Erasmus School of Economics

Erasmus MC University Medical Center Rotterdam

Master Thesis Econometrics and Management Science
Business Analytics and Quantitative Marketing

Comparative simulation study on calibrating MISCAN-colon using ABC-SMC with adaptive multi-dimensional tolerance updating

Anne de Weerdt
412029

28th of October 2019

Supervisor
Shadi Sharif Azadeh PhD

Second assessor
Martina Zaharieva PhD

Supervisors Erasmus Medical Center
Iris Lansdorp-Vogelaar PhD

Steffie Naber PhD

ABSTRACT

This study aimed to investigate the performance of approximate Bayesian computation as a method for calibrating stochastic micro-simulation models and compare it to an algorithm used in current practice. We implemented Approximate Bayesian Computation Sequential Monte Carlo (ABC-SMC) with adaptive multi-dimensional tolerance updating and used this to estimate two colorectal cancer stool test sensitivities. As a baseline algorithm for comparison we implemented Nelder-Mead calibration extended for stochastic models. The model calibrated in this study is MISCAN-colon, developed in collaboration between Memorial Sloan-Kettering Cancer Center and Erasmus Medical Center. Calibration data was simulated with this same model. Our implementation of ABC-SMC turned out to improve the accuracy, efficiency and consistency of calibration significantly compared to Nelder-Mead.

KEYWORDS

Approximate Bayesian Computation Sequential Monte Carlo, ABC-SMC, Calibration, disease modeling, micro-simulation models, MISCAN, colorectal cancer

TABLE OF CONTENTS

1. INTRODUCTION

In the field of public health, stochastic micro-simulation models are becoming increasingly important (Stout, Knudsen, Kong, McMahon, & Gazelle, 2009). They combine the results of clinical studies in order to simulate trials that are not performed in reality (C. Y. Kong, McMahon, & Gazelle, 2009). Moreover, employing these models reduces the amount of clinical trials necessary to make informed decisions about certain policy changes, by exploring infinitely many scenarios which would be unfeasible and/or unethical without simulation.

However, stochastic micro-simulation models are heavily dependent on the definition of the corresponding input parameters. Accordingly, reliable parameters are crucial to make a proper model. Furthermore, as often a lot of model parameters need to estimated, it is very important to use an appropriate estimation method. This thesis focusses on the MISCAN-colon simulation model, which is used to test different screening strategies for colorectal cancer and aims to provide insight into the calibration of the input parameters. We propose a Bayesian inspired estimation method and compare it to an estimation method commonly used in practice.

In the remainder of this introduction, we first provide more information about the MISCAN-colon simulation model (Section 1.1). In consecutive order, the aim, input parameters, applications and model quality are treated. After this we discuss what current practice of model calibration looks like (Section 1.2) followed by the contribution of this research (Section 1.3 and 1.4).

1.1. MISCAN-colon Simulation Model

In this research we focus on the calibration of the input parameters of MISCAN-colon, short for MIcrosimulation SCreening ANalysis colorectal cancer. This specific stochastic micro-simulation model simulates the lifetime of a variety of individuals and the (possible) colorectal cancer development in these individuals over their lifetime. Moreover, the model also simulates multiple screening strategies to detect cancer early or prevent cancer altogether. A possible screening strategy could, for instance, be inviting all males and females aged 50 to 75 for stool testing every two years. The MISCAN-colon model is thus used to compare the performance of different colorectal cancer screening strategies. This is done by means of cost-effectiveness analysis, namely finding screening strategies that improve the balance between costs and gains compared to a scenario without screening. An example of costs versus gains is the amount of money spent on screening and treatment versus the expected (healthy) life years due to screening.

With respect to input parameters, the MISCAN-colon model has numerous, such as population size, life expectancies, colorectal cancer risks, screening participation rates and screen test

sensitivities. These input parameters can be chosen in various ways. Firstly, some parameters are chosen by the researcher to define the screening analysis such as the population size, screening participation rates and screening ages. The other parameters can be further divided into two groups; observable and non-observable. Choosing values for the parameters belonging to the former group is straightforward, as they arise from observational data such as national registries providing birth and death rates. The latter group of parameters need to be estimated by simulation and are therefore the focus of this thesis. Specifically we investigate screen test sensitivity, which is the probability to obtain a positive screen test result in a person that actually has the disease for which he/she was screened. Subsequently, when parameters are set, we can run simulations with the MISCAN-colon model resulting in outputs containing the complete life and disease histories of all individuals with and without screening. For cost-effectiveness analyses these outputs are aggregated to a population level resulting in model outcomes like the total number of cancers found and the number of years lived in a healthy state.

MISCAN-colon is one of the three colorectal cancer simulation models included in the Cancer Intervention and Surveillance Modeling Network of the US National Cancer Institute and is regularly employed by international cancer prevention institutions like US Preventive Services Task Force, American Cancer Society, Cancer Care Ontario (Canada) and Cancer Council Western Australia to help with evaluation of screening programs and setting up or altering guidelines (e.g. Zauber et al. (2008), Cenin, St John, Ledger, Slevin, and Lansdorp-Vogelaar (2014), Goede et al. (2017), Meester et al. (2018)). Research using MISCAN-colon already played a great role in implementing the national colorectal cancer screening program in The Netherlands (e.g. van Hees et al. (2015)) for which the phased roll-out started in 2014.

As mentioned above, the reliability of these kind of analyses are heavily dependent on the quality of the stochastic micro-simulation model. One of the measures to asses model quality is represented by its ability to replicate real-world populations, which is called validation. The choice of model parameters is key in obtaining these realistic model outcomes. Since many input parameters are not straightforward and observable, they need to be estimated. This estimation is generally done by means of calibration, namely searching for parameter values which generate model outcomes that match observed data, called calibration targets. These calibration targets result from data of clinical trials and similarly model validation is done by matching model behavior to observed data from clinical trials (e.g. C. M. Rutter et al. (2016)). Correct calibration is vital for validation of the model (Stout et al., 2009).

1.2. Current Practice

As multiple input parameters are estimated, a level of uncertainty is added into the MISCAN model. Therefore, it is necessary to perform probabilistic sensitivity analysis on the estimated parameters to analyze the uncertainty in the cost-effectiveness results caused by uncertainty in the parameter estimates. However, without distributional assumptions on the parameter estimates this is not possible, hence only few cost-effectiveness studies based on stochastic micro-simulation cancer models incorporate probabilistic sensitivity analysis in their research (Stout et al., 2009).

Complex models like MISCAN-colon are treated like black box methods in calibration, meaning that only the in- and output of the simulation are used and no knowledge of the functional form of the model itself (Pflug, 2012). Therefore derivative-free techniques like Nelder-Mead, but also Grid Search, Simulated Annealing and Genetic algorithm are used for such cancer simulation models (Stout et al., 2009). Currently, MISCAN-colon is calibrated employing the Nelder-Mead Simplex Method (H. Neddermeijer, van Oortmarssen, Piersma, Dekker, & Habbema, 2000). This is a deterministic minimization algorithm that defines a multi-dimensional simplex of solutions that moves around the solution space using different procedures such as reflecting and contracting.

Although Nelder-Mead is frequently used for calibration of stochastic micro-simulation models (H. Neddermeijer et al., 2000), it can struggle with stagnation at non-optimal points (Nocedal & Wright, 2006) and this depends heavily on the values of the starting simplex. Unfortunately, applying sufficient initializations to circumvent this problem is often not feasible due to the high computational intensity associated with complex disease simulation models like MISCAN-colon. Moreover, an even more pressing concern with Nelder-Mead might be its non-stochastic nature, because the algorithm relies heavily on ranking the simplex solutions from best to worst by comparing their simulated model outcomes to observed data (calibration targets). Since each set of model outcomes is based on a simulated population resulting from a single model run, and the stochasticity of the simulation model causes randomness in these model outcomes, consequently the simplex solutions ranking becomes unreliable and, therefore, impossible to order. In practice, calibration with Nelder-Mead is therefore done by using large population sample sizes or by combining populations from multiple model runs. This reduces the randomness in the model outcomes, but will not remove it altogether.

1.3. Bayesian Calibration

To tackle these problems with stochasticity, researchers have widened their range from directed searches and engineering inspired algorithms, to approximate Bayesian approaches for estimating input parameters of stochastic micro-simulation disease models (e.g. C. M. Rutter, Miglioretti, and Savarino (2009), Seigneurin et al. (2011), Whyte, Walsh, and Chilcott (2011), C. Rutter, Ozik, DeYoreo, and Collier (2018)).

Unlike methods as Nelder-Mead, Bayesian approaches do not treat the simulation model as deterministic, instead they acknowledge that model outcomes are stochastic and therefore never hold on to solutions that seemingly give good model outcomes. There are a variety of different approximate Bayesian calibration algorithms, of which all are based on the concept of consecutively drawing parameter values from a (prior) distribution and accepting or rejecting these based on their corresponding simulated model outcomes. This leads to a population of parameter estimates constituting an empirical distribution. A great advantage for cost-effectiveness analysis with cancer simulation models calibrated like this is the possibility to perform aforementioned probabilistic sensitivity analysis.

1.4. Research Questions

Accordingly, Bayesian seems like a promising direction for calibration of stochastic micro-simulation disease models like MISCAN-colon because it is stochastically driven and estimates a distribution over the parameters, thus allowing for probabilistic sensitivity analysis on all calibrated parameters. For these reasons we implement an approximate Bayesian algorithm for calibration of MISCAN-colon and compare its performance to Nelder-Mead calibration leading us to the following research questions:

1. Can approximate Bayesian calibration improve accuracy of MISCAN-colon parameters compared to Nelder-Mead calibration?
2. Does approximate Bayesian calibration give more consistent estimates than Nelder-Mead calibration?
3. How does the computational demand of approximate Bayesian calibration compare to Nelder-Mead calibration?

To answer these research questions we implement two algorithms for the (partial) calibration of MISCAN-colon. As benchmark algorithm we implement Nelder-Mead and as approximate Bayesian algorithm we implement ABC-SMC, short for Approximate Bayesian Computation – Sequential Monte Carlo. The performance of these two algorithms are compared by means of a

simulation study. That is, we simulate fictive datasets using MISCAN-colon with pre-defined parameters, which are then used as calibration targets to estimate back these parameters.

The remainder of this paper will be structured as followed. Section 2 describes the general concept of parameter calibration, introduces relevant notation and terminology, and briefly describes the simulation model MISCAN-colon. After this an overview of previous literature regarding different approximate Bayesian methods is given and motivation for choosing ABC-SMC (Section 3). Then in Section 4 our implementation of ABC-SMC is described as introduced in Section 3. Here we also elaborate on the Nelder-Mead implementation and the details of the simulation comparison study. Lastly the results (Section 5) and conclusions (Section 6) are presented.

2. CONTEXT DESCRIPTION

This section first describes the general concept of calibration in the context of a cancer screening simulation model and introduces some relevant symbols and notation useful for following the explanation of the calibration algorithms (Section 3 and 4). Table 1 below gives an overview of these symbols, the order is based on order of introduction. Besides this, Appendix 8.1 contains a list of abbreviations used in this thesis. Next we give a description of the MISCAN-colon model and explain relevant parameters and model outcomes.

Table 1. Overview of Symbols and Notation

Notation/Symbol		Meaning
$\bar{\mu} = [\mu_1, \dots, \mu_J]'$	=	Vector with calibration targets
$\bar{\theta} = [\theta_1, \dots, \theta_K]'$	=	Vector with model parameters
$M(\bar{\theta})$	=	Model outcomes resulting from one model run with input parameters $\bar{\theta}$
$f_j(\bar{\theta})$	=	Function that summarizes model outcomes from a model run with input parameters $\bar{\theta}$ into a numeric value on the same scale as calibration target j
$\bar{s}$	=	$[f_1(\bar{\theta}), \dots, f_J(\bar{\theta})]'$
ε	=	(final) tolerance, the maximum amount of distance/deviance allowed between function evaluation and calibration target

2.1. Introducing Calibration

The aim of calibration is to find values of the model parameters that ensure the simulated model outcomes to closely resemble reality. During calibration we depict this "reality" with observed data which are summarized into calibration targets These J calibration targets are chosen a-priori and denoted by $\bar{\mu} = [\mu_1, \dots, \mu_J]'$. They should be (strongly) influenced by the parameter(s) of interest, for example when calibrating the parameter cancer risk, it makes sense to use the amount of people who have cancer as a calibration target. Calibration targets are usually taken

from clinical studies. The K model parameters are stored in a vector, $\bar{\theta} = [\theta_1, \dots, \theta_K]'$, which will be referred to as a particle. These parameters are used as input for the model. A model run delivers a simulated population of individuals and their disease- and screen history. Simulated outcomes resulting from a single model run with input parameter from particle $\bar{\theta}$ are denoted by $M(\bar{\theta})$.

Keeping in mind the aim of calibration we need to find a way to effectively compare model outcomes with calibration targets. In order to do this we define the set of functions $\{f_j(\bar{\theta})\}^{\langle j=1:J \rangle}$, where $f_j(\bar{\theta})$ aggregates outcomes $M(\bar{\theta})$ into the model predicted estimate of calibration target j. These are referred to as function evaluations and saved into vector $\bar{s} = [f_1(\bar{\theta}), \dots, f_J(\bar{\theta})]'$. A one-to-one mapping can be made between the elements in $\bar{s}$ and $\bar{\mu}$. During calibration candidate particle(s) are generated after which their function evaluation(s) are computed and compared to the calibration target(s), a schematic depiction of this can be seen below in Figure 1.

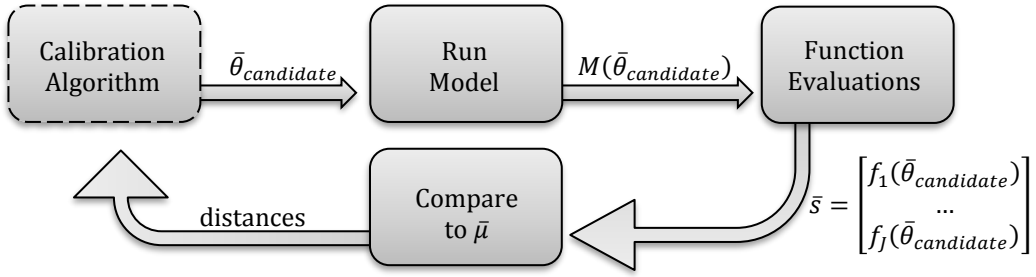

Figure 1. Schematic depiction of general calibration process

Similarity between function evaluations and calibration targets is measured by their distance or deviance. Distance is measured between one calibration target, μ_j, and one element of the function evaluation, s_j. For this we use the absolute distance. The ABC-SMC algorithm implemented in this research directly uses these distances. Nelder-Mead on the other hand takes another step and summarizes the distances into a single number, this is referred to as the deviance. In this research the Euclidean deviance is used;

$$\sqrt{\sum_{j=1}^{J} \left(f_j(\bar{\theta}) - \mu_j\right)^2}. \qquad (1)$$

In calibration we define (final) tolerance(s) denoted by ε for termination of the calibration algorithms. Tolerance indicates the distance/deviance-upper bound that a particle is allowed to have in order to categorize the function evaluation as similar enough to the calibration target.

2.2. MISCAN-colon

It is of great practical use to implement the calibration methods of interest on a complicated stochastic micro-simulation disease model to investigate their practical performance and generalizability. As introduced before, we use the MISCAN-colon model which abbreviates; MIcrosimulation SCreening ANalysis colorectal cancer model. This section gives a brief overview of the model, a detailed description of MISCAN-colon can be found in Loeve, Boer, van Oortmarssen, van Ballegooijen, and Habbema (1999).

MISCAN-colon is a microsimulation model meaning that a population is simulated individual by individual, each with a certain probability to get colorectal cancer. The disease simulation module of MISCAN-colon (natural history module) operates based on the adenoma-carcinoma sequence (Leslie, Carey, Pratt, & Steele, 2002). This means that the development of cancer starts with the onset of small adenomas; (0, 5]mm which grow to medium (5, 10]mm and possibly to large (10,∞)mm adenomas. Cancer state I (i.e. localized disease) can start from medium, or large adenomas and afterwards can progress to cancer states II, III and IV sequentially. The amount of adenomas and their (possible) progression to next (cancer) states is dependent on age. Figure 2 below gives a visualization of the development from adenoma to cancer, here a division is made between benign (not harmful) and malignant (harmful).

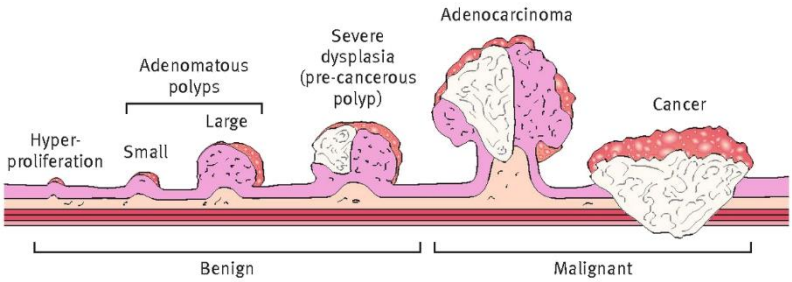

Figure 2. Adenoma carcinoma sequence[a]

2.2.1 MISCAN-colon Natural History Module

In the natural history module of MISCAN-colon the life and disease history without screening are simulated for each individual. The following explanation applies to a single individual. First the date of birth and then the age of death from other causes than colorectal cancer is determined using life expectancy tables from national registries. Then a personal risk index is drawn from a

[a] Adapted from Johns Hopkins Colorectal Cancer Research Center

gamma distribution.[b] For the onset of adenomas we use hazard rates which are computed by multiplying a person's risk index with an age factor. Simulation of cancer starts with determining the onset of small (0, 5]mm adenomas one by one; by sequentially drawing durations from the exponential distribution with mean equal to the hazard rate. This process starts at the age of birth, and ends once the sum of all durations exceeds the age of death of other causes. This step of the natural history module creates a non-homogenous Poisson process because the exponential distribution evolves with age. The location of each adenoma is determined based on proportions over the colorectal area. After determining the onset of adenomas, the model determines the progression to next states. This is done for each adenoma independently and follows a similar process as the simulation of onset of adenomas. At the age of occurrence of the small adenoma, first the next state is determined depending on age and possibly location. Then the duration of this state is drawn from the exponential distribution with state dependent mean. Whenever the age of onset plus the duration falls before the death of other causes, the next state is reached.

2.2.2 MISCAN-colon Screening Module

After the natural history is simulated for all individuals, the screening protocol is applied. This incorporates aspects such as screening participation rates and sensitivity of screen tests (this can be interpreted as the probability to get a positive screen test result conditional on having adenomas or colorectal cancer) as well as other test characteristics. In this research we use the Faecal Immunochemical Test (FIT) which detects haemoglobin in stool indicating possible bleeding of adenomas or cancers in the colon. After positive screen results, individuals undergo disease confirmation possibly followed by diagnosis, treatment and surveillance. In this study disease confirmation is done using a colonoscopy procedure. The positive impact of screening can be two-fold, firstly it can possibly prevent cancer by removing benign lesions (see Figure 2 above) and second it could detect cancer at an earlier stage than symptoms generally start to show improving the chance of a successful treatment.

2.2.3 MISCAN-colon Parameters

Many model parameters are mentioned in the brief description given above (e.g. gamma parameters, age factors, location proportions), some of which are directly based on literature or expert knowledge. Most of the parameters however must be calibrated because they are impossible to observe. Calibrating all parameters is a costly procedure and therefore the calibration is divided into smaller parts. In this research we evaluate calibration of two

[b] The gamma parameters are fixed and chosen a-priori to ensure the appropriate amount of heterogeneity in risk amongst the simulated population

parameters; screen test sensitivity for detecting cancers ($sens_{CRC}$) and screen test sensitivity for detecting large adenomas ($sens_{LAde}$). Realistic calibration targets for these parameters are the number of cancers ($pFIT_{CRC}$) and large adenomas ($pFIT_{LAde}$) found during diagnostic colonoscopy after a positive FIT result. These parameters are assumed to be uncorrelated.

3. LITERATURE REVIEW

Before going into the different Approximate Bayesian Computation (ABC) algorithms we first explain the main reasoning behind them. Bayesian analysis is based on Bayes' theorem (Greenberg, 2012) stating that;

$$p(\bar{\theta}|data) \propto p(\bar{\theta})p(data|\bar{\theta}), \quad (2)$$

where $\bar{\theta}$ is the parameter vector (particle) and p a probability density function. This theorem defines the relationship between the posterior $\left(p(\bar{\theta}|data)\right)$, prior $\left(p(\bar{\theta})\right)$ and likelihood $\left(p(data|\bar{\theta})\right)$. The prior and posterior are both distributions over the parameters but the posterior is conditional on the data and the prior is not. The likelihood tells us how likely it is that the data was obtained with underlying parameter values $\bar{\theta}$. In black-box estimation it is not possible to compute likelihood functions, for this reason ABC methods substitute likelihood evaluation by a comparison between function evaluations and corresponding real-world outcomes (Toni, Welch, Strelkowa, Ipsen, & Stumpf, 2008). In calibration this means that particles in the population must generate function evaluations that are similar to the calibration targets. ABC calibration always starts with a population of particles from the prior distribution and ends with a population of particles that approximate the posterior distribution. Over the last two decades different approaches have been proposed for moving from the prior to the posterior with ABC, below we give a brief overview of this.

3.1. ABC Simple Rejection Algorithm

The complexity and efficiency of ABC methods have evolved over the years. It started with a simple rejection algorithm, of which implementation is very easy (Marjoram, Molitor, Plagnol, & Tavare, 2003); in each iteration you sample a particle from the prior distribution and accept or reject it based on the similarity between function evaluations and calibration targets. This is continued until enough particles are accepted to give a good estimate of the posterior. One of the earliest publications describing an estimation approach like this was by Tavare, Balding, Griffiths, and Donnelly (1997) and Pritchard, Seielstad, Perez-Lezaun, and Feldman (1999).

This simple form of the rejection algorithm is however inefficient (Toni & Stumpf, 2009) because many particles are commonly rejected when the prior and posterior distribution are dissimilar (Toni et al., 2008). This inefficiency can be partially overcome by taking advantage of parallel computation, possible because all draws from the prior distribution are independent (Marjoram et al., 2003). Nevertheless, choosing good priors can be difficult especially for stochastic disease simulation models like MISCAN-colon. This is because the parameters to be estimated are often unobserved, and quality expert opinion about their possible values are lacking. We therefore have to resort to alternative approaches such as using previous calibration results to form the priors, as done in C. Rutter et al. (2018).

3.2. ABC Markov Chain Monte Carlo

The next phase of ABC algorithms incorporates likelihood free Markov Chain Monte Carlo (MCMC) in an attempt to extend the rejection algorithm into a more efficient algorithm (Drovandi & Pettitt, 2011). This was introduced by Marjoram et al. (2003) and called ABC-MCMC. It is based on the Metropolis-Hastings MCMC algorithm and adjusted such that no likelihoods have to be estimated. In this method a dynamic (non-symmetric) proposal distribution over the parameters (conditional on the parameters of the previous iteration) is used to sample particles from the approximate posterior. In each iteration we sample one particle from this proposal distribution and when this delivers function evaluations that are close enough to the calibration targets, it is accepted with some probability; this probability is based on the ratio between the approximate posterior probability of the new particle versus the old particle.

Even though (when using non-symmetric proposal distributions) ABC-MCMC has the advantage over the rejection algorithm in that it relies less on the prior distribution, it still has a major drawback; the highly correlated nature of the sample from the proposal distribution (Sisson, Fan, & Tanaka, 2007; Toni & Stumpf, 2009). For this reason parallel computation is not an option. Besides this disadvantage, Toni et al. (2008) and Sisson et al. (2007) also point out that because of potentially low acceptance rates this algorithm may stay in non-optimal regions too long.

3.3. ABC Sequential Monte Carlo

In attempt to solve the above mentioned problems, Sisson et al. (2007) created a framework for ABC using Sequential Monte Carlo (SMC). The main idea of the SMC approach is to move from the prior to the approximate posterior through a series of intermediate distributions. Similar as in before mentioned methods, all intermediate distributions are represented by their empirical distribution; a population of N particles. Different approaches to sequential updating exist, but

the general framework agrees across algorithms, this will be explained following the schematic depiction in Figure 3 below.

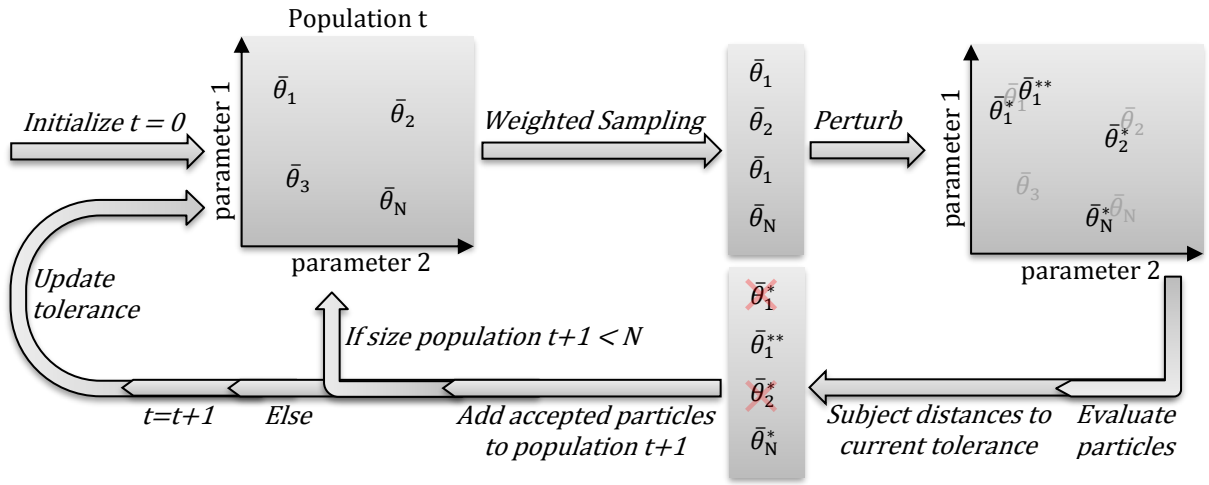

Figure 3. Depiction of ABC-SMC algorithm (N = 4)

First the algorithm initializes a population of N particles using the prior distribution (population t = 0). Then in each consecutive iteration a new population is generated. This is done by performing weighted sampling from the previous population, where the weights depend on the prior and likelihood.

After being sampled from the previous distribution particle are adjusted. This adjusting of the particles is called perturbation and the aim is to create a population of particles of which the parameter values are close to those of the previous population but not identical. Perturbation is an essential step in ABC-SMC because we are estimating real valued parameters and do not wish to be stuck with only the parameter values of the initial population. After perturbation of each particle their function evaluations are obtained and the distances are computed (see Figure 1). Only particles that have distances within the current tolerance ($\bar{\varepsilon}_t$) are accepted into the next population (t+1). This process continues until N particles are accepted into the next population. In each new iteration the tolerance becomes more stringent. Termination of the algorithm is achieved when the final tolerance level is reached.

Within one iteration of ABC-SMC the weighted sampling and perturbation is only dependent on the previous population, allowing for parallel computation in each iteration. Because of this advantage we decide to narrow our focus to ABC-SMC. In most algorithms describing ABC-SMC, one-dimensional tolerance is used (e.g. Toni and Stumpf (2009), Filippi, Barnes, and P.H. (2011), Liepe et al. (2014)). This means that only one tolerance level is defined regardless of the amount of calibration targets. This forces us to summarize the distances between calibration targets and function evaluations into a single value (deviance) as is done in Nelder-Mead (see Section 4.2).

With one-dimensional tolerances, calibration allows some function evaluations to be further away from their target than others, and leaves no control over this whatsoever. Beside this, it can be difficult to find an informed method of summarizing the different distances into a single deviance. For these reasons we decide to incorporate multi-dimensional tolerances, this means that tolerance levels are defined for each calibration target separately and distances are evaluated individually.

Besides one-dimensional tolerances, often the maximum number of iterations (T) and the corresponding tolerance schedule ($\varepsilon_1 \dots \varepsilon_T$), which heavily impact the performance of the algorithm (Moral, Doucet, & Jasra, 2012), are predefined. In order to minimize the amount of predefined constants and allow for a more natural tolerance schedule we implement adaptive updating, inspired by research of Drovandi and Pettitt (2011) and C. Rutter et al. (2018).

Lastly we want to tackle the issue of particle degeneration, meaning that many particles are assigned very small weights, causing a small group or even one particle to dominate the population. Even though the resampling and perturbation in each iteration helps prevent degeneracy (Drovandi & Pettitt, 2011), it seems like a good idea to monitor the quality of the solution throughout the algorithm. We do this by means of the effective sample size and coefficient of variation inspired by the approach of Moral et al. (2012). Whenever this criterion detects degeneracy, the population is resampled after which the weights are reset. Full details on our implementation of ABC-SMC will be given next (Section 4.1).

4. METHODOLOGY

This section first gives a detailed description of our implementation of the ABC-SMC algorithm followed by the benchmark algorithm Nelder-Mead. Then details are given on the simulation comparison set-up. Both algorithms were implemented in R.

4.1. Approximate Bayesian Computation – Sequential Monte Carlo

This section describes our implementation of ABC-SMC following the general pseudocode shown below in Figure 4. Full pseudocode can be found in Appendix 8.2 (Algorithm 1).

Step 1 Initialization (pseudocode lines 1:2)

The first step of ABC-SMC is to generate an initial population by drawing N particles from the prior distribution.

ALGORITHM ABC-SMC

INPUT

Posterior size (N), prior distribution, perturbation kernel, initial tolerance, final tolerance, tolerance updating percentile, CV^2 threshold

OUTPUT

Approximate empirical posterior distribution $\left(\{\bar{\theta}_i\}^{(i=1,\dots,N)}\right)$

PROCEDURE

1: Initialize population of N particles
2: $t \leftarrow 1$
3: Sample particles from population $t-1$
4: Perturb particles, get function evaluations, compute distances
5: Accept particles that have distances within tolerance and compute their weights
6: Update tolerance level
7: Compute CV^2 of weights, if it exceeds threshold then resample.
8: $t \leftarrow t+1$
9: Go to 3 if termination is not reached

Figure 4. Short pseudocode ABC-SMC

Step 2 Weighted Sampling and Perturbation (pseudocode lines 3:4)

In each consecutive iteration a new population of particles is created. This is done by sampling from the previous population. Intermediate particle populations are not identically distributed, therefore we use weighted sampling. Afterwards the sampled particles are perturbed using a perturbation kernel and then their function evaluations are generated, after which the distances to each target are computed. Perturbation of a particle can be interpreted as adding a bit of noise to the particle values, in our case the magnitude of this noise is determined by the covariance matrix of the perturbation kernel. The weight of a particle i in iteration t is determined using the following formula;

$$w_i^t \propto \frac{\pi\left(\bar{\theta}_i^t\right)}{\sum_{m=1}^{N} w_m^{t-1} K\left(\bar{\theta}_m^{t-1} \rightarrow \bar{\theta}_i^t\right)}, \tag{3}$$

where $\pi\left(\bar{\theta}_i^t\right)$ gives the prior density of particle $\bar{\theta}_i^t$, and $K\left(\bar{\theta}_m^{t-1} \rightarrow \bar{\theta}_i^t\right)$ the kernel density of moving from particle $\bar{\theta}_m^{t-1}$ to $\bar{\theta}_i^t$ (Sisson, Fan, & Tanaka, 2008). As perturbation kernel we use a truncated multivariate normal distribution with a variable mean (mean is equal to the particle that is being perturbed), a covariance matrix equal to the weighted sample covariance of the particle population from the previous iteration and lower- and upper bounds corresponding to the parameters of interest. After all weights of a population are computed they are standardized such that $\sum_{i=1}^{N} w_i^t = 1$. The weighting formula (3) can be interpreted as the ratio between prior and current knowledge. The numerator gives the relative probability of obtaining the particle based on prior knowledge alone, and the denominator gives the relative probability of obtaining the particle from perturbation of the previous population, while satisfying the current tolerance restriction. Whenever a proposed particle is on the outskirts of the previous population, but has enough prior support, it will be given a larger weight. This intends to prevent ABC-SMC from

getting stuck in local optima. Particles located in more dense areas of the population will obtain lower weights because they add relatively little information as there are many other particles like them.

Step 3 Acceptance/Rejection and Tolerance Updating (pseudocode lines 5:6)

Only those particles that give function evaluations similar enough to the calibration targets are accepted. Acceptance of particles is determined using a tolerance that starts at infinity for each calibration target, and is iteratively updated by taking the τ^{th} percentile of the distances. This approach makes sure that the tolerances decrease in such a way that approximately τ% of the particles in the previous population would still be accepted into the next population. Depending on the choice of τ, this could help prevent unwanted drops in the acceptance rate. Choosing τ too low could lead to low acceptance rates causing iterations to need many model evaluations. Even though high values of τ ensure a good acceptance rate, they can also cause the algorithm to move very slowly, again requiring many model runs. For the main analysis we use the 50th percentile for updating and vary this in the sensitivity analysis to the 25th and 75th percentile.

Step 4 Population evaluation and possible resampling (pseudocode lines 7:9)

When many particles are assigned very small weights, commonly referred to as particle degeneracy, this causes a small group or even one particle to dominate the population. Preferably we want all N particles in the population to add significant information to the estimation such that we have a perfect sample from the posterior. In practice however the so called effective sample size (ESS) is hardly ever equal to N and therefore we must be satisfied with values of the ESS that are high enough. The ESS can take on values 1 till N. Low values of the ESS indicate particle degeneracy causing bad accuracy of the estimator (Moral et al., 2012), when this happens we force the algorithm to resample from the particle population after which all the weights are reset to be equal. After this procedure a new iteration proceeds. We use the squared coefficient of variation (CV^2) over the weights (A. Kong, Liu, & Wong, 1994) to evaluate the ESS indirectly. This criterion directly links to the effective sample size; $ESS = \frac{N}{1+CV^2}$, such that values of CV^2 equal to $\left(3, 1, \frac{1}{3}\right)$ ensure an ESS of respectively 25, 50 and 75 percent of the total population (see derivation in Appendix 8.5). For the main analysis we require an ESS of at least half of the population size, and as sensitivity analyses we use the more conservative 75% and also the lower 25%.

Step 5 Termination

Termination is achieved when the final tolerances are reached, when the maximum number of model runs is exceeded or when no particles have been expected for 5 consecutive iterations.

Termination also happens when in the final iteration resampling is done 6 times but the degeneracy criterion is still not satisfied. Final tolerances can be determined by bootstrapping the available data after which you compute summary statistics such as e.g. the standard deviations of the calibration targets and use these as terminal tolerances. ABC-SMC returns a population of N (different) particles, $\{\bar{\theta}_i\}^{\langle i=1:N\rangle}$, with corresponding weights; $\{w_i\}^{\langle i=1:N\rangle}$ as solution.

4.2. Benchmark Calibration – Nelder-Mead

Currently MISCAN-colon models are calibrated using the Nelder-Mead simplex method. Research of H. Neddermeijer et al. (2000) proposed a variety of adjustments and extensions to the original Nelder-Mead algorithm to better suit calibration of stochastic simulation models. In order to create a baseline that represents current practice in the most optimal way we use the Nelder-Mead algorithm with recommended adjustments and extensions from H. Neddermeijer et al. (2000). In the next paragraph we shortly describe the algorithm, for full pseudocode see Appendix 8.2 (Algorithm 2).

Nelder-Mead starts with a simplex of K + 1 particles, which are either randomly generated or pre-defined. Throughout the algorithm these particles are ordered based on their deviance. In each iteration, the goal is to replace the worst particle ($\bar{\theta}_{worst}$), the one with the largest deviance, by a better one. This is done by proposing replacement candidates. New particles are created using four possible procedures. First of all, when looking for a good candidate it is intuitive to move in the direction opposite of the worst particle, this is done by reflecting the simplex away from the worst particle. On the other hand it also makes sense to search towards a good particle, this is done by expanding the simplex in that direction. Whenever Nelder-Mead fails to find better solutions it retreats to previously found good solutions by diminishing the simplex. This can be done in two ways; either the whole simplex is shrunk towards the best particle, or the simplex is contracted by moving a particle closer towards the center.

Each of the above mentioned approaches use the following formula: $(1 - \mathrm{c})\bar{\theta}_a + \mathrm{c}\bar{\theta}_b$, where c is an a priori defined coefficient specific to the different procedures. For all procedures except shrinkage, $\bar{\theta}_a$ represents the centroid, which is computed by averaging parameter values over the K best particles in the simplex. For shrinking, $\bar{\theta}_a$ represents the best particle so far. Particle $\bar{\theta}_b$ differs depending on the steps of the algorithm.

For reflection ($c = -1$), expansion ($c = 0.5$) and contraction ($c = 2$) of particle $\bar{\theta}_b$, coefficients recommended by Nelder and Mead (1965) are used. Lastly for shrinking we adopt the coefficient $c = 0.9$ and add a re-evaluation of the best particle as used in H. Neddermeijer et al. (2000),

recommended by multiple studies investigating Nelder-Mead minimization performance on stochastic objective functions (Barton and Ivey (1996) Tomick, Arnold, and Barton (1995), Humphrey and Wilson (2000), H. G. Neddermeijer, Piersma, van Oortmarssen, Habbema, and Dekker (1999)).

At the beginning of every iteration t the original candidate, $\bar{\theta}_t^{\langle original \rangle}$, is computed by reflecting the worst particle ($\bar{\theta}_b = \bar{\theta}_{worst}$) through the centroid. The best-case scenario is that this candidate gives the lowest deviance of all K + 1 solutions. Then we try to improve the solution even more by expanding it $\left(\bar{\theta}_b = \bar{\theta}_t^{\langle original \rangle}\right)$ through the centroid. Whichever performs better is chosen as replacement. In the second-best case, $\bar{\theta}_t^{\langle original \rangle}$ delivers a solution that is not the best nor the worst (better than the two worst solutions) and is therefore immediately accepted.[c] A third option is that $\bar{\theta}_t^{\langle original \rangle}$ ranks between the worst and second-to-worst, it $\left(\bar{\theta}_b = \bar{\theta}_t^{\langle original \rangle}\right)$ is then contracted through the centroid. Whichever is better is chosen as replacement. In case this turns out to be $\bar{\theta}_t^{\langle original \rangle}$, subsequently the whole simplex is shrunk towards best solution. Lastly, in the worst-case scenario $\bar{\theta}_t^{\langle original \rangle}$ gives no improvement, then a new candidate is generated by contracting the worst particle ($\bar{\theta}_b = \bar{\theta}_{worst}$) through the centroid. If this improves the worst particle, it is accepted, otherwise the whole simplex is shrunk.

4.2.1 Evaluation and Noise Reduction Action

Following the recommendation of H. Neddermeijer et al. (2000) we add the Dominant Noise – Increase Replications (DN-IR) extension. This approach requires that we run the model multiple times for evaluation of a particle, resulting in multiple deviances which are then averaged. An advantage that comes with this approach is the opportunity to perform the model runs in parallel. The number of replications used at the start of the algorithm is chosen a priori and denoted by R_0. After every iteration the dominant noise criterion is computed, possibly resulting in a noise reduction action. The dominant noise criterion checks the null-hypothesis stating that the deviances of the particles in the simplex are identical. To test this an F-test with degrees of freedom (K, $(R_r - 1)(K + 1)$) and significance level λ is performed, where R_t is the number of replications used in iteration t. We take the level of λ to be 10%. If the null hypothesis is not rejected then this gives reason to suspect that the calibration process is too much influenced by simulation noise. In order to decrease this noise, subsequently R_t is increased by multiplying with an a priori defined constant $R_{increase}$, to a maximum of R_{max} which is also a priori defined. For $R_{increase}$ we use 1.25. Termination of the Nelder-Mead algorithm can be varied, for this analysis

[c] This only applies when at least two parameters are being calibrated

termination is reached when the terminal tolerance (ε) of the deviance is obtained, when the maximum amount of model runs is exceeded or when the parameter estimates have not changed for five consecutive iterations. Nelder-Mead returns the best particle of the final simplex as solution.

In the previous sections both calibration algorithms implemented in this study were explained and some differences between them were already highlighted. An overview of the most important differences between ABC-SMC and Nelder-Mead are shown below in Table 2.

Nelder-Mead	ABC-SMC
Treats function evaluations as *deterministic*	Treats function evaluations as *stochastic*
Estimates *a single point estimate*	Estimates *an empirical distribution*
Moves one *particle at a time*	Moves one *distribution at a time*
Creates function evaluations with model population size *larger than* observed data	Creates function evaluations with model population size *equal to* observed data
Incorporates prior information *only at start of* algorithm	Incorporate prior information *throughout* algorithm
Holds on to solutions	*Samples and perturbs* solutions
Judge particles based on *deviance*	Judge particles based on *individual distances*

Table 2. Overview differences Nelder-Mead versus ABC-SMC

4.3. Simulation Study Set-up

In this comparison study we compare two calibration algorithms and their ability to accurately and efficiently estimate MISCAN-colon parameters. To do so we first create fictive calibration targets in the pre-processing phase. Then, in the analysis phase we calibrate the model on these calibration targets using the different calibration algorithms and settings. Lastly in the post-processing phase the calibration results are summarized and compared. Detailed descriptions of these three phases are given below.

4.3.1 Pre-processing

As mentioned in Section 2.2.3Table 2 we calibrate two model parameters (K=2). Both parameters are sensitivities of the FIT stool screen test for colorectal cancer. One parameter gives the sensitivity specific for large adenomas ($sens_{LAde}$) and the other for cancer stages I-IV combined ($sens_{CRC}$). The true underlying values are decided to be 0.159 and 0.700 respectively. Using these parameter values we simulate different MISCAN-colon populations by varying the random seed. The simulated model outcomes, $pFIT_{LAde}$ and $pFIT_{CRC}$ resulting from a MISCAN-colon population are treated as calibration targets as if obtained from real world studies. For each analysis we create 20 independent populations referred to as datasets and denoted by $D = 20$. There will be

separate calibrations for each dataset and these calibrations are all repeated 5 times indicated by $C = 5$.

To prevent non-positive values of the relevant model outcomes, $pFIT_{LAde}$ and $pFIT_{CRC}$, we require a MISCAN-colon population sample size of at least 10,000. For the main analysis we use a ten-fold higher sample size (100,000), and in order to investigate the performance of the calibration algorithms on datasets with different levels of uncertainty due to sample size we create datasets using sample sizes 10,000 and 1,000,000 respectively. For another sensitivity analysis, which is explained further on, we create 80 populations using the same sample size as the main analysis (100,000). The resulting datasets are then added to those of the main analysis resulting in $D = 100$ datasets. Together this leads to a total of 140 separate datasets. All other parameters in MISCAN-colon remain constant throughout the calibration, details on this can be found in Appendix 8.4.

The terminal tolerances used in ABC-SMC are determined by running MISCAN-colon 20,000 times with parameters as mentioned above. We take the resulting standard deviation of the model outcomes as terminal tolerances. This approach is similar to the bootstrapping mentioned in the ABC-SMC algorithm (Section 4.1), but is more suitable for a comparison simulation study like this, as it gives a better estimation of the overall uncertainty in all calibration targets. The above procedure is done separately for the three different population sample sizes.

4.3.2 Analysis

Below we give a description of the calibrations performed in this research, an overview of the calibration algorithm settings can be found in Appendix 8.3.

Main Analysis

For the main analysis we first calibrate $C = 5$ times on each of the $D = 20$ datasets using ABC-SMC resulting in 100 separate calibrations. Termination is achieved when the final tolerances are reached, when the maximum number of model runs are exceeded or when no points have been accepted for 5 consecutive population draws. No prior information about the parameters is assumed and therefore the prior distribution is taken to be standard uniform. The ABC-SMC population size is set at $N = 40$.

Secondly we calibrate in the same fashion as above but now using Nelder-Mead instead of ABC-SMC. For Nelder-Mead we initially require $R_0 = 5$ model runs for one function evaluation, this is equivalent to a population sample size of 500,000 MISCAN-colon individuals. The maximum is set to $R_{max} = 100$ (10,000,000 simulated individuals). Termination is achieved as described in Section

4.2. The terminal tolerance is computed based on the results of ABC-SMC as follows; for each dataset the weighted[d] Euclidean deviance is computed and averaged over the C calibrations.

Sensitivity Analysis

After the main analysis we perform two types of sensitivity analyses; first we change the calibration data and second we change the calibration settings. An overview of the all sensitivity analyses can be seen in Table 3. Sensitivity analyses SA2-SA11 are performed C=5 times on the same D = 20 datasets. Full details on all analyses are given in Appendix 8.3.

Name Analysis{a}	ABC-SMC	Nelder-Mead	Change with respect to the main analysis
SA1	✓		More calibrations on different datasets
SA2	✓	✓	Less reliable data
SA3	✓	✓	More reliable data
SA4	✓	✓	Prior information on parameters
SA5	✓	✓	More prior information on parameters
SA6	✓		Three-fold posterior sample size
SA7	✓		Six-fold posterior sample size
SA8	✓		Less restrictive tolerance updating percentile
SA9	✓		More restrictive tolerance updating percentile
SA10	✓		Less strict weight degeneracy criterion
SA11	✓		More strict weight degeneracy criterion

{a} SA = Sensitivity Analysis

Table 3. Overview Sensitivity Analyses

There are three sensitivity analysis of the first type. Here first calibrations on 80 additional datasets are performed, of which the results are merged with those of the main analysis. This is only done for ABC-SMC.

Then we investigate how the calibration algorithms perform with more and less uncertainty in the calibration targets by using the datasets with MISCAN-colon population sample sizes 10,000 and 1,000,000, this is done for both ABC-SMC as Nelder-Mead. The maximum amount of model runs are adjusted according to sample size.

For the second type of sensitivity analyses we change a number of settings in the ABC-SMC algorithm of which some were already mentioned in the ABC-SMC methodology (Section 4.1). Most importantly we test two scenarios in which we add prior knowledge to the algorithm by modelling a truncated normal prior centered around the true underlying parameter values, one with a large variance (0.04) and another with a smaller variance (0.0025). This is applied to the Nelder-Mead algorithm as follows; the starting simplex is sampled from the initial prior

[d] using weights from weighting formula given in equation (3)

population of ABC-SMC, this is done for each calibration independently. Next we increase the ABC-SMC population size from 40 to 120 and 240. Then we vary the updating percentile and lastly the CV^2 threshold is changed up- and downwards.

4.3.3 Post-processing

After finishing all analyses of both algorithms we determine the accuracy of their point estimates, the estimation consistency and corresponding computational demand. In order to assess the estimation accuracy of ABC-SMC we first compute a point estimate for each calibration. This point estimate is taken to be the estimated mean (location parameter) of the resulting posterior distribution, and is computed by taking the weighted[e] average of the particles in the estimated posterior distribution.

In order to answer the first research question mentioned in Section 1.4 we need to declare a measure for comparing accuracy of the algorithms. For this we first compute the relative absolute errors ($rAE_{k,d,c}$);

$$\frac{\left|\hat{\theta}_k^{\langle d,c\rangle} - \theta_k\right|}{\theta_k}, \forall\, \mathrm{k} = 1, \dots, \mathrm{K}, \text{and } \mathrm{d} = 1, \dots, \mathrm{D} \text{ and } \mathrm{c} = 1, \dots, \mathrm{C}, \tag{4}$$

where D is the number of datasets, C the number of calibrations performed on each dataset, $\hat{\theta}_k^{\langle d,c\rangle}$ is the point estimate of parameter k on dataset d during calibration c and lastly θ_k is the true underlying value of parameter k. These are used to compute the individual parameter accuracy as well as the overall accuracy. The latter is used to allow for a direct straightforward comparison between algorithms. The former is to distinguish estimation behavior between parameters and is computed by averaging the relative absolute errors over all calibrations;

$$\frac{1}{D\,C}\sum_{d=1}^{D}\sum_{c=1}^{C} rAE_{k,d,c}\,, \forall\, k = 1, \dots, K, \tag{5}$$

which will be referred to as rAE_k. For the overall accuracy we use the average relative mean absolute error;

$$\frac{1}{K}\sum_{k=1}^{K} rAE_k, \tag{6}$$

which will be referred to as rMAE.

[e] using weights from weighting formula given in equation (3)

Moving on to the second research question (Section 1.4) we need a measure for comparing consistency of the different calibration algorithms. For this we look at the variation in parameter estimates of calibrations on the same dataset. Since the average parameter estimates and their variation are likely to depend on the dataset it is only fair to standardize the variation per dataset, for this we use the coefficient of variation. In order to allow for a more direct comparison between algorithms we then average these coefficients of variation over de different datasets. This is done as follows;

$$\frac{1}{D}\sum_{d=1}^{D} CV\left(\left\{\hat{\theta}_k^{\langle d,c\rangle}\right\}^{\langle c=1,\dots,C\rangle}\right), \forall\, k = 1,\dots,K, \tag{7}$$

where $CV\left(\left\{\hat{\theta}_k^{\langle d,c\rangle}\right\}^{\langle c=1\dots C\rangle}\right)$ is the coefficient of variation over all estimates of parameter k using dataset d for calibration targets. To evaluate whether these averaged coefficients of variation statistically differs between algorithms we use a Wilcoxon signed rank test[f].

This leads us to the final research question (Section 1.4) regarding computational effort. The computational demand will be represented by the number of model runs necessary to terminate the algorithm, since the majority of computational effort goes into getting a function evaluation by running the model multiple times and both algorithms allow for parallel computation. The running time of MISCAN-colon does not have a linear relation with the population sample size, and it is therefore only fair to compare computational effort of algorithm settings that use the same population sample size.

Before comparing the two algorithms we first evaluate ABC-SMC individually by looking at the outcomes across the different settings of ABC-SMC. Besides evaluating the location parameter of the posterior distribution we also investigate the scale estimate of the posterior distribution between different settings and additionally investigate the acceptance rate of the final iteration for the main analysis (MA) and sensitivity analyses 8 and 9 (SA8, SA9).

To determine whether estimation accuracy, computation demand and acceptance rate statistically differ between algorithms and between algorithm settings we use the two-sided Wilcoxon rank sum test.

[f] Except for sensitivity analysis 1 we use the Wilcoxon rank sum test because this analysis contains more datasets than the main analysis

5. RESULTS

In this section relevant results of the calibration comparison study are presented following the structure of Section 4.3. Descriptions of the sensitivity analyses SA1-SA11 can be found in Table 3 and full details can be seen in Appendix 8.3.

5.1. Pre-processing

Figure 5 below depicts a scatterplot of the datasets used in the main analysis (MA) and the first sensitivity analysis (SA1). Here a MISCAN-colon population of 100,000 individuals is simulated. This scatter plot indicates no correlation between the two parameters, which is in line with our assumptions. The scatter plot also give us insight into the variability of the datasets and shows the location of the additional datasets used in SA1 with respect to MA.

Figure 6 below gives the scatterplots for population sizes 10,000 (SA2) and 1,000,000 (SA3). Summary statistics of all datasets can be found in Table 4 below. All scatterplots in Figure 5 and Figure 6 include one point indicating the expected values of the calibration targets. These are based on 20,000 model runs for each sample size. Exact values and additional summary statistics can be found in Table 5 below. The averages of the datasets used for calibration (Table 4) seem to approximately correspond to their expected values (Table 5) for most cases, however for the $D = 20$ datasets with largest sample size (1,000,000) the sample means are rather high with respect to the expected means. This can also be seen in the right scatterplot of Figure 6, as the majority of datasets are located to the top right of the expected value.

In Section 4.3 we mentioned that the runtime of MISCAN-colon is not linear with respect to population size, this can be seen in Table 5 which shows that the runtime per individual decreases as the MISCAN-colon population size grows.

5.2. Analysis

Terminal tolerances for the deviance in Nelder-Mead are computed from the ABC-SMC calibration results. For a MISCAN-colon sample size of 100,000 the terminal tolerance (Euclidean deviance) ranged between 7.0-8.3. For sample sizes 10,000 and 1,000,000 it ranged between 2.1-2.4 and 22.9-24.9 respectively. All terminal tolerances are displayed in Appendix 8.6, Table 12.

5.3. Post-processing

Following the structure from Section 4.3 we first discuss results of ABC-SMC individually, after which these results are compared to those of Nelder-Mead. Calibration results for all analyses of

both ABC-SMC as Nelder-Mead are evaluated based on their accuracy (Table 6), consistency and computational demand (Table 7).

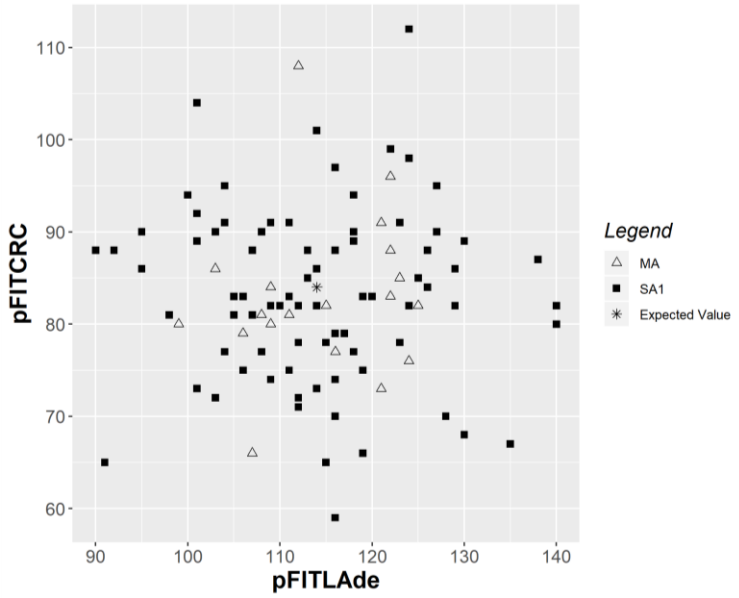

Figure 5. Scatterplot of calibration targets of sample size 100,000[g]

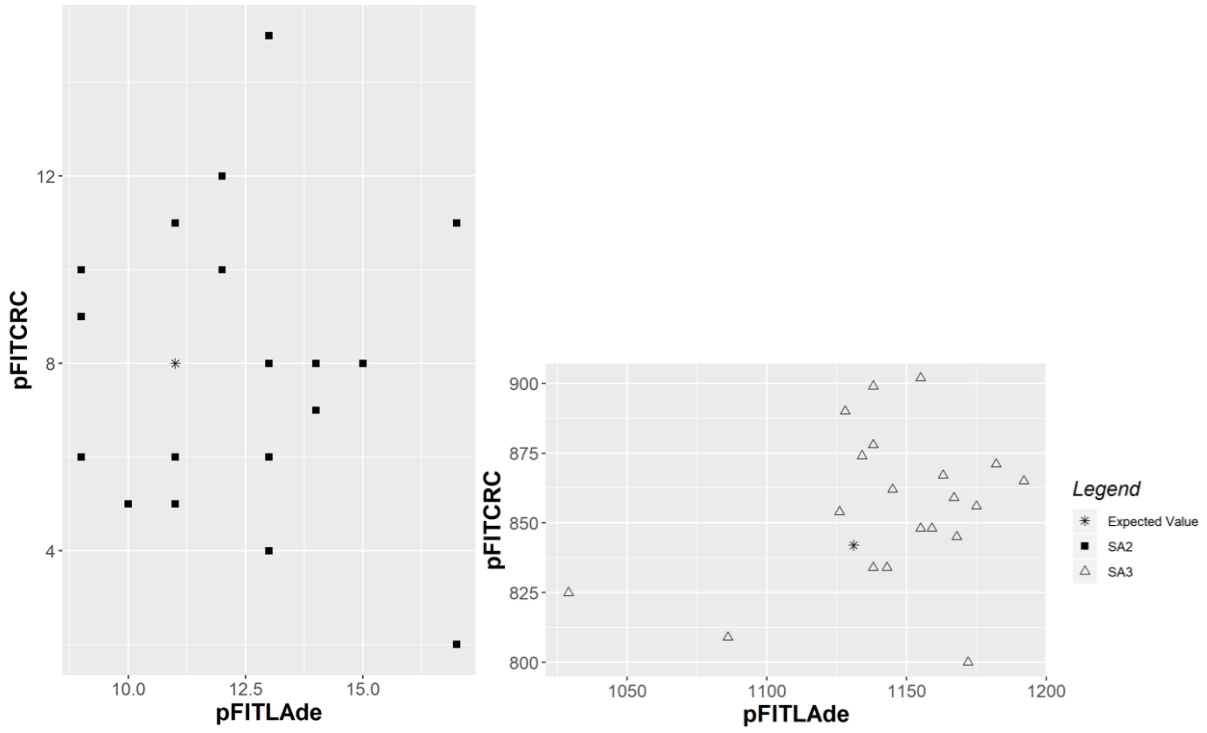

Figure 6. Scatterplots of calibration targets of sample sizes 10,000 (left) and 1,000,000 (right)

[g] All datapoints of the main analysis (MA) are also included in sensitivity analysis 1 (SA1)

Table 4. Summary statistics of calibration targets

Sample Size	$pFIT_{LAde}$			$pFIT_{CRC}$		
(Analysis{a})	Min	Mean(SD)	Max	Min	Mean(SD)	Max
10,000(SA2)	9	12(2.5)	17	2	7.9(3.1)	15
100,000(MA)	99	115(7.8)	125	66	83(8.7)	108
100,000(SA1)	90	114(10.4)	140	59	83(9.4)	112
1,000,000(SA3)	1029	1145(36.1)	1192	800	856(27.1)	902

{a} MA = Main Analysis, SA = Sensitivity Analysis, see Table 3 and Appendix 8.3 for details on the different analyses

Table 5. Summary statistics model analysis MISCAN-colon

Sample Size{a}	$pFIT_{LAde}$	$pFIT_{CRC}$	Runtime{b} (seconds)	
	Mean(SD)	Mean(SD)	Total{c}	Pp{d}
10,000	11(3.3)	8(2.9)	4.53	4.3E-04
100,000	114(10.7)	84(9.2)	6.54	5.1E-05
1,000,000	1131(33.6)	842(29.0)	28.88	2.8E-05

{a} MISCAN-colon population sample size
{b} The runtime can be divided into three components; (1) initializing and writing input files (2) running the MISCAN-colon executable and (3) reading and processing the output data. The second component dominates as it takes up from 96 to 99% of the total runtime depending on the population size of MISCAN-colon. Here we display the sum of all three components for a 64bit computer using only 1 core
{c} Runtime (as in {b}) of MISCAN-colon with true underlying parameters and sample size from first column, averaged over 20,000 independent runs
{d} Runtime (as in {c}) averaged per simulated individual

5.3.1 Evaluation ABC-SMC Calibrations

All except one of the ABC-SMC calibrations managed to reach their terminal tolerance before the maximum amount of model runs was exceeded while satisfying the particle degeneracy criterion. One calibration of sensitivity analysis 8 (SA8) did not manage to satisfy the degeneracy criterion after 6 re-samplings.

In the main analysis (MA) of ABC-SMC the estimates of $sens_{LAde}$ and $sens_{CRC}$ between 0.118-0.270 and 0.363-0.917 with averages of 0.180 and 0.675 respectively. The true underlying values are 0.159 and 0.700 respectively. We notice that on average the estimates of $sens_{CRC}$ were (relatively) closer to the true value than $sens_{LAde}$ with a relative absolute error about twice as large as $sens_{CRC}$ (Table 6). Estimates of $sens_{CRC}$ are more scattered on the other hand, with a larger relative range and slightly higher average coefficient of variation (CV) of the error (Table 7). In the final iteration the acceptance rate of proposed particles was 16% on average. The CVs of the final posterior estimate distribution were on average 0.269 and 0.256 for $sens_{LAde}$ and $sens_{CRC}$ respectively (See Appendix 8.6, Table 13).

When we compare results of MA on $D = 20$ datasets to the same analysis on $D = 100$ datasets (SA1) we see (Table 6) that errors do not significantly differ between MA and SA1. However when

investigating the errors averaged per dataset $\left(\frac{1}{C}\sum_{c=1}^{C} rMAE_{c,d}\right)$ we saw that the 11 largest relative mean absolute errors are from datasets not included in MA, these errors ranged between 0.49-0.26. Secondly the calibration consistency (Table 7) is not affected. Lastly Table 7 shows that the extra datasets do not significantly change the computational effort.

Sensitivity analyses 2 and 3 (SA2, SA3) used calibration targets of different sizes. In Table 6 we see that the overall estimate accuracy significantly increases with target sample size. The intuition behind this is that larger sample sizes ensure more certainty in the calibration targets. When looking at the average individual errors however we see that SA3, with the largest target sample size, has a higher average error for $sens_{CRC}$ than MA. Next we investigated the average values of the parameter point estimates (Table 6). Here we notice that the average point estimate of $sens_{CRC}$ is lower than the true value of 0.700 for all analyses except the one with the largest target sample size (SA3). This can be explained by the corresponding datasets which, as mentioned above (pre-processing), have higher means than their expected value. Lastly from Table 7 we can see that for both parameters the consistency of the estimates significantly improve with target sample size because the average CVs of the errors significantly decrease.

Sensitivity analyses 4 and 5 (SA4, SA5) incorporate prior information, where SA5 contains the most prior information. Table 6 shows that there is a significant increase in overall estimation accuracy and decrease in computational effort with increasing prior information. Looking at Table 6 and Table 7 we conclude that improvement in estimation accuracy when going from no prior information (MA) to a little prior information (SA4) seems to be based solemnly on better estimation of $sens_{CRC}$ as no significant improvement can be found for $sens_{LAde}$. From Table 7 we can read that the average CV of the errors, hence the consistency, improves for both parameter estimates when moving from no prior information (MA) to the most prior information (SA5). For less prior information (SA4) only the average CV of the error of parameter $sens_{CRC}$ significantly improves.

Sensitivity analyses 6 and 7 (SA6, SA7) vary the posterior size to 120 and 240 respectively with respect to MA which uses 40. We notice that the estimation accuracy does not significantly differ between the different posterior sizes (Table 6), however logically the computation effort significantly increases with higher posterior size (Table 7). This process seems to be linear. Secondly, from Table 7 we see that the average CVs of the errors significantly decrease when increasing the posterior size. Lastly the CV of the final posterior distribution of $sens_{LAde}$ significantly increased from 0.269 to 0.281 and 0.289 respectively when moving to a larger posterior sample size (Appendix 8.6, Table 13). No significant differences were found for the other parameter estimate.

Table 6. Accuracy of calibrations

Analysis[a]		$pFIT_{LAde}$		$pFIT_{CRC}$		Mean
		Estimate[b]	Error[c]	Estimate[b]	Error[c]	Error[d]
MA	ABC-SMC	0.180	0.20	0.675	0.11	0.15
	Nelder-Mead	0.179	**0.34**	0.217	**0.31**	**0.33**
SA1	ABC-SMC	0.181	0.24	0.678	0.12	0.18
SA2	ABC-SMC	0.358	**1.26**	0.585	**0.18**	**0.72**
	Nelder-Mead	0.421	**1.87**	0.279	**0.40**	**1.13**
SA3	ABC-SMC	0.167	**0.08**	0.777	**0.14**	**0.11**
	Nelder-Mead	0.170	**0.33**	0.224	**0.32**	**0.32**
SA4	ABC-SMC	0.181	0.19	0.694	**0.06**	**0.12**
	Nelder-Mead	0.670	**3.23**	0.516	**0.74**	**1.98**
SA5	ABC-SMC	0.166	**0.09**	0.699	**0.01**	**0.05**
	Nelder-Mead	0.666	**3.19**	0.516	**0.74**	**1.96**
SA6	ABC-SMC	0.183	0.20	0.183	0.10	0.15
SA7	ABC-SMC	0.183	0.20	0.679	0.10	0.15
SA8	ABC-SMC	0.183	0.20	0.692	0.11	0.15
SA9	ABC-SMC	0.185	0.21	0.689	0.10	0.16
SA10	ABC-SMC	0.181	0.21	0.676	0.10	0.15
SA11	ABC-SMC	0.179	0.20	0.686	0.10	0.15

{a} MA = Main Analysis, SA = Sensitivity Analysis, see Table 3 and Appendix 8.3 for details on the different analyses

{b} Average point estimate of parameter.

{c} Average relative absolute error of parameter k (rAE_k), see equation 5. Significant difference at 5% level indicated with boldface. Nelder-Mead errors are compared to their ABC-SMC counterpart (row above) and ABC-SMC errors are compared to ABC-SMC MA (first row).

{d} Average relative mean absolute error (rMAE), see equation 6. Significance indicated as in {c}

Table 7. Consistency and computational demand of calibrations

Analysis[a]		Average Coefficient of Variation [b]		Model Runs[c]
		$sens_{LAde}$	$sens_{CRC}$	
MA	ABC-SMC	0.05	0.06	954
	Nelder-Mead	**0.34**	**0.65**	**1367**
SA1	ABC-SMC	0.05	0.05	1009
SA2	ABC-SMC	**0.14**	**0.08**	769
	Nelder-Mead	**0.64**	**0.57**	**3530**
SA3	ABC-SMC	**0.02**	**0.03**	1388
	Nelder-Mead	**0.35**	**0.66**	**1818**
SA4	ABC-SMC	0.05	**0.04**	**720**
	Nelder-Mead	**0.31**	**0.11**	**1480**
SA5	ABC-SMC	**0.04**	**0.01**	**528**
	Nelder-Mead	**0.17**	**0.13**	**1198**
SA6	ABC-SMC	**0.03**	**0.03**	**2992**
SA7	ABC-SMC	**0.03**	**0.03**	**5743**
SA8	ABC-SMC	**0.07**	0.07	**1399**
SA9	ABC-SMC	0.04	**0.04**	**1052**
SA10	ABC-SMC	0.05	0.06	942
SA11	ABC-SMC	0.05	0.05	**1057**

{a} MA = Main Analysis, SA = Sensitivity Analysis , see Table 3 and Appendix 8.3 for details on the different analyses

{b} Coefficient of variation of the estimates averaged over the different datasets (7). Significant difference at 5% level indicated with boldface. Nelder-Mead results are compared to their ABC-SMC counterpart (row above) and ABC-SMC results are compared to ABC-SMC MA results (first row).

{c} Average number of times the MISCAN-colon executable was run with sample size corresponding to analysis. No significance indicated for ABC-SMC results of SA2 and SA3 due to their different MISCAN-population sample size

Sensitivity analyses 8 and 9 (SA8, SA9) vary the tolerance updating percentiles from 50 to 75 and 25 respectively, there are no significant differences between the estimation errors (Table 6). MA has the smallest computation demand (Table 7) with respect to SA8 and SA9, which confirms our choice of choosing an updating percentile of 50. However, when looking at the average CV of the errors (Table 7) we see a decrease in $sens_{CRC}$ for the lowest updating percentile (SA9) with respect to the main analysis (MA). The average acceptance rate of the final iteration significantly increases from 16% to 20% when moving from the 50^{th} to the 25^{th} percentile (SA8) and in line it decreases to 10% for SA9 which takes the largest steps in tolerance updating.

Sensitivity analyses 10 and 11 (SA10, SA11) vary the thresholds for detecting particle degeneracy from less to more stringent respectively. Again here we do not have significant differences between estimation errors (Table 6). The same holds for the average CVs of the errors. Only for the most strict degeneracy criterion (SA11) the computational effort is significantly larger than that of MA (Table 7).

5.3.2 Comparison of ABC-SMC and Nelder-Mead Calibrations

Looking at the errors given in Table 6, we see that for all comparable (MA,SA2-SA5) analyses the errors are smaller for ABC-SMC than Nelder-Mead. This applies for both the individual errors as the mean errors. This is translated into the Euclidean deviance of the final Nelder-Mead solutions, which are all significantly higher than that of ABC-SMC (see Appendix 8.6, Table 13). Besides this we notice that the individual estimation error of parameter $sens_{LAde}$ is often relatively much larger than that of $sens_{CRC}$ for both calibration algorithms. All ABC-SMC analyses managed to reach their terminal tolerances. Nelder-Mead on the other hand stagnated for almost all calibrations before reaching the terminal tolerance. Only for SA2 (calibration targets of sample size 10,000) the algorithm reached final tolerance 11 out of 100 calibrations and for SA5 (starting simplex from informative prior) once.

Table 7 shows the average CVs for each parameter estimate. Here we see that for all comparable analyses ABC-SMC is more consistent than Nelder-Mead, with lower values of the CVs for both parameters.

Lastly we investigate the difference in computational demand. For all comparable analyses Nelder-Mead needed significantly more model runs than ABC-SMC (Table 7), however when increasing the posterior sample size from 40 to 120/240 (SA6/SA7) the average number of model runs more than doubled/quadrupled with respect to Nelder-Mead.

In summary; with lower errors (Table 6), lower estimate variation and less model runs (Table 7) all comparable scenarios of ABC-SMC are significantly more accurate, consistent and efficient than Nelder-Mead.

6. DISCUSSION AND CONCLUSION

This study implemented an approximate Bayesian computation (ABC) method to estimate screen test sensitivity parameters of the stochastic colorectal cancer screening microsimulation model MISCAN-colon[h]. Estimation of these parameters is done using a calibration approach, meaning that the algorithm searches for parameter values which give function evaluations that match observed data (calibration targets) as closely as possible. We implemented a Bayesian calibration method based on sequential Monte Carlo (SMC). This ABC-SMC approach was implemented with multi-dimensional tolerances, meaning that each calibration target is evaluated on its own. Besides this we also included automatic updating of the tolerances starting at infinity and evolving towards their final levels in each iteration. For baseline comparison we implemented the Nelder-Mead calibration algorithm adjusted for stochastic models.

In the main analysis we compared the algorithms without prior information using calibration targets based on 100,000 simulated individuals. The robustness of ABC-SMC to different datasets was investigated by calibrating on additional simulated datasets. For both ABC-SMC as Nelder-Mead we performed two sensitivity analyses in which different levels of prior information were added. Besides this, both algorithms were tested with calibration targets of smaller and larger sample size. We also included six more sensitivity analyses for ABC-SMC in which specific algorithm settings were adjusted.

The results show that on average all ABC-SMC calibrations give more accurate and consistent parameter estimates compared to their Nelder-Mead counterparts. This is accompanied with lower computational intensity and function evaluations that more closely resemble their calibration targets. When looking at ABC-SMC individually we see a logical pattern in overall estimation accuracy for sensitivity analyses that increased and decreased calibration target sample size and also for those that added prior information. Besides this, ABC-SMC remained consistent when adding analyses on extra datasets. The other sensitivity analyses adjusting specific ABC-SMC algorithm settings showed no significant differences in parameter accuracy compared to the main analysis.

High accuracy of model parameters is vital for performance of a simulation model. Without correctly calibrated parameters model outcomes are unreliable. Proper calibration means that parameters are consistently accurate. This ensures that resulting model outcomes can be

[h] This model was developed in collaboration between Memorial Sloan-Kettering Cancer Center and Erasmus Medical Center

validated, which is crucial for policy influencing analyses such as the recommendation of national screening guidelines. With algorithms like Nelder-Mead and ABC-SMC estimation becomes more complicated as the dimensionality increases. It is therefore also important to take into consideration the computational demand in order to judge whether calibration of the whole model can be done in a reasonable time span.

Little research can be found comparing calibration algorithms that differ in nature as much as Nelder-Mead and approximate Bayesian computation, especially for stochastic micro simulation models like MISCAN-colon. The unique comparison of these different approaches on a practically relevant simulation model like MISCAN-colon is one of the strengths characterizing this research.

Limitations of this study include the low dimensionality and uncorrelated nature of the parameters. In order to judge the practicality of the calibration algorithms we need to take into consideration the effect of estimating more parameters on both the computational demand as the parameter accuracy. We also want to know how the different algorithms deal with (highly) correlated parameters in order to judge which algorithm is more suited at calibrating all model parameters. Secondly it is important to further investigate why the averages of the ABC-SMC estimates exceeded their true underlying values. Calibrations on different parameters could give more insight into this. Thirdly we should calibrate with larger posterior sizes. As we have stressed before, one of the attractions of using Bayesian approaches for calibration is the natural ability to perform probabilistic sensitivity analysis on the resulting parameter estimates. However in order to do this properly a larger posterior size is required to correctly capture the variation of the distribution, because in our analyses the average coefficient of variation did not stagnate for the posterior of parameter $sens_{LAde}$ using the posterior sizes 40,120 and 240. As can be logically reasoned and concluded from the sensitivity analyses results, larger posterior size will increase the computational demand. This increase seems to be linear based on our results, however larger posterior sizes should be used to investigate this hypothesis further. Lastly different kernels could be tried for ABC-SMC to possibly increase accuracy of the estimates.

Taken above mentioned limitations into account this research gives serious indication that parameters of stochastic disease simulation models like MISCAN-colon are better calibrated with a method like ABC-SMC than Nelder-Mead. Besides the better accuracy and consistency of the estimated parameters, ABC-SMC also directly allows for probabilistic sensitivity analysis over all calibrated parameters because it estimates a distribution instead of merely point estimates. Such probabilistic sensitivity analysis can strengthen the quality of a cost-effectiveness analysis by adding appropriate confidence intervals of the relevant results.

In conclusion our implementation of ABC-SMC performed better than the baseline Nelder-Mead. Important to note is that ABC-SMC has the benefit of directly performing probabilistic sensitivity analysis on all calibrated parameters and it can incorporate prior knowledge of the parameters in a statistically founded way. To answer the three research questions mentioned in Section 1.4; our results show that approximate Bayesian calibration can improve accuracy of MISCAN-colon parameters compared to Nelder-Mead. Besides this the estimates are also more consistent, and lastly the computational demand is lower for ABC-SMC with low posterior sizes, however when increasing posterior size this no longer holds.

7. BIBLIOGRAPHY

Barton, R. R., & Ivey, J. S. (1996). Nelder-Mead Simplex Modifications for Simulation Optimization. *Management Science, 42*(7), 939-1092.

Cenin, D. R., St John, D. J., Ledger, M. J., Slevin, T., & Lansdorp-Vogelaar, I. (2014). Optimising the expansion of the National Bowel Cancer Screening Program. *The Medical Journal of Australia, 201*(8), 456-461. doi:10.5694/mja13.00112 [pii]

Drovandi, C. C., & Pettitt, A. N. (2011). Estimation of Parameters for Macroparasite Population Evolution Using Approximate Bayesian Computation. *Biometrics, 67*(1), 225-233. doi:10.1111/j.1541-0420.2010.01410.x

Filippi, S., Barnes, C., & P.H., S. M. (2011). On optimal kenrels for ABC SMC. *arXiv preprint arXiv:1106.6280*.

Goede, S. L., Rabeneck, L., van Ballegooijen, M., Zauber, A. G., Paszat, L. F., Hoch, J. S., . . . Lansdorp-Vogelaar, I. (2017). Harms, benefits and costs of fecal immunochemical testing versus guaiac fecal occult blood testing for colorectal cancer screening. *Public Library of Science one, 12*(3), e0172864. doi:10.1371/journal.pone.0172864 PONE-D-16-27384 [pii]

Greenberg, E. (2012). *Introduction to Bayesian econometrics*: Cambridge University Press.

Humphrey, D. G., & Wilson, J. R. (2000). A revised simplex search procedure for stochastic simulation response surface optimization. *Informs journal on computing, 12*(4), 257-360. doi:https://doi.org/10.1287/ijoc.12.4.272.11879

Kong, A., Liu, J. S., & Wong, W. H. (1994). Sequential Imputations and Bayesian Missing Data Problems. *Journal of the American Statistical Association, 89*(425), 278-288. doi:Doi 10.2307/2291224

Kong, C. Y., McMahon, P. M., & Gazelle, G. S. (2009). Calibration of Disease Simulation Model Using an Engineering Approach. *Value in Health, 12*(4), 521-529. doi:10.1111/j.1524-4733.2008.00484.x

Leslie, A., Carey, F. A., Pratt, N. R., & Steele, R. J. (2002). The colorectal adenoma-carcinoma sequence. *British Journal of Surgery, 89*(7), 845-860. doi:2120 [pii]

10.1046/j.1365-2168.2002.02120.x

Liepe, J., Kirk, P., Filippi, S., Toni, T., Barnes, C. P., & Stumpf, M. P. (2014). A framework for parameter estimation and model selection from experimental data in systems biology using approximate Bayesian computation. *Nat Protoc, 9*(2), 439-456. doi:nprot.2014.025 [pii] 10.1038/nprot.2014.025

Loeve, F., Boer, R., van Oortmarssen, G. J., van Ballegooijen, M., & Habbema, J. D. F. (1999). The MISCAN-COLON simulation model for the evaluation of colorectal cancer screening. *Computers and Biomedical Research, 32*(1), 13-33. doi:DOI 10.1006/cbmr.1998.1498

Marjoram, P., Molitor, J., Plagnol, V., & Tavare, S. (2003). Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences of the United States of America, 100*(26), 15324-15328. doi:10.1073/pnas.0306899100

Meester, R. G. S., Peterse, E. F. P., Knudsen, A. B., de Weerdt, A. C., Chen, J. C., Lietz, A. P., . . . Lansdorp-Vogelaar, I. (2018). Optimizing colorectal cancer screening by race and sex: Microsimulation analysis II to inform the American Cancer Society colorectal cancer screening guideline. *Cancer, 124*(14), 2974-2985. doi:10.1002/cncr.31542

Moral, P. D., Doucet, A., & Jasra, A. (2012). An adaptive sequential Monte Carlo method for approximate Bayesian computation. *Statistcs and Computing, 22*(5), 1009-1020.

Neddermeijer, H., van Oortmarssen, G., Piersma, N., Dekker, R., & Habbema, D. (2000). *Adaptive extensions of the nelder and mead simplex method for optimization of stochastic simulation models* (No. EI 2000-22/A). Retrieved from hdl.handle.net/1765/1655

Neddermeijer, H. G., Piersma, N., van Oortmarssen, G. J., Habbema, J. D. F., & Dekker, R. (1999). *Comparison of response surface methodology and the Nelder and Mead simplex method for optimization in microsimulation models*. Retrieved from http://hdl.handle.net/1765/1595

Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. *The computer journal, 7*, 308-313.

Nocedal, J., & Wright, S. J. (2006). *Numerical Optimization* (2 ed.): Springer.

Pflug, G. C. (2012). *Optimization of stochastic models: The interface between simulation and optimization* (Vol. 373): Springer Science & Business Media.

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

Rutter, C., Ozik, J., DeYoreo, M., & Collier, N. (2018). Microsimulation model calibration using incremental mixture approximate Bayesian computation. *arXiv preprint arXiv:1804.02090*.

Rutter, C. M., Knudsen, A. B., Marsh, T. L., Doria-Rose, V. P., Johnson, E., Pabiniak, C., . . . Lansdorp-Vogelaar, I. (2016). Validation of Models Used to Inform Colorectal Cancer Screening Guidelines: Accuracy and Implications. *Medical Decision Making, 36*(5), 604-614. doi:10.1177/0272989x15622642

Rutter, C. M., Miglioretti, D. L., & Savarino, J. E. (2009). Bayesian Calibration of Microsimulation Models. *Journal of the American Statistical Association, 104*(488), 1338-1350. doi:10.1198/jasa.2009.ap07466

Seigneurin, A., Francois, O., Labarere, J., Oudeville, P., Monlong, J., & Colonna, M. (2011). Overdiagnosis from non-progressive cancer detected by screening mammography: stochastic simulation study with calibration to population based registry data. *British Medical Journal, 343*, d7017. doi:10.1136/bmj.d7017

Sisson, S. A., Fan, Y., & Tanaka, M. M. (2007). Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences of the United States of America, 104*(6), 1760-1765. doi:10.1073/pnas.0607208104

Sisson, S. A., Fan, Y., & Tanaka, M. M. (2008). *A note on backward kernel choice for sequential Monte Carlo without likelihoods*. Retrieved from

Stout, N. K., Knudsen, A. B., Kong, C. Y., McMahon, P. M., & Gazelle, G. S. (2009). Calibration methods used in cancer simulation models and suggested reporting guidelines. *Pharmacoeconomics, 27*(7), 533-545. doi:2 [pii] 10.2165/11314830-000000000-00000

Tavare, S., Balding, D. J., Griffiths, R. C., & Donnelly, P. (1997). Inferring coalescence times from DNA sequence data. *Genetics, 145*(2), 505-518.

Tomick, J. J., Arnold, S. F., & Barton, R. R. (1995). *Sample size selection for improved Nelder-Mead performance*. Paper presented at the Winter Simulation Conference Proceedings, 1995., Arlington, VA, USA, USA.

Toni, T., & Stumpf, P. H. (2009). Tutorial on ABC rejection and ABC SMC for parameter estimation and model selection. *arXiv preprint arXiv:0910.4472*.

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., & Stumpf, M. P. H. (2008). Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface, 6*(31), 187-202. doi:10.1098/rsif.2008.0172

van Hees, F., Zauber, A. G., van Veldhuizen, H., Heijnen, M. L., Penning, C., de Koning, H. J., . . . Lansdorp-Vogelaar, I. (2015). The value of models in informing resource allocation in colorectal cancer screening: the case of The Netherlands. *Gut, 64*(12), 1985-1997. doi:gutjnl-2015-309316 [pii] 10.1136/gutjnl-2015-309316

Whyte, S., Walsh, C., & Chilcott, J. (2011). Bayesian calibration of a natural history model with application to a population model for colorectal cancer. *Medical Decision Making, 31*(4), 625-641. doi:0272989X10384738 [pii]

10.1177/0272989X10384738

Zauber, A. G., Lansdorp-Vogelaar, I., Knudsen, A. B., Wilschut, J., van Ballegooijen, M., & Kuntz, K. M. (2008). Evaluating test strategies for colorectal cancer screening: a decision analysis for the U.S. Preventive Services Task Force. *Annals of Internal Medicine, 149*(9), 659-669. doi:0000605-200811040-00244 [pii]

8. APPENDIX

This appendix contains pseudocode of the calibration algorithms (8.2), settings of the calibration algorithms for the main analysis and sensitivity analyses (8.3), relevant model settings (8.4) a derivation (8.5) and some additional results (8.6).

8.1. ABBREVIATIONS

ABC	=	Approximate Bayesian Computation
CRC	=	ColoRectal Cancer
CV	=	Coefficient of Variation
ESS	=	Effective Sample Size
FIT	=	Faecal Immonuchemical Test for Haemoglobin
MA	=	Main Analysis
MISCAN	=	MIcro simulation Screening ANalysis
MISCAN-colon	=	MISCAN-colorectal cancer
$pFIT_{LAde}$	=	number of large adenomas detected during diagnostic colonoscopy after a positive FIT result
$pFIT_{CRC}$	=	number of colorectal cancers detected during diagnostic colonoscopy after a positive FIT result
SA	=	Sensitivity Analysis
SD	=	Standard Deviation
$sens_{CRC}$	=	FIT screen test sensitivity for detecting colorectal cancer
$sens_{LAde}$	=	FIT screen test sensitivity for detecting large adenomas
SMC	=	Sequential Monte Carlo

8.2. PSEUDOCODE

Algorithm 1. ABC-SMC with adaptive multi-dimensional tolerance updating

Input:

- *population size (N)*
- *tolerance updating percentile* (τ)
- *Joint prior distribution* $= \pi(\bar{\theta})$
- *Joint kernel distribution* $= K(\bar{\theta};\ \widehat{\sum}(\{\bar{\theta}_i\}^{\langle i \rangle}))$
- *Threshold* $CV^2 = CVthreshold$
- *distance measure(s) and calibration target(s)* $= distance_j(\cdot\,;\bar{\mu})\ \forall\ j = 1 \dots J$
- *termination constants* $= (stagnationMax, \bar{\varepsilon}^{\langle T \rangle}, maxModelRuns)$

Output:

- *population of particles*

algorithm

- $t \leftarrow 0$
- $terminate \leftarrow false$
- $toleranceReached \leftarrow false$
- $draw\ \bar{\theta}_i^{\langle t \rangle} \sim \pi\ , \forall\ i = 1 \dots N$
- $w_i^{\langle t \rangle} \leftarrow \frac{1}{N}, \forall\ i = 1 \dots N$
- $compute\ distance_j\left(\theta_{i,j}^{\langle t \rangle}\,;\bar{\mu}\right) \forall\ i = 1 \dots N,\ j = 1 \dots J$
- $modelRuns \leftarrow N$
- $\varepsilon_j^{\langle t \rangle} \leftarrow \tau^{th}\ percentile\ of\ \left\{distance_j\left(\theta_{i,j}^{\langle t \rangle}\,;\bar{\mu}\right)\right\}^{\langle i=1\dots N \rangle}, \forall\ j = 1 \dots J$

while $terminate = false$ **do**

- $N_t^{\langle 0 \rangle} \leftarrow \infty$
- $i \leftarrow 1$
- $t_{sub} \leftarrow 1$
- $N_t^{\langle t_{sub} \rangle} \leftarrow N$

while $N_t^{\langle t_{sub} \rangle} > 0$ **AND** $terminate = false$ **do**

- $stagnationCounter \leftarrow 0$
- $draw\ \bar{\theta}_p{}^* \sim \left\{\bar{\theta}_p^{\langle t-1 \rangle}, w_q^{\langle t-1 \rangle}\right\}^{\langle q=1:N \rangle}, \forall\ p = 1 \dots N_t^{\langle t_{sub} \rangle}$
- $perturb\ \bar{\theta}_p{}^* \sim K\left(\bar{\theta}_p{}^*;\ \widehat{\sum}\left(\left\{\bar{\theta}_q^{\langle t-1 \rangle}\right\}^{\langle q=1:N \rangle}\right)\right)\ , \forall\ p = 1 \dots N_t^{\langle t_{sub} \rangle}$
- $compute\ distance_j\left(\theta_{p,j}^*\,;\bar{\mu}\right) \forall\ p = 1 \dots N_t^{\langle t_{sub} \rangle},\ j = 1 \dots J$
- $modelRuns = modelRuns + N_t^{\langle t_{sub} \rangle}$

21 **for** $p = 1 \dots N_t^{\langle t_{sub} \rangle}$ **do**

22 **if** $\text{distance}_j\left(\theta^*_{p,j}\,;\bar{\mu}\right) \leq \varepsilon_j^{\langle t \rangle}, \forall\, j = 1 \dots J$ **do**

23 ➢ $\bar{\theta}_i^{\langle t \rangle} \leftarrow \bar{\theta}_p^{\,*}$

24 ➢ $i \leftarrow i + 1$

25 ➢ *compute* $w_i^{\langle t \rangle}$ *with formula* ***(3)***

26 ➢ $N_t^{\langle t_{sub} \rangle} \leftarrow N_t^{\langle t_{sub} \rangle} - 1$

27 **if** $N_t^{\langle t_{sub}-1 \rangle} - N_t^{\langle t_{sub} \rangle} = 0$ **do**

28 ➢ $stagnationCounter \leftarrow stagnationCounter + 1$

29 **else do**

30 ➢ $stagnationCounter \leftarrow 0$

31 **if** $modelRuns \geq maxModelRuns$ **OR** $stagnationCounter \geq stagnationMax$ **do**

32 ➢ $terminate \leftarrow true$

33 ➢ $t \leftarrow t - 1$

34 ➢ $N_t^{\langle t_{sub}+1 \rangle} \leftarrow N_t^{\langle t_{sub} \rangle}$

35 ➢ $t_{sub} \leftarrow t_{sub} + 1$

36 **if** $terminate = false$ **do**

37 ➢ *compute* CV^2 *of weights*

38 **if** $CV^2 > CVthreshold$ **do**

39 ➢ *replace* $\bar{\theta}_i^{\langle t \rangle}$ *by a draw from* $\left\{\bar{\theta}_q^{\langle t \rangle}, w_q^{\langle t \rangle}\right\}^{\langle q = 1:N \rangle}, \forall\, i = 1 \dots N$

40 ➢ $w_i^{\langle t \rangle} \leftarrow \frac{1}{N}, \forall\, i = 1 \dots N$

41 **else**

42 **if** $toleranceReached = true$ **do**

43 ➢ $terminate \leftarrow true$

44 **else**

45 ➢ $\varepsilon_j^{\langle t \rangle} \leftarrow min\left(\varepsilon_j^{\langle T \rangle}, \tau^{th}\ percentile\ of\ \left\{distance_j\left(\theta_{i,j}^{\langle t \rangle}\,;\bar{\mu}\right)\right\}^{\langle i=1\dots N \rangle}\right), \forall\, j = 1 \dots J$

46 **if** $\varepsilon_j^{\langle t \rangle} \leq \varepsilon_j^{\langle T \rangle}, \forall\, j = 1 \dots J$ do

47 ➢ $toleranceReached \leftarrow true$

48 ➢ $t \leftarrow t + 1$

49 ➢ ***return*** $\left\{\bar{\theta}_i^{\langle t-1 \rangle}, w_i^{\langle t-1 \rangle}\right\}^{\langle i = 1:N \rangle}$

Algorithm 2. Nelder-Mead simplex algorithm with DN-IR extension

Input:

- *simplex;* $\{\bar{\theta}_i\}^{\langle i=1:K+1 \rangle}$
- *reflect,contract, expand, shrink constants* ($\alpha, \beta, \gamma, \delta$)
- *dominant noise extension constants* (R_0, $R_{increase}$, R_{max} , λ)
- *deviance function and calibration target(s);* $deviance(\cdot\ ; \bar{\mu})$
- *termination constants (improveThreshold, stagnationMax,* ε*, maxModelRuns)*

Output:

- *particle*

algorithm

- $t \leftarrow 0$
- $terminate \leftarrow false$
- $noImprovement \leftarrow 0$
- *compute* $deviance(\bar{\theta}_i; \bar{\mu}), \forall\ i \ldots K+1$
- $modelRuns \leftarrow (K+1) * R_t$
- *compute dominant noise criterion;* DN

if DN is not rejected **do**

- $R_{t+1} \leftarrow min(R_t * R_{increase}, R_{max})$
- *Add extra model runs and compute deviance again*
- $modelRuns \leftarrow modelRuns + (R_{t+1} - R_t) * (K+1)$
- *order simplex s.t.* $deviance(\bar{\theta}_1; \bar{\mu}) \leq \cdots \leq deviance(\bar{\theta}_K; \bar{\mu}) \leq deviance(\bar{\theta}_{K+1}; \bar{\mu})$

while $terminate = false$ **do**

- $\bar{\theta}_{centroid} \leftarrow \frac{1}{K}\sum_{i=1}^{K} \bar{\theta}_i$
- $\bar{\theta}_t^{\langle original \rangle} \leftarrow (1-\alpha) * \bar{\theta}_{centroid} + \alpha * \bar{\theta}_{K+1}$
- *compute* $deviance\left(\bar{\theta}_t^{\langle original \rangle}; \bar{\mu}\right)$
- $modelRuns \leftarrow modelRuns + R_t$

if $deviance(\bar{\theta}_{reflect}; \bar{\mu}) < deviance(\bar{\theta}_1; \bar{\mu})$ **do**

- $\bar{\theta}_{expand} \leftarrow (1-\gamma) * \bar{\theta}_{centroid} + \gamma * \bar{\theta}_t^{\langle original \rangle}$
- *compute* $deviance(\bar{\theta}_{expand}; \bar{\mu})$
- $modelRuns \leftarrow modelRuns + R_t$

if $deviance(\bar{\theta}_{expand}; \bar{\mu}) < deviance\left(\bar{\theta}_t^{\langle original \rangle}\right); \bar{\mu}$ **do**

- $\bar{\theta}_{K+1} \leftarrow \bar{\theta}_{expand}$

else do

- $\bar{\theta}_{K+1} \leftarrow \bar{\theta}_t^{\langle original \rangle}$

if $deviance(\bar{\theta}_1; \bar{\mu}) < deviance\left(\bar{\theta}_t^{\langle original \rangle}; \bar{\mu}\right) < deviance(\bar{\theta}_K; \bar{\mu})$ **do**

- $\bar{\theta}_{K+1} \leftarrow \bar{\theta}_t^{\langle original \rangle}$

if $deviance(\bar{\theta}_K; \bar{\mu}) < deviance\left(\bar{\theta}_t^{\langle original \rangle}; \bar{\mu}\right) < deviance(\bar{\theta}_{K+1}; \bar{\mu})$ **do**

- $\bar{\theta}_{contract} \leftarrow (1-\beta) * \bar{\theta}_{centroid} + \beta * \bar{\theta}_t^{\langle original \rangle}$
- *compute* $deviance(\bar{\theta}_{contract}; \bar{\mu})$
- $modelRuns \leftarrow modelRuns + R_t$

if $deviance(\bar{\theta}_{contract}; \bar{\mu}) < deviance\left(\bar{\theta}_t^{\langle original \rangle}; \bar{\mu}\right)$ **do**

- $\bar{\theta}_{K+1} \leftarrow \bar{\theta}_{contract}$

else do

- $\bar{\theta}_{K+1} \leftarrow \bar{\theta}_t^{\langle original \rangle}$
- $\bar{\theta}_i \leftarrow (1-\delta) * \bar{\theta}_1 + \delta * \bar{\theta}_i \,, \forall\, i \ldots K+1$
- *compute* $deviance(\bar{\theta}_i; \bar{\mu}), \forall\, i \ldots K+1$
- $modelRuns \leftarrow modelRuns + R_t * (K+1)$

if $\boldsymbol{deviance(\bar{\theta}_{K+1}; \bar{\mu}) < deviance\left(\bar{\theta}_t^{\langle original \rangle}; \bar{\mu}\right)}$ **do**

- $\bar{\theta}_{contract} \leftarrow (1-\beta) * \bar{\theta}_{centroid} + \beta * \bar{\theta}_{K+1}$

if $\boldsymbol{deviance(\bar{\theta}_{contract}; \bar{\mu}) < deviance(\bar{\theta}_{K+1}; \bar{\mu})}$ **do**

- $\bar{\theta}_{K+1} \leftarrow \bar{\theta}_{contract}$

else do

- $\bar{\theta}_i \leftarrow (1-\delta) * \bar{\theta}_1 + \delta * \bar{\theta}_i \,, \forall\, i \ldots K+1$
- *compute* $deviance(\bar{\theta}_i; \bar{\mu}), \forall\, i \ldots K+1$
- $modelRuns \leftarrow modelRuns + R_t * (K+1)$

- *compute dominant noise criterion; DN*

if $\boldsymbol{DN}$ is not rejected **do**

- $R_{t+1} \leftarrow min(R_t * R_{increase}, R_{max})$
- *Add extra model runs and compute deviances again*
- $modelRuns \leftarrow modelRuns + (R_{t+1} - R_t) * (K+1)$

- *order simplex s.t.* $deviance(\bar{\theta}_1; \bar{\mu}) \leq \cdots \leq deviance(\bar{\theta}_K; \bar{\mu}) \leq deviance(\bar{\theta}_{K+1}; \bar{\mu})$
- $deviance_t^{best} \leftarrow deviance(\bar{\theta}_1; \bar{\mu})$

```
57      if |deviance_t^best − deviance_{t−1}^best| < improveThreshold do
58          ➢ noImprovement ← noImprovement + 1
59      ➢ order simplex s.t. deviance(θ̄_1; μ̄) ≤ ⋯ ≤ deviance(θ̄_K; μ̄) ≤ deviance(θ̄_{K+1}; μ̄)
60      if deviance_t^best ≤ ε OR noImprovement > stagnationMax OR modelRuns ≥ maxModelRuns do
61          ➢ termination ← true
62      ➢ t ← t + 1
        return θ̄_1
63
```

8.3. CALIBRATION SETTINGS

Table 8. Calibration Settings Main Analysis ABC-SMC

Description		Setting
Calibrations		
Number of Calibrations ($C * D$)	=	$5 * 20$
Sample Size		
Sample size observed calibration targets (ss_{obs})	=	100,000
Sample size MISCAN (ss_{sim})	=	100,000
Prior knowledge		
Prior colorectal cancer sensitivity FIT ($\pi_{sensCRC}$)	=	Standard Uniform
Prior Large adenoma sensitivity FIT ($\pi_{sensLAde}$)	=	Standard Uniform
Similarity Measure		
Distance measure colorectal cancer sensitivity FIT	=	Absolute
Distance measure Large adenoma sensitivity FIT	=	Absolute
Termination criteria		
Terminal tolerance colorectal cancer sensitivity FIT ($\varepsilon_{sensCRC}$)	=	9.2
Terminal tolerance large adenoma sensitivity FIT ($\varepsilon_{sensLAde}$)	=	10.7
Maximum number of MISCAN model runs (maxModelRuns)	=	105,619 ($\pm$1day)
Maximum number of consecutive subiterations with zero accepted particles	=	5
Maximum number of additional iterations to satisfy CV^2	=	6
Algorithm specific settings		
Minimum ESS (Maximum CV^2)	=	50%(1.0)
Tolerance updating percentile (τ)	=	50
Population size (N)	=	40
Perturbation Kernel (K)	=	TruncNorm{a}

{a} Truncated Normal with covariance equal to sample covariance of previous posterior population

Table 9. Calibration Settings Sensitivity Analyses ABC-SMC

Analyis	Change with respect to main analysis
SA1	$D = 100$
SA2	$ss_{obs} = 1e+04$; $ss_{sim} = 1e+04$; $\varepsilon_{sensCRC} = 3.3$; $\varepsilon_{sensLAde} = 2.9$; maxModelRuns = 152,683 ($\pm$1day)
SA3	$ss_{obs} = 1e+06$; $ss_{sim} = 1e+06$; $\varepsilon_{sensCRC} = 33.4$; $\varepsilon_{sensLAde} = 29.0$; maxModelRuns = 23,935 ($\pm$1day)
SA4	$\pi_{sensCRC} = truncN(mean = 0.7, variance = 0.04)$, $\pi_{sensLAde} = truncN(mean = 0.159, variance = 0.04)$,
SA5	$\pi_{sensCRC} = truncN(mean = 0.7, variance = 0.0025)$, $\pi_{sensLAde} = truncN(mean = 0.159, variance = 0.0025)$,
SA6	$N = 120$
SA7	$N = 240$
SA8	$\tau = 0.75$
SA9	$\tau = 0.25$
SA10	ESS minimum = 25% (Maximum $CV^2 = 3$)
SA11	ESS minimum = 75% (Maximum $CV^2 = 1/3$)

{a} All other settings are the same as in the main analysis (Table 8)

All sensitivity analyses of ABC-SMC except 11 are performed $C = 5$ on $D = 20$ datasets. Sensitivity analysis 11 combines the 100 calibrations already performed for the main analysis with 400 additional calibrations. In total this results in 1500 ABC-SMC calibrations.

Table 10. Calibration Settings Main Analysis Nelder-Mead

Description		Setting
Calibrations		
Number of Calibrations ($C * D$)	=	$5 * 20$
Sample Size		
Sample size observed calibration targets (ss_{obs})	=	100,000
Sample size MISCAN (ss_{sim})	=	100,000
Initial number of replications for function evaluation (R_0)	=	5
Maximum number of replications for function evaluation (R_{max})	=	100
Prior knowledge		
Starting simplex colorectal cancer sensitivity FIT	=	random
Starting simplex large adenoma sensitivity FIT	=	random
Similarity Measure		
Deviance function	=	Euclidean
Termination criteria		
Minimum improvement deviance (improveThreshold)	=	0.5
Maximum number of MISCAN model runs (maxModelRuns)	=	105,619 ($\pm$1day)
Maximum number of consecutive iterations with less than improveThreshold in deviance (stagnationMax)	=	5
Terminal tolerance deviance	=	Euclidean deviance{a}
Algorithm specific settings		
Significance level for Dominant Noise criterion (λ)	=	0.1
Dominant noise replications increase constant ($R_{increase}$)	=	1.25
Dominant noise significance level (λ)	=	0.1
Reflection constant (α)	=	-1
Contraction constant (β)	=	0.5
Expansion constant (γ)	=	2
Shrinking constant (δ)	=	0.9

{a} Euclidean deviance over average distances resulting from ABC-SMC calibration (separately computed for each dataset)

Table 11. Calibration Settings Sensitivity Analyses Nelder-Mead

Analysis	Change with respect to main analysis
SA2	$ss_{obs} = 1e+04$; $ss_{sim} = 1e+04$; terminal tolerance from ABC-SMC corresponding to SA2; maxModelRuns = 152,683 ($\pm$1day)
SA3	$ss_{obs} = 1e+06$; $ss_{sim} = 1e+06$; terminal tolerance from ABC-SMC corresponding to SA3; maxModelRuns = 23,935 ($\pm$1day)
SA4	Starting simplex colorectal cancer and large adenoma sensitivity FIT randomly chosen from initial population of ABC-SMC SA4
SA5	Starting simplex colorectal cancer and large adenoma sensitivity FIT randomly chosen from initial population of ABC-SMC SA4

{a} All other settings are the same as in the main analysis (Table 10)

Together with the main analysis and these sensitivity analyses we perform in total 500 Nelder-Mead calibrations.

8.4. MODEL SETTINGS – MISCAN-colon

For this calibration study we simulated a cohort of individuals born in 1966 who receive yearly FIT screening from the age of 50 until 75. We modeled 100% adherence to screening, diagnostic testing and surveillance. The diagnostic colonoscopy test was modelled to have a fatal complication rate for all adenoma disease states of 0.00191% and a test sensitivity of 75, 85 and 95% for small adenoma-, medium adenoma- and all colorectal cancer- and large adenoma disease states together respectively. For FIT we modelled a lack of specificity in in each disease state of 3.6%. FIT sensitivity was assumed to be 11.4, 15.9 and 70% for medium adenoma-, large adenoma- and all colorectal cancer disease states respectively and zero for the small adenoma disease state.

8.5. DERIVATION – Effective Samples Size

$$
\begin{aligned}
CV^2(w) = & \left(\frac{sd(w)}{mean(w)}\right)^2 \\
= & \frac{E(w^2) - E(w)^2}{E(w)^2} \\
\approx & \frac{\frac{1}{n}\sum_{i=1}^{N}\left(w_i^t\right)^2 - \left(\frac{1}{n}\sum_{i=1}^{N} w_i^t\right)^2}{\left(\frac{1}{n}\sum_{i=1}^{N} w_i^t\right)^2} \\
= & \frac{\frac{1}{n}\sum_{i=1}^{N}\left(w_i^t\right)^2 - \frac{1}{n} * \frac{1}{n}\left(\sum_{i=1}^{N} w_i^t\right)^2}{\frac{1}{n} * \frac{1}{n}\left(\sum_{i=1}^{N} w_i^t\right)^2} \\
= & \frac{N\sum_{i=1}^{N}\left(w_i^t\right)^2}{\left(\sum_{i=1}^{N} w_i^t\right)^2} - 1
\end{aligned}
$$

Here $w = \left\{W_i^t\right\}_i^N$ and $E(\cdot)$ gives the expected value

A value for CV^2 corresponding to an ESS equal to x −percentage of the population (N) is obtained using the relation $CV^2 = \frac{N}{ESS} - 1$, and substituting $ESS = \frac{x}{100} * N$ resulting in $CV^2 = \frac{100-x}{x}$. Hence for $x = (25{,}50{,}75)$, $CV^2 = \left(\frac{75}{25}, \frac{50}{50}, \frac{25}{75}\right) = (3, 1, \frac{1}{3})$.

8.6. ADDITIONAL RESULTS

Table 12. Summary Overview of Termination Criteria of Nelder-Mead ($D = 20$)

Dataset	Analysis[1]				
	MA	SA2	SA3	SA4	SA5
1	7.67	2.33	23.60	7.22	7.20
2	7.43	2.12	23.66	6.91	7.22
3	7.53	2.24	22.89	7.16	7.39
4	7.30	2.21	23.90	7.70	7.20
5	7.24	2.18	23.02	7.63	7.11
6	7.43	2.40	24.77	7.56	7.26
7	7.56	2.22	25.26	7.33	7.19
8	7.28	2.21	23.64	7.22	7.30
9	7.51	2.27	24.65	7.57	7.07
10	7.40	2.13	23.90	7.93	8.54
11	7.68	2.30	24.55	7.53	7.26
12	7.53	2.40	23.95	7.48	7.17
13	7.33	2.24	24.52	7.36	7.57
14	7.42	2.33	24.58	7.32	7.96
15	7.42	2.38	24.50	7.56	7.26
16	7.38	2.14	24.48	7.37	7.67
17	7.21	2.34	24.33	6.94	7.17
18	7.93	2.25	23.71	8.36	8.21
19	7.42	2.22	23.93	7.67	7.25
20	7.48	2.12	23.82	7.61	7.33

{a} MA = Main Analysis, SA = Sensitivity Analysis , see Table 3 and Appendix 8.3 for details on the different analyses

Table 13. Variation and Average Deviance of Calibrations

Analysis[a]		Euclidean Deviance[b]	CV $sens_{LAde}$[c]	CV $sens_{CRC}$[c]
MA	ABC-SMC	7.48	0.27	0.26
	Nelder-Mead	**22.01**		
SA1	ABC-SMC	**7.58**	0.27	0.26
SA2	ABC-SMC	**2.27**	**0.60**	**0.44**
	Nelder-Mead	**5.31**		
SA3	ABC-SMC	**23.91**	**0.09**	**0.12**
	Nelder-Mead	**185.62**		
SA4	ABC-SMC	7.51	0.26	**0.21**
	Nelder-Mead	**21.74**		
SA5	ABC-SMC	7.40	**0.19**	**0.07**
	Nelder-Mead	**20.16**		
SA6	ABC-SMC	7.58	**0.28**	0.27
SA7	ABC-SMC	**7.62**	**0.29**	0.27
SA8	ABC-SMC	7.57	0.26	**0.24**
SA9	ABC-SMC	**7.64**	**0.29**	0.25
SA10	ABC-SMC	7.53	0.27	0.26
SA11	ABC-SMC	7.46	0.26	0.25

{a} MA = Main Analysis, SA = Sensitivity Analysis , see Table 3 and Appendix 8.3 for details on the different analyses

{b} Average Euclidean deviance corresponding to estimates. Significant difference at 5% level indicated with boldface. Nelder-Mead results are compared to their ABC-SMC counterpart (row above) and ABC-SMC results are compared to ABC-SMC MA results (first row).

{c} Average coefficient of variation of final posterior, significant differences denoted as in {b}