation step c. Apart from $\mathbf{n}(\mathcal{X}_\mathbf{0})$, the others were estimated through $n(\mathcal{X}_c)^{lb}$. $\mathbf{n}(\mathcal{X}_\mathbf{c})_\mathbf{ESS}^\mathbf{lb}$: a lower bound for $n(\mathcal{X}_c)$ but estimated based of ESS. $\mathbf{N_c^{trials}}$: total number of trials to achieve the acceptance. $\mathbf{P_c^{accep}} = \mathbf{N_c}/\mathbf{N_c^{trials}}$: the acceptance rate for step c. $\mathbf{P_c^{prior}}$: the acceptance rate for a single-iterated ABC-PMC, where the samples were drawn directly from the priors made up by MCMC samples. For the reliability, P_c^{prior}s were computed based on 10,000,190 particles drawn from $\pi(\boldsymbol{\theta})$. To ensure comparability, P_c^{prior}s must be computed under the exactly same setting that produced P_c^{accep}s. That is, for each c in $\{0, 1, ..., C\}$, the 10,000,190 particles were normalised with MADs $\boldsymbol{\sigma}_{c-1}$ first, the threshold δ_{c-1} then employed to compute P_c^{prior}. It took approximately 2,541 minutes to complete this single-iterated algorithm with 10,000,190 particles. $\mathbf{t_c}$: the total amount of time, measured in minutes, used to complete the cth iteration.

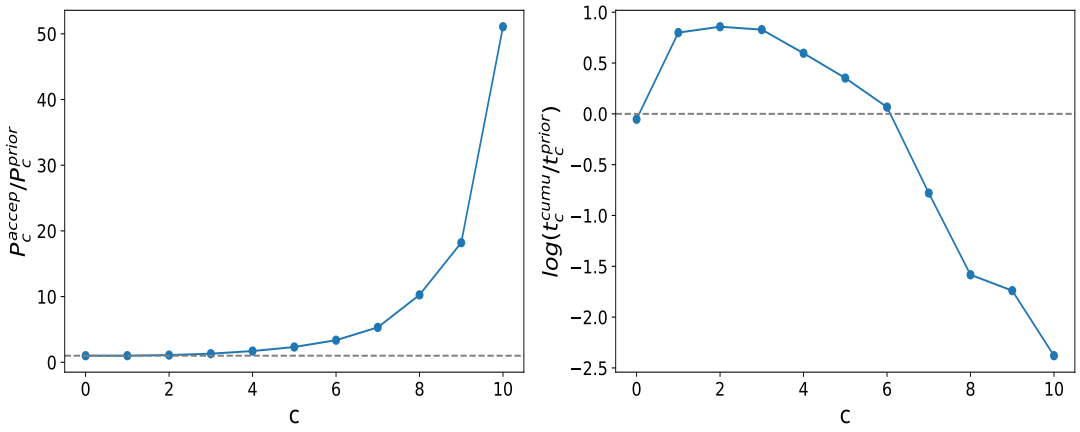


Figure B.4: Left: The ratio of P_c^{accep} and P_c^{prior} over simulation time c. The grey horizontal line cross the point representing $P_c^{accep}/P_c^{prior} = 1$. **Right:** The logarithmic ratio between the cumulative time spent on ABC-PMC, represented as t_c^{cumu}, and the expected implementation time, estimated as t_c^{prior}. The t_c^{cumu} is obtained by adding up individual times for each single iteration t_c until the current iteration c, $t_c^{cumu} = \sum_0^c t_c$. Instead of timing in real practices, t_c^{prior} was estimated using the formula $t_c^{prior} = (N_c/(P_c^{prior} * 10000190)) * 2541$ based on the 10,000,190 particles (see Table B.7).

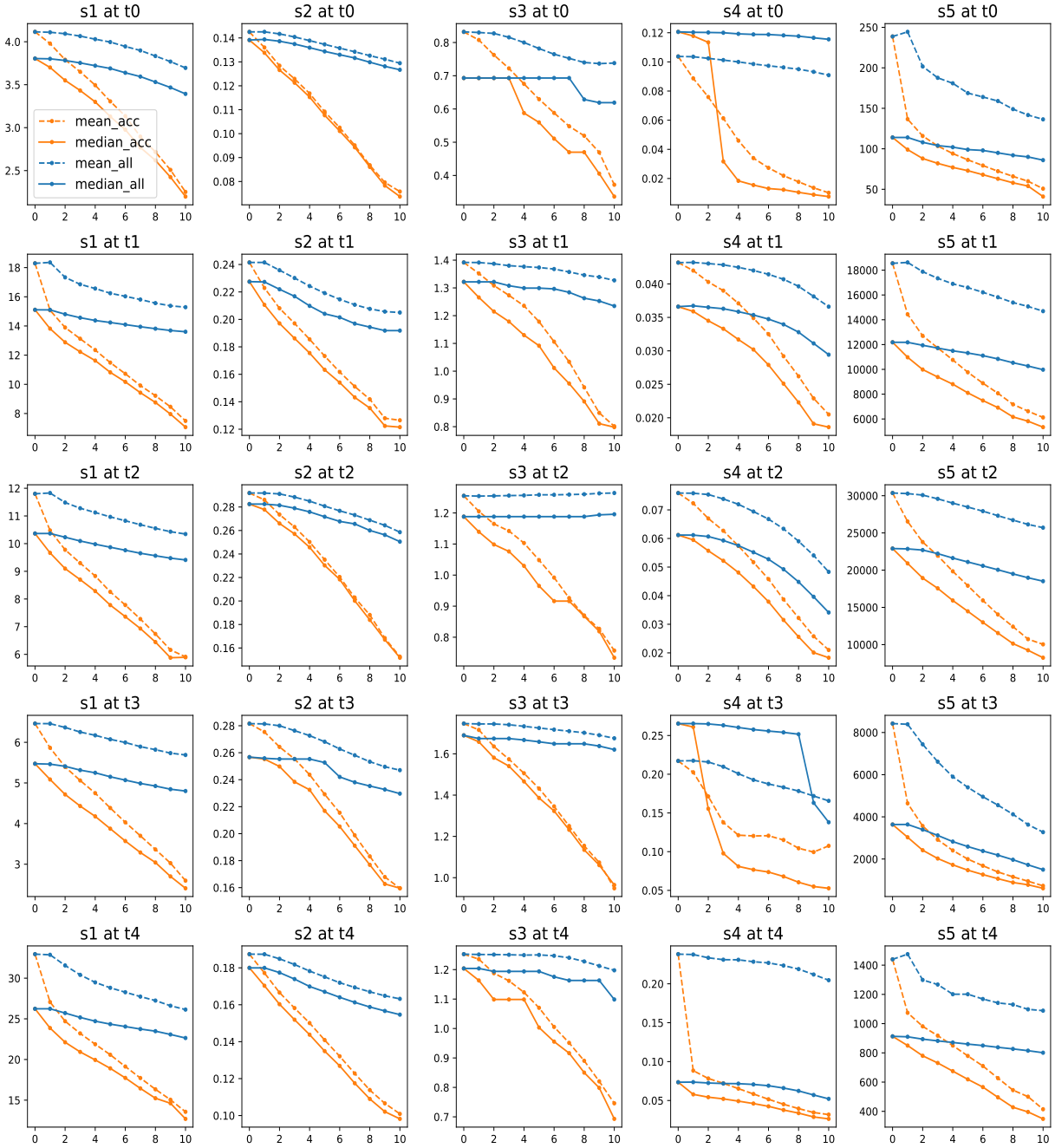


Figure B.5: The mean (dashed line) and median (solid line) for the 25 summary statistics over the ABC-PMC simulation time c ($x-$axis). The orange lines describing the information of summary statistics for accepted particles only, while the blues stand for all particles including those rejected. **s1** to **s5** represent the five types of summary statistics, **t0**, **t1**, ..., **t4** indicate year 2003-2004, 2004-2005, ..., 2007-2008.

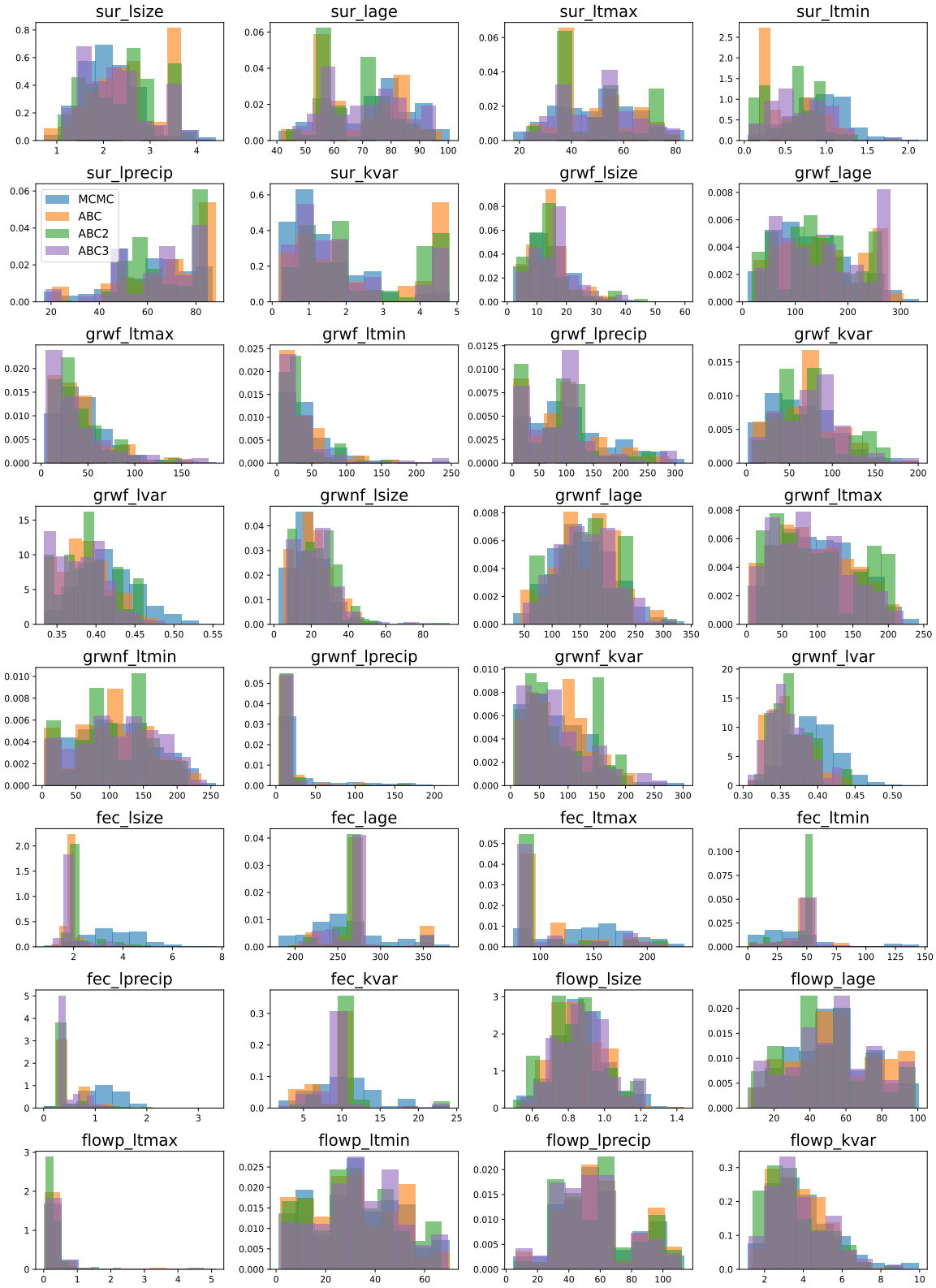


Figure B.6: Density histograms of estimated marginal posterior distributions for all hyperparameters under methods GP and ABC_GP.

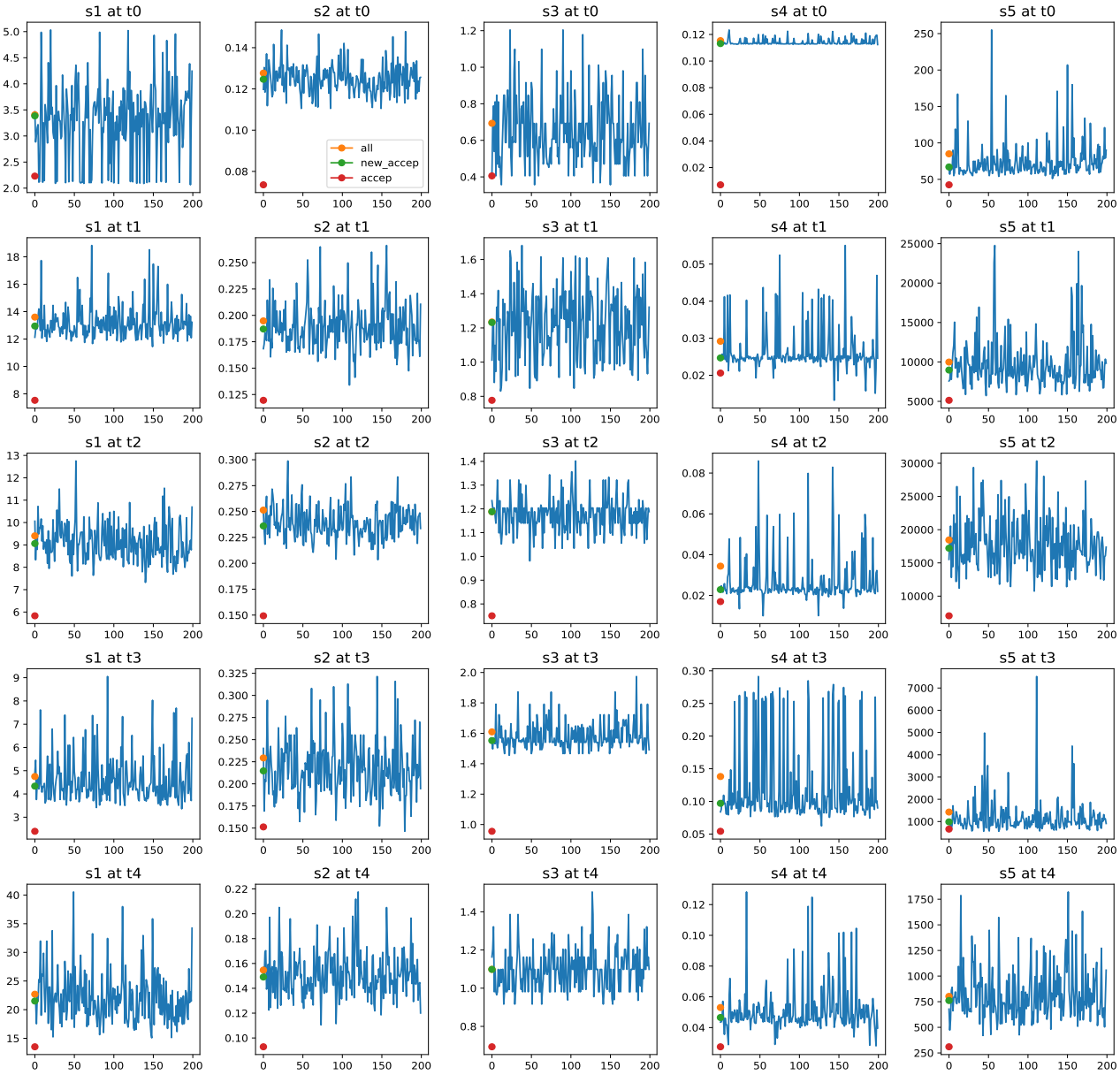


Figure B.7: We created 1,000 simulations for each of 200 samples drawn from the ABC posterior distribution. The median was then computed from the summary statistics of each set of simulations, yielding 200 median values. These medians were used to construct the **trace** seen in each plot. The **green** dot in each plot represents the overall median of these 200 medians. Meanwhile, the **orange** and **red** dots denote the median of the simulated summary statistics vectors for the accepted particles only, and for all particles (including rejected ones) during the ABC-PMC sampling process, respectively. The labels 's1' through 's5' represent the five different types of summary statistics used in our analysis. The labels 't0', 't1', through 't4' correspond to the years from 2003-2004 to 2007-2008.

B.3.1 Marginal posteriors in real case study

We compared the marginal posteriors of GLM, GP, and ABC_GP methods across five vital rates and yearly datasets in Figure B.8 for the real case study of *C. Flava*. This analysis is intended to illustrate model behaviour with real datasets, and should not be interpreted as providing species-specific biological insight.

Noticeable difference across models lie in survival, flowering, and fecundity. These differences reflect how the three models differ in their structural assumptions and in how they express uncertainty across years. GLMs rely on fixed functional forms, which lead to relatively consistent posterior shapes across years (for the same vital rate). This stability may offer easier interpretability and reduce computational burden, but it limits the model's ability to adapt when the underlying patterns vary across years or under different environmental conditions.

In contrast, GP and ABC_GP methods allow more flexibility. The shape of the GP posterior varies across years, reflecting the influence of year-specific data and allowing for non-linear responses. ABC_GP introduces an additional layer of constraint by rejecting parameter combinations that lead to unrealistic population trajectories. Unlike approaches that regularise each sub-model directly, the adjustment here comes from evaluating the behaviour of the combined model. The adjustment leads to visible changes. ABC_GP often narrows the credible intervals and noticeably shifts the posterior mean.

This flexibility of GP and ABC_GP can be helpful when patterns change over time, but it can also lead to instability, especially in test years with limited data or when climate conditions differ greatly from those seen during training, for example, an unusually warm year 2011-2012. In these extrapolations, credible intervals tend to widen, reflecting increased uncertainty, and posterior means become visibly flatter. This does not necessarily indicate over-fitting; rather, it can reflect the model's sensitivity to weak signals and its tendency to revert to the zero-mean prior when data support is limited. GLM, by contrast, maintains its pre-defined form and gives more stable forecasts, though these may under-represent uncertainty when

extrapolating beyond the training range. In such cases, uncertainty or flatness in flexible models may more honestly reflect what the data can support.

For growth rates, all models produce broadly similar results. In some years, however, ABC-GP yields wider intervals for non-breeders.

These results do not suggest that any one approach is universally better. Rather, they highlight how model assumptions interact with data availability. GLMs prioritise consistency, GPs offer flexibility, and ABC-GP provides a way to moderate that flexibility by incorporating population-level information.

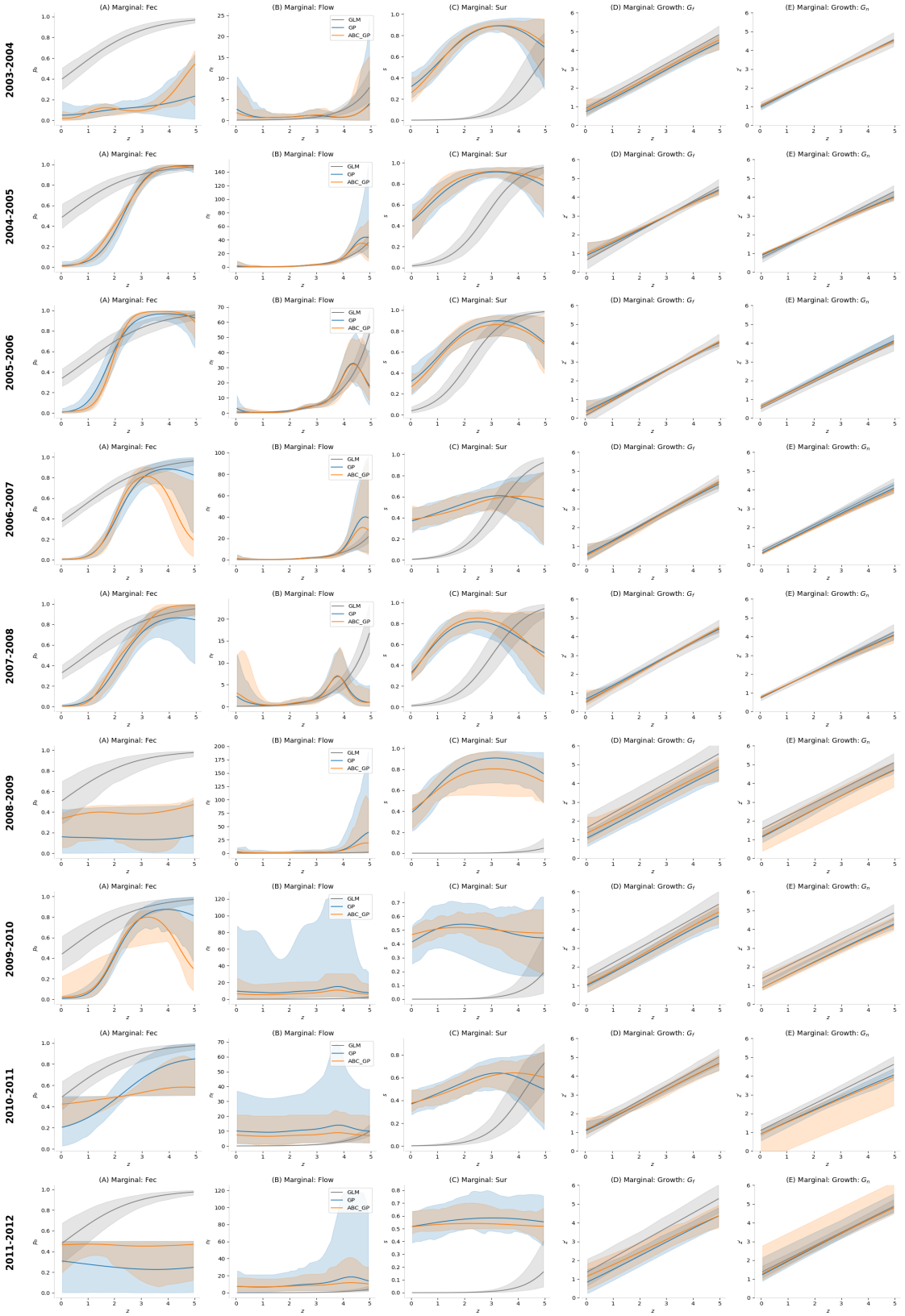


Figure B.8: Posteriors across five vital rates and nine years under three modelling methods (GLM, GP, and ABC_GP). For posterior visualisation, we fixed age at 2, discretised logged size into 50 mesh points between 0 and 5, and included the real year-specific climate information as model inputs. In the figure, each row represents a different vital rate, and each column corresponds to a different year.

C
Appendix for Chapter 5

For the case study of Soay sheep, each vital rate was modelled separately as a function of the individual's state in the current year (z) or the following year (z'), along with environmental covariates $\boldsymbol{\omega}$ representing modified weather and population density factors. Specifically:

- probabilities of survival $s(z|\boldsymbol{\omega})$ and probabilities of recruitment $p_f(z|\boldsymbol{\omega})$, using GPs with Bernoulli likelihoods;
- growth function $G(z'|z, \boldsymbol{\omega})$ and newborn size distribution $r_{size}(z'|z, \boldsymbol{\omega})$, using GP regressions.

For each of the GP models, the mean function was set to zero, and the Gaussian kernel function was applied. The Gaussian kernel function was employed with the automatic relevance determination setting, allowing it to separately modulate the magnitude of similarity in each input coordinate (size, modified weather and density factors). Bayesian individual vital rate model fitting was conducted using non-informative priors and Hamiltonian Monte Carlo methods (Hensman et al. 2015). All the GP model fitting tasks were implemented using the `GPflow` package in the `Python` programming language (Van Rossum and Drake Jr 1995; Matthews et al. 2017).

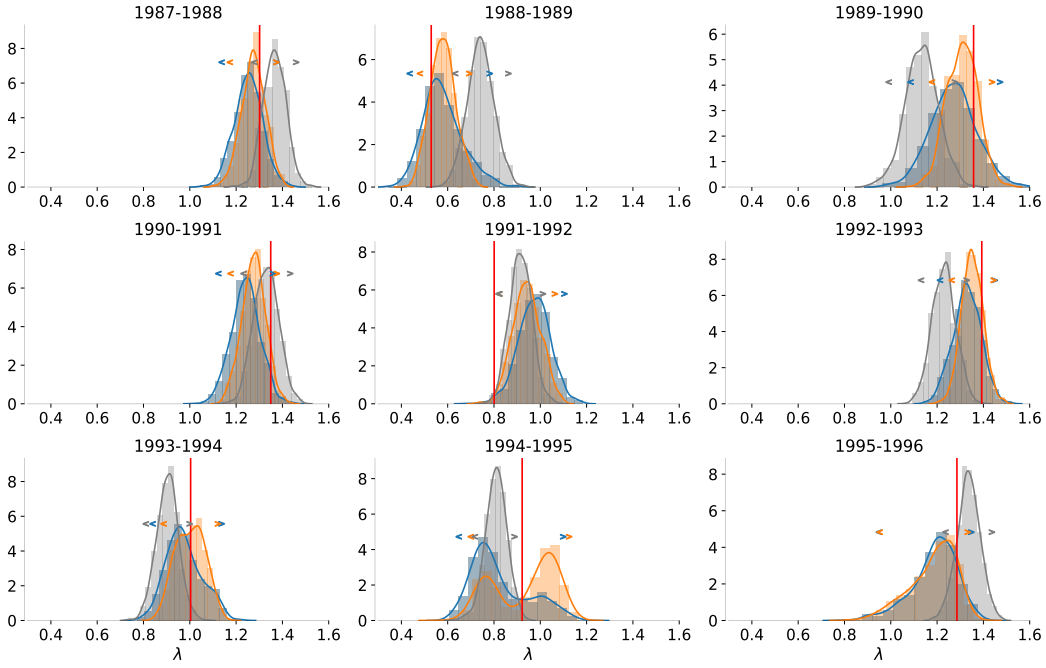


Figure C.1: Histograms represent the density of simulated population growth rates generated through **Long-term** forecasting. The associated 95% prediction intervals are denoted by angle brackets, each in its corresponding colour, positioned at the top of each histogram. The red vertical line represents the true observed population growth rate.

References

Amrein, M. and H. R. Künsch (2011). "A variant of importance splitting for rare event estimation: Fixed number of successes". In: *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 21.2, pp. 1–20.

Andrieu, C., A. Lee, and M. Vihola (2018). "Theoretical and methodological aspects of MCMC computations with noisy likelihoods". In: *Chapman & Hall/CRC Handbooks of Modern Statistical Methods*.

Andrieu, C. and G. O. Roberts (2009). "The pseudo-marginal approach for efficient Monte Carlo computations". In: *The Annals of Statistics* 37.2, pp. 697–725. URL: https://doi.org/10.1214/07-AOS574.

Baden, H. M. et al. (2021). "The effects of age on the demography of a perennial plant depend on interactions with size and environment". In: *Journal of Ecology* 109.2, pp. 1068–1077.

Bay, X., L. Grammont, and H. Maatouk (2016). "Generalization of the Kimeldorf-Wahba correspondence for constrained interpolation". In: *Electronic journal of statistics* 10.1, pp. 1580–1595.

Beaumont, M. A., J.-M. Cornuet, et al. (2009). "Adaptive Approximate Bayesian Computation". In: *Biometrika* 96.4, pp. 983–990.

Beaumont, M. A., W. Zhang, and D. J. Balding (2002). "Approximate Bayesian Computation in population genetics". In: *Genetics* 162.4, pp. 2025–2035.

Belgrano, A., O. Lindahl, and B. Hernroth (1999). "North Atlantic Oscillation primary productivity and toxic phytoplankton in the Gullmar Fjord, Sweden (1985–1996)". In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 266.1418, pp. 425–430.

Bernardo, J. et al. (1998). "Regression and classification using Gaussian Process priors". In: *Bayesian statistics* 6, p. 475.

Bernton, E. et al. (2017). "Inference in generative models using the Wasserstein distance". In: *arXiv preprint arXiv:1701.05146* 1.8, p. 9.

— (2019). "Approximate Bayesian Computation with the Wasserstein distance". In: *arXiv preprint arXiv:1905.03747*.

Blum, M. G. et al. (2013). "A comparative review of dimension reduction methods in Approximate Bayesian Computation". In: *Statistical Science* 28.2, pp. 189–208.

Bonassi, F. V. and M. West (2015). "Sequential Monte Carlo with adaptive weights for Approximate Bayesian Computation". In: *Bayesian Analysis* 10.1, pp. 171–187.

Bond-Taylor, S. et al. (2021). "Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models". In: *IEEE transactions on pattern analysis and machine intelligence*.

Bonnaffé, W. and T. Coulson (2022). "Fast fitting of neural ordinary differential equations by Bayesian neural gradient matching to infer ecological interactions from time series data". In: *arXiv preprint arXiv:2209.06184*.

Bornn, L. et al. (2017). "The use of a single pseudo-sample in Approximate Bayesian Computation". In: *Statistics and Computing* 27, pp. 583–590.

Boyd, J. and P. Jewell (1974). *The Soay sheep and their environment: a synthesis. Island Survivors.*

Cappé, O. et al. (2004). "Population Monte Carlo". In: *Journal of Computational and Graphical Statistics* 13.4, pp. 907–929.

Capretto, T. et al. (2022). "Bambi: A Simple Interface for Fitting Bayesian Linear Models in Python". In: *Journal of Statistical Software* 103.15, pp. 1–29. URL: `https://www.jstatsoft.org/index.php/jss/article/view/v103i15`.

Casper, B., I. Forseth, and D. A. Wait (2005). "Variation in carbon isotope discrimination in relation to plant performance in a natural population of *Cryptantha flava*". In: *Oecologia* 145.4, pp. 541–548.

Casper, B. B. (1988). "Evidence for selective embryo abortion in *Cryptantha flava*". In: *The American Naturalist* 132.3, pp. 318–326.

— (1996). "Demographic consequences of drought in the herbaceous perennial *Cryptantha flava*: effects of density, associations with shrubs, and plant size". In: *Oecologia* 106.2, pp. 144–152.

Casper, B. B., I. Forseth, and D. A. Wait (2006). "A stage-based study of drought response in *Cryptantha flava* (Boraginaceae): gas exchange, water use efficiency, and whole plant performance". In: *American Journal of Botany* 93.7, pp. 978–987.

Caswell, H. (2000). *Matrix Population Models.* Vol. 1. Sinauer Sunderland, MA, USA.

Catchpole, E. A. et al. (2000). "Factors influencing Soay sheep survival". In: *Journal of the Royal Statistical Society Series C: Applied Statistics* 49.4, pp. 453–472.

Cavender-Bares, J. and F. Bazzaz (2000). "Changes in drought response strategies with ontogeny in *Quercus rubra*: implications for scaling from seedlings to mature trees". In: *Oecologia* 124.1, pp. 8–18.

Chen, Z. (2017). "Gaussian Process Regression methods and extensions for stock market prediction". PhD thesis. University of Leicester.

Cheng, C.-A. and B. Boots (2017). "Variational inference for Gaussian Process models with linear complexity". In: *Advances in Neural Information Processing Systems* 30.

Childs, D. Z. et al. (2003). "Evolution of complex flowering strategies: an age–and size–structured Integral Projection Model". In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 270.1526, pp. 1829–1838.

— (2004). "Evolution of size–dependent flowering in a variable environment: construction and analysis of a stochastic Integral Projection Model". In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 271.1537, pp. 425–434.

Chu, C. and P. B. Adler (2014). "When should plant population models include age structure?" In: *Journal of Ecology* 102.2, pp. 531–543.

Clarté, G. et al. (2021). "Componentwise Approximate Bayesian Computation via Gibbs-like steps". In: *Biometrika* 108.3, pp. 591–607.

Clutton–Brock, T. and T. Coulson (2002). "Comparative ungulate dynamics: the devil is in the detail". In: *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* 357.1425, pp. 1285–1298.

Clutton-Brock, T., B. Grenfell, et al. (2004). *Population dynamics in Soay sheep.*

Clutton-Brock, T., A. Illius, et al. (1997). "Stability and instability in ungulate populations: an empirical analysis". In: *The American Naturalist* 149.2, pp. 195–219.

Clutton-Brock, T., J. Pemberton, et al. (2004). *The sheep of St Kilda.*

Clutton-Brock, T., I. Stevenson, et al. (1996). "Population fluctuations, reproductive costs and life-history tactics in female Soay sheep". In: *Journal of Animal Ecology*, pp. 675–689.

Clutton-Brock, T. H. and J. M. Pemberton (2004). *Soay sheep: dynamics and selection in an island population*. Cambridge University Press.

Coles, C. et al. (2023). "Using Integral Projection Models to explore management strategies for silver carp (*Hypophthalmichthys molitrix*)". In: *Spora: A Journal of Biomathematics* 9.1, pp. 37–48.

Çorbacıoğlu, Ş. K. and G. Aksel (2023). "Receiver operating characteristic curve analysis in diagnostic accuracy studies: A guide to interpreting the area under the curve value". In: *Turkish journal of emergency medicine* 23.4, pp. 195–198.

Coulson, T. (2012). "Integral Projection Models, their construction and use in posing hypotheses in ecology". In: *Oikos* 121.9, pp. 1337–1350.

Coulson, T., E. A. Catchpole, et al. (2001). "Age, sex, density, winter weather, and population crashes in Soay sheep". In: *Science* 292.5521, pp. 1528–1531.

Coulson, T., S. Tuljapurkar, and D. Z. Childs (2010). "Using evolutionary demography to link life history theory, quantitative genetics and population ecology". In: *Journal of Animal Ecology* 79.6, pp. 1226–1240.

Cox, D. R. and D. V. Hinkley (1979). *Theoretical statistics*. CRC Press.

Crawley, M. et al. (2004). *Vegetation and sheep population dynamics.*

Csilléry, K., M. G. Blum, et al. (2010). "Approximate Bayesian Computation (ABC) in practice". In: *Trends in ecology & evolution* 25.7, pp. 410–418.

Csilléry, K., O. François, and M. G. Blum (2012). "abc: an R package for Approximate Bayesian Computation (ABC)". In: *Methods in ecology and evolution* 3.3, pp. 475–479.

Dahlgren, J. P., K. Bengtsson, and J. Ehrlén (2016). "The demography of climate-driven and density-regulated population dynamics in a perennial plant". In: *Ecology* 97.4, pp. 899–907.

Dahlgren, J. P., M. B. García, and J. Ehrlén (2011). "Nonlinear relationships between vital rates and state variables in demographic models". In: *Ecology* 92.5, pp. 1181–1187.

Daly, A. C. et al. (2017). "Comparing two Sequential Monte Carlo samplers for exact and approximate Bayesian inference on biological models". In: *Journal of The Royal Society Interface* 14.134, p. 20170340.

DeAngelis, D. L. and V. Grimm (2014). "Individual-based models in ecology after four decades". In: *F1000prime reports* 6.

Del Moral, P., A. Doucet, and A. Jasra (2012). "An adaptive Sequential Monte Carlo method for Approximate Bayesian Computation". In: *Statistics and computing* 22.5, pp. 1009–1020.

Donovan, L. A. et al. (2007). "Phenotypic selection on leaf water use efficiency and related ecophysiological traits for natural populations of desert sunflowers". In: *Oecologia* 152.1, pp. 13–25.

Drovandi, C. C. and A. N. Pettitt (2011). "Estimation of parameters for macroparasite population evolution using Approximate Bayesian Computation". In: *Biometrics* 67.1, pp. 225–233.

Drovandi, C. C., A. N. Pettitt, and M. J. Faddy (2011). "Approximate Bayesian Computation using indirect inference". In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 60.3, pp. 317–337.

Duvenaud, D. (2014). "Automatic model construction with Gaussian Processes". PhD thesis. University of Cambridge.

Easterling, D. R. et al. (2000). "Climate extremes: observations, modeling, and impacts". In: *science* 289.5487, pp. 2068–2074.

Easterling, M. R., S. P. Ellner, and P. M. Dixon (2000). "Size-specific sensitivity: applying a new structured population model". In: *Ecology* 81.3, pp. 694–708.

Edelfeldt, S., K. Bengtsson, and J. P. Dahlgren (2019). "Demographic senescence and effects on population dynamics of a perennial plant". In: *Ecology* 100.8, e02742.

Elderd, B. D. and T. E. Miller (2016). "Quantifying demographic uncertainty: Bayesian methods for Integral Projection Models". In: *Ecological Monographs* 86.1, pp. 125–144.

Ellison, A. M. (2004). "Bayesian inference in ecology". In: *Ecology letters* 7.6, pp. 509–520.

Ellner, S. P., D. Z. Childs, and M. Rees (2016). *Data-driven Modelling of Structured Populations: A Practical Guide to the Integral Projection Model*. Springer.

Ellner, S. P. and M. Rees (2006). "Integral Projection Models for species with complex demography". In: *The American Naturalist* 167.3, pp. 410–428.

— (2007). "Stochastic stable population growth in Integral Projection Models: theory and application". In: *Journal of mathematical biology* 54.2, pp. 227–256.

Erhardt, R. and S. A. Sisson (2016). "Modelling extremes using Approximate Bayesian Computation". In: *Extreme Value Modelling and Risk Analysis*, pp. 281–306.

Everitt, R. G. and P. A. Rowińska (2021). "Delayed acceptance ABC-SMC". In: *Journal of Computational and Graphical Statistics* 30.1, pp. 55–66.

Evers, S. M. et al. (2021). "Lagged and dormant season climate better predict plant vital rates than climate during the growing season". In: *Global Change Biology* 27.9, pp. 1927–1941.

Fagan, D. (2012). *Canyon Country Wildflowers: A Guide to Common Wildflowers, Shrubs, and Trees*. Rowman & Littlefield.

Farrell, S. et al. (2021). "The potential for complex computational models of aging". In: *Mechanisms of Ageing and Development* 193, p. 111403.

Fearnhead, P. and D. Prangle (2012). "Constructing summary statistics for Approximate Bayesian Computation: semi-automatic Approximate Bayesian Computation". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 74.3, pp. 419–474.

Félix-Burruel, R. E. et al. (2024). "Population decline of the saguaro cactus throughout its distribution is associated with climate change". In: *Annals of Botany*, mcae094.

Fieberg, J. and S. P. Ellner (2001). "Stochastic Matrix Models for conservation and management: a comparative review of methods". In: *Ecology letters* 4.3, pp. 244–266.

Figueiredo, R., P. Nunes, and M. C. Brito (2018). "Multiyear calibration of simulations of energy systems". In: *Energy* 157, pp. 932–939.

Filippi, S. et al. (2013). "On optimality of kernels for Approximate Bayesian Computation using Sequential Monte Carlo". In: *Statistical applications in genetics and molecular biology* 12.1, pp. 87–107.

Forchhammer, M. C. et al. (1998). "Population dynamics of Norwegian red deer: density–dependence and climatic variation". In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 265.1393, pp. 341–350.

Friel, N. and A. N. Pettitt (2008). "Marginal likelihood estimation via power posteriors". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 70.3, pp. 589–607.

Fu, Y.-X. and W.-H. Li (1997). "Estimating the age of the common ancestor of a sample of DNA sequences." In: *Molecular biology and evolution* 14.2, pp. 195–199.

Fung, Y. L. et al. (2022). "Building Integral Projection Models with nonindependent vital rates". In: *Ecology and Evolution* 12.3, e8682.

Gail, M. H. and R. M. Pfeiffer (2018). "Breast cancer risk model requirements for counseling, prevention, and screening". In: *JNCI: Journal of the National Cancer Institute* 110.9, pp. 994–1002.

Gershman, S. J. and D. M. Blei (2012). "A tutorial on Bayesian nonparametric models". In: *Journal of Mathematical Psychology* 56.1, pp. 1–12.

Golding, N. and B. V. Purse (2016). "Fast and flexible Bayesian species distribution modelling using Gaussian Processes". In: *Methods in Ecology and Evolution* 7.5, pp. 598–608.

González, E. J., C. Martorell, and B. M. Bolker (2016). "Inverse estimation of Integral Projection Model parameters using time series of population-level data". In: *Methods in Ecology and Evolution* 7.2, pp. 147–156.

Gramacy, R. B. (2020). *Surrogates: Gaussian Process modeling, design, and optimization for the applied sciences*. Chapman and Hall/CRC.

Grimm, V. et al. (2006). "A standard protocol for describing individual-based and agent-based models". In: *Ecological modelling* 198.1-2, pp. 115–126.

Grubb, P. et al. (1974). "Island survivors: the ecology of the Soay sheep of St Kilda". In: *Ch* 8, pp. 195–223.

Grubb, P. (1974). "Population dynamics of the Soay sheep". In: *Island Survivors; the ecology of the Soay Sheep of St. Kilda. PA Jewell, C. Milner, & J. Morton Boyd, eds.*

Gutmann, M. U. et al. (2018). "Likelihood-free inference via classification". In: *Statistics and Computing* 28, pp. 411–425.

Hamilton, G. et al. (2005). "Bayesian estimation of recent migration rates after a spatial expansion". In: *Genetics* 170.1, pp. 409–417.

Harris, C. R. et al. (Sept. 2020). "Array programming with NumPy". In: *Nature* 585.7825, pp. 357–362. URL: https://doi.org/10.1038/s41586-020-2649-2.

Harrison, J. U. and R. E. Baker (2020). "An automatic adaptive method to combine summary statistics in Approximate Bayesian Computation". In: *PloS one* 15.8, e0236954.

Hastie, T. et al. (2009). *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer.

Hensman, J. et al. (2015). "MCMC for variationally sparse Gaussian Processes". In: *Advances in Neural Information Processing Systems* 28.

Hey, J. and C. A. Machado (2003). "The study of structured populations—new hope for a difficult and divided science". In: *Nature Reviews Genetics* 4.7, pp. 535–543.

Hilborn, R., R. C. Francis, and D. E. Rogers (2006). "Marine and freshwater climatic factors affecting interannual variation in the timing of return migration to fresh water of sockeye salmon (*Oncorhynchus nerka*)". In: *Fisheries Oceanography* 15.1, pp. 1–24.

Hobbs, N. T. and R. Hilborn (2006). "Alternatives to statistical hypothesis testing in ecology: a guide to self teaching". In: *Ecological Applications* 16.1, pp. 5–19.

Hosmer Jr, D. W., S. Lemeshow, and R. X. Sturdivant (2013). *Applied Logistic Regression*. Vol. 398. John Wiley & Sons.

Hunter, J. D. (2007). "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3, pp. 90–95.

Huxman, T. E. et al. (2004). "Convergence across biomes to a common rain-use efficiency". In: *Nature* 429.6992, pp. 651–654.

Ingram, M., D. Vukcevic, and N. Golding (2020). "Multi-output Gaussian Processes for species distribution modelling". In: *Methods in Ecology and Evolution* 11.12, pp. 1587–1598.

Iyer, R. et al. (2016). "Screening peers softly: Inferring the quality of small borrowers". In: *Management Science* 62.6, pp. 1554–1577.

Jabot, F. and J. Chave (2011). "Analyzing tropical forest tree species abundance distributions using a nonneutral model and through approximate Bayesian inference". In: *The American Naturalist* 178.2, E37–E47.

Jasra, A. (2015). "Approximate Bayesian Computation for a class of time series models". In: *International Statistical Review* 83.3, pp. 405–435.

Jasra, A., A. Lee, et al. (2013). "The Alive Particle Filter". In: *arXiv preprint arXiv:1304.0151*.

Jasra, A., S. S. Singh, et al. (2012). "Filtering via Approximate Bayesian Computation". In: *Statistics and Computing* 22, pp. 1223–1237.

Jones, O. (2003). "Grazing Ecology, Parasitism and Performance of Soay Sheep on St Kilda". In: *University of London, London*.

Joyce, P. and P. Marjoram (2008). "Approximately sufficient statistics and Bayesian computation". In: *Statistical applications in genetics and molecular biology* 7.1.

Kelly, C. K. (1992). "Resource choice in *Cuscuta europaea*." In: *Proceedings of the National Academy of Sciences* 89.24, pp. 12194–12197.

Kimeldorf, G. S. and G. Wahba (1970). "A correspondence between Bayesian estimation on stochastic processes and smoothing by splines". In: *The Annals of Mathematical Statistics* 41.2, pp. 495–502.

King, R. et al. (2006). "Factors influencing soay sheep survival: a Bayesian analysis". In: *Biometrics* 62.1, pp. 211–220.

Lauenroth, W. K. and P. B. Adler (2008). "Demography of perennial grassland plants: survival, life expectancy and life span". In: *Journal of Ecology* 96.5, pp. 1023–1032.

Lenormand, M., F. Jabot, and G. Deffuant (2013). "Adaptive Approximate Bayesian Computation for complex models". In: *Computational Statistics* 28.6, pp. 2777–2796.

Levin, S. C. et al. (2021). "ipmr: Flexibly implement Integral Projection Models in R". In: *bioRxiv*.

Li, S. and M. Jakobsson (2012). "Estimating demographic parameters from large-scale population genomic data using Approximate Bayesian Computation". In: *BMC genetics* 13, pp. 1–15.

Li, Y., E. Schofield, and M. Gönen (2019). "A tutorial on Dirichlet process mixture modeling". In: *Journal of mathematical psychology* 91, pp. 128–144.

Lingjærde, O. C. et al. (2001). "Exploring the density-dependent structure of blowfly populations by nonparametric additive modeling". In: *Ecology* 82.9, pp. 2645–2658.

Lintusaari, J. et al. (2017). "Fundamentals and recent developments in Approximate Bayesian Computation". In: *Systematic biology* 66.1, e66–e82.

Lucas, R. W., I. N. Forseth, and B. B. Casper (2008). "Using rainout shelters to evaluate climate change effects on the demography of *Cryptantha flava*". In: *Journal of Ecology* 96.3, pp. 514–522.

Luciani, F. et al. (2009). "The epidemiological fitness cost of drug resistance in *Mycobacterium tuberculosis*". In: *Proceedings of the National Academy of Sciences* 106.34, pp. 14711–14715.

Lux, T. (2021). "Bayesian estimation of agent-based models via adaptive particle Markov chain Monte Carlo". In: *Computational Economics*, pp. 1–27.

Macdonald, K. R., J. J. Rotella, and J. T. Paterson (2023). "Evaluating the importance of individual heterogeneity in reproduction to Weddell seal population dynamics using Integral Projection Models". In: *Journal of Animal Ecology* 92.9, pp. 1828–1839.

Marin, J.-M. et al. (2012). "Approximate Bayesian Computational methods". In: *Statistics and Computing* 22.6, pp. 1167–1180.

Marjoram, P., J. Molitor, et al. (2003). "Markov chain Monte Carlo without likelihoods". In: *Proceedings of the National Academy of Sciences* 100.26, pp. 15324–15328.

Marjoram, P. and S. Tavaré (2006). "Modern computational approaches for analysing molecular genetic variation data". In: *Nature Reviews Genetics* 7.10, pp. 759–770.

Martín Abadi et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. URL: `https://www.tensorflow.org/`.

Martorell, C. and A. Martínez-Ballesté (2019). "Stress and disturbance determine the demographic dynamics of two grass species along a grazing gradient in Southern Mexico". In: *Population Ecology* 61.2, pp. 160–170.

Matthews, A. G. d. G. et al. (2017). "GPflow: A Gaussian Process library using TensorFlow". In: *Journal of Machine Learning Research* 18.40, pp. 1–6. URL: `http://jmlr.org/papers/v18/16-537.html`.

McNamara, J. M. and A. I. Houston (1996). "State-dependent life histories". In: *Nature* 380.6571, pp. 215–221.

Mencuccini, M. et al. (2005). "Size-mediated ageing reduces vigour in trees". In: *Ecology letters* 8.11, pp. 1183–1190.

Merow, C., J. P. Dahlgren, et al. (2014). "Advancing population ecology with Integral Projection Models: a practical guide". In: *Methods in Ecology and Evolution* 5.2, pp. 99–110.

Merow, C., A. M. Latimer, et al. (2014). "On using Integral Projection Models to generate demographically driven predictions of species' distributions: development and validation using sparse data". In: *Ecography* 37.12, pp. 1167–1183.

Messerman, A. F. et al. (2023). "Applying stochastic and Bayesian Integral Projection Modeling to amphibian population viability analysis". In: *Ecological Applications* 33.2, e2783.

Metcalf, C. J. E. et al. (2013). "IPMpack: an R package for Integral Projection Models". In: *Methods in Ecology and Evolution* 4.2, pp. 195–200.

Minka, T. P. (2013). "Expectation propagation for approximate Bayesian inference". In: *arXiv preprint arXiv:1301.2294*.

Mishra, S. et al. (2022). "π VAE: a stochastic process prior for Bayesian deep learning with MCMC". In: *Statistics and Computing* 32.6, p. 96.

Morris, W. F. and D. F. Doak (2002). *Quantitative Conservation Biology: Theory and Practice of Population Viability Analysis*.

Mysterud, A. (2006). "The concept of overgrazing and its role in management of large herbivores". In: *Wildlife Biology* 12.2, pp. 129–141.

National Trust for Scotland (2024). *St. Kilda World Heritage Site*. Accessed: 2024-07-29. URL: `https://www.nts.org.uk/visit/places/st-kilda`.

Neal, R. M. (1996). "Sampling from multimodal distributions using tempered transitions". In: *Statistics and computing* 6, pp. 353–366.

Newman, K. et al. (2022). "State-space models for ecological time-series data: Practical model-fitting". In: *Methods in Ecology and Evolution*.

Nielsen, R. and J. Wakeley (2001). "Distinguishing migration from isolation: a Markov Chain Monte Carlo approach". In: *Genetics* 158.2, pp. 885–896.

NOAA (2013). "National Temperature and Precipitation Analysis". In: *https://www.ncei.noaa.gov/access/monitoring/monthly-report/national/201213*.

Numminen, E. et al. (2013). "Estimating the transmission dynamics of Streptococcus pneumoniae from strain prevalence data". In: *Biometrics* 69.3, pp. 748–757.

Obeso, J. R. (2002). "The costs of reproduction in plants". In: *New phytologist* 155.3, pp. 321–348.

Opper, M. and C. Archambeau (2009). "The variational Gaussian approximation revisited". In: *Neural computation* 21.3, pp. 786–792.

Orbanz, P. and Y. W. Teh (2010). "Bayesian nonparametric models." In: *Encyclopedia of machine learning* 1.

Ozgul, A. et al. (2009). "The dynamics of phenotypic change and the shrinking sheep of St. Kilda". In: *Science* 325.5939, pp. 464–467.

Partridge, L. and N. H. Barton (1993). "Optimally, mutation and the evolution of ageing". In: *Nature* 362.6418, pp. 305–311.

Peek, M. S. and I. N. Forseth (2003). "Microhabitat dependent responses to resource pulses in the aridland perennial, *Cryptantha flava*". In: *Journal of Ecology* 91.3, pp. 457–466.

Peters, G. W., Y. Fan, and S. A. Sisson (2012). "On Sequential Monte Carlo, partial rejection control and Approximate Bayesian Computation". In: *Statistics and Computing* 22.6, pp. 1209–1222.

Picó, F., A. Rodrigo, and J. Retana (2008). "Plant Demography". In: *Encyclopedia of Ecology*. Ed. by S. E. Jørgensen and B. D. Fath. Oxford: Academic Press, pp. 2811–2817.

Plard, F. et al. (2019). "IPM 2: toward better understanding and forecasting of population dynamics". In: *Ecological Monographs* 89.3, e01364.

Post, E. and M. C. Forchhammer (2002). "Synchronization of animal population dynamics by large-scale climate". In: *Nature* 420.6912, pp. 168–171.

Post, E. and N. C. Stenseth (1998). "Large-scale climatic fluctuation and population dynamics of moose and white-tailed deer". In: *Journal of animal ecology* 67.4, pp. 537–543.

Prangle, D. (2017). "Adapting the ABC distance function". In: *Bayesian Analysis* 12.1, pp. 289–309.

— (2018). "Summary statistics". In: *Handbook of Approximate Bayesian Computation*. Chapman and Hall/CRC, pp. 125–152.

Prangle, D., R. G. Everitt, and T. Kypraios (2018). "A rare event approach to high-dimensional Approximate Bayesian Computation". In: *Statistics and Computing* 28, pp. 819–834.

Pritchard, J. K. et al. (1999). "Population growth of human Y chromosomes: a study of Y chromosome microsatellites." In: *Molecular biology and evolution* 16.12, pp. 1791–1798.

Quarles, B. M. and D. A. Roach (2019). "Ageing in an herbaceous plant: Increases in mortality and decreases in physiology and seed mass". In: *Journal of Ecology* 107.3, pp. 1409–1418.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: `https://www.R-project.org/`.

Ravindran, S. et al. (2022). "The association between female reproductive performance and leukocyte telomere length in wild Soay sheep". In: *Molecular Ecology* 31.23, pp. 6184–6196.

Rebarber, R., B. Tenhumberg, and S. Townley (2012). "Global asymptotic stability of density dependent integral population projection models". In: *Theoretical Population Biology* 81.1, pp. 81–87.

Rees, M., D. Z. Childs, J. C. Metcalf, et al. (2006). "Seed dormancy and delayed flowering in monocarpic plants: selective interactions in a stochastic environment". In: *The American Naturalist* 168.2, E53–E71.

Rees, M., D. Z. Childs, and S. P. Ellner (2014). "Building Integral Projection Models: a user's guide". In: *Journal of Animal Ecology* 83.3, pp. 528–545.

Rees, M. and S. P. Ellner (2009). "Integral Projection Models for populations in temporally varying environments". In: *Ecological Monographs* 79.4, pp. 575–594.

Rees, M. and K. E. Rose (2002). "Evolution of flowering strategies in *Oenothera glazioviana*: an Integral Projection Model approach". In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 269.1499, pp. 1509–1515.

Roach, D. A. (2012). "Age, growth and size interact with stress to determine life span and mortality". In: *Experimental gerontology* 47.10, pp. 782–786.

Roberts, S. et al. (2013). "Gaussian Processes for time-series modelling". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 371.1984, p. 20110550.

Rodrigues, G. S. et al. (2018). "Inferences on the Acquisition of Multi-Drug Resistance in *Mycobacterium tuberculosis* using Molecular Epidemiological Data". In: *Handbook of Approximate Bayesian Computation*. Chapman and Hall/CRC, pp. 481–511.

Rogers, J. C. (1984). "The association between the North Atlantic Oscillation and the Southern Oscillation in the northern hemisphere". In: *Monthly Weather Review* 112.10, pp. 1999–2015.

Rose, K. E., S. M. Louda, and M. Rees (2005). "Demographic and evolutionary impacts of native and invasive insect herbivores on Cirsium canescens". In: *Ecology* 86.2, pp. 453–465.

Ruiz-Suarez, S. et al. (2020). "Using approximate Bayesian inference for a "steps and turns" continuous-time random walk observed at regular time intervals". In: *PeerJ* 8, e8452.

Salguero-Gomez, R. (2011). "Physiological bases of plant shrinkage and its demographic implications". PhD thesis. University of Pennsylvania.

Salguero-Gomez, R. et al. (2012). "A demographic approach to study effects of climate change in desert plants". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 367.1606, pp. 3100–3114.

Salguero-Gómez, R. and B. B. Casper (2011). "A hydraulic explanation for size-specific plant shrinkage: developmental hydraulic sectoriality". In: *New Phytologist* 189.1, pp. 229–240.

Salguero-Gómez, R., R. P. Shefferson, and M. J. Hutchings (2013). "Plants do not count... or do they? New perspectives on the universality of senescence". In: *Journal of Ecology* 101.3, pp. 545–554.

Sato, J. (2023). "Hand-Eye Calibration via Linear and Nonlinear Regressions". In: *Automation* 4.2, pp. 151–163.

Schälte, Y., E. Alamoudi, and J. Hasenauer (2021). "Robust adaptive distance functions for approximate Bayesian inference on outlier-corrupted data". In: *bioRxiv*.

Schälte, Y. and J. Hasenauer (2022). "Informative and adaptive distances and summary statistics in Sequential Approximate Bayesian Computation". In: *bioRxiv*.

Schulz, E., M. Speekenbrink, and A. Krause (2018). "A tutorial on Gaussian Process Regression: Modelling, exploring, and exploiting functions". In: *Journal of Mathematical Psychology* 85, pp. 1–16.

Scranton, K., J. Knape, and P. de Valpine (2014). "An Approximate Bayesian Computation approach to parameter estimation in a stochastic stage-structured population model". In: *Ecology* 95.5, pp. 1418–1428.

Seabold, S. and J. Perktold (2010). "statsmodels: Econometric and statistical modeling with Python". In: *9th Python in Science Conference*.

Semenova, E., M. Cairney-Leeming, and S. Flaxman (2023). "PriorCVAE: scalable MCMC parameter inference with Bayesian deep generative modelling". In: *arXiv preprint arXiv:2304.04307*.

Semenova, E., Y. Xu, et al. (2022). "PriorVAE: encoding spatial priors with variational autoencoders for small-area estimation". In: *Journal of the Royal Society Interface* 19.191, p. 20220094.

Siciliano, B., O. Khatib, and T. Kröger (2008). *Springer handbook of robotics*. Vol. 200. Springer.

Silk, D., S. Filippi, and M. P. Stumpf (2013). "Optimizing threshold-schedules for Sequential Approximate Bayesian Computation: applications to molecular systems". In: *Statistical applications in genetics and molecular biology* 12.5, pp. 603–618.

Simmonds, E. G. and T. Coulson (2015). "Analysis of phenotypic change in relation to climatic drivers in a population of Soay sheep *Ovis aries*". In: *Oikos* 124.5, pp. 543–552.

Simola, U. et al. (2021). "Adaptive Approximate Bayesian Computation tolerance selection". In: *Bayesian analysis* 16.2, pp. 397–423.

Sisson, S. and Y. Fan (2018). "ABC samplers". In: *Handbook of Approximate Bayesian Computation*, pp. 87–123.

Sisson, S. A. and Y. Fan (2010). "Likelihood-free Markov Chain Monte Carlo". In: *arXiv preprint arXiv:1001.2058*.

Sisson, S. A., Y. Fan, and M. A. Beaumont (2018). "Overview of ABC". In: *Handbook of Approximate Bayesian Computation*. Chapman and Hall/CRC, pp. 3–54.

Sisson, S. A., Y. Fan, and M. M. Tanaka (2007). "Sequential Monte Carlo without likelihoods". In: *Proceedings of the National Academy of Sciences* 104.6, pp. 1760–1765.

— (2009). "Correction: Sequential Monte Carlo without likelihoods". In: *Proceedings of the National Academy of Sciences* 106.39, pp. 16889–16890.

Stearns, S. C. (1989). "Trade-offs in life-history evolution". In: *Functional ecology* 3.3, pp. 259–268.

Struckman, S. et al. (2019). "The demographic effects of functional traits: an Integral Projection Model approach reveals population-level consequences of reproduction-defence trade-offs". In: *Ecology Letters* 22.9, pp. 1396–1406.

Stubberud, M. W. et al. (2019). "Effects of size- and sex-selective harvesting: An Integral Projection Model approach". In: *Ecology and Evolution* 9.22, pp. 12556–12570.

Tavaré, S. et al. (1997). "Inferring coalescence times from DNA sequence data". In: *Genetics* 145.2, pp. 505–518.

Team, S. K. S. S. P. (2024). *The St Kilda Soay Sheep Project*. Accessed: 2024-07-31. URL: `https://soaysheep.bio.ed.ac.uk/data-collection`.

Tishkoff, S. A. et al. (2001). "Haplotype diversity and linkage disequilibrium at human G6PD: recent origin of alleles that confer malarial resistance". In: *Science* 293.5529, pp. 455–462.

Tomczak, J. M. (2021). "Deep Generative Modeling for Neural Compression". In: *Deep Generative Modeling*. Springer, pp. 173–188.

Toni, T. et al. (2009). "Approximate Bayesian Computation scheme for parameter inference and model selection in dynamical systems". In: *Journal of the Royal Society Interface* 6.31, pp. 187–202.

Van Rossum, G. and F. L. Drake Jr (1995). *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam.

Vindenes, Y., S. Engen, and B.-E. Sæther (2008). "Individual heterogeneity in vital parameters and demographic stochasticity". In: *The American Naturalist* 171.4, pp. 455–467.

Virtanen, P. et al. (2020). "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17, pp. 261–272.

Wall, J. D. (2000). "A comparison of estimators of the population recombination rate". In: *Molecular Biology and Evolution* 17.1, pp. 156–163.

Wegmann, D., C. Leuenberger, and L. Excoffier (2009). "Efficient Approximate Bayesian Computation coupled with Markov Chain Monte Carlo without likelihood". In: *Genetics* 182.4, pp. 1207–1218.

Williams, C. K. and C. E. Rasmussen (2006). *Gaussian Processes for machine learning*. Vol. 2. MIT press Cambridge, MA.

Williams, C. K. and D. Barber (1998). "Bayesian classification with Gaussian Processes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.12, pp. 1342–1351.

Williams, J. L., T. E. Miller, and S. P. Ellner (2012). "Avoiding unintentional eviction from Integral Projection Models". In: *Ecology* 93.9, pp. 2008–2014.

Wood, S. N. (2001). "Partially specified ecological models". In: *Ecological Monographs* 71.1, pp. 1–25.

Wright, S. (1949). "The genetical structure of populations". In: *Annals of eugenics* 15.1, pp. 323–354.

Zhang, J. et al. (2017). "Linking individual-based and statistical inferential models in movement ecology: A case study with black petrels (*Procellaria parkinsoni*)". In: *Ecological modelling* 360, pp. 425–436.

Zou, S. et al. (2009). "Holocene natural rhythms of vegetation and present potential ecology in the Western Chinese Loess Plateau". In: *Quaternary International* 194.1-2, pp. 55–67.