Improving the Generation and Selection of Virtual Populations in Quantitative Systems Pharmacology Models

Theodore R. Rieger[1], Richard J. Allen[1], Lukas Bystricky[2], Yuzhou Chen[3], Glen Colopy[4], Yifan Cui[5], Angelica Gonzalez[6], Yifei Liu[7], R. D. White[8], R. A. Everett[8], H. T. Banks[8], Cynthia J. Musante[1]

[1]Internal Medicine Research Unit, Pfizer Inc, Cambridge, MA, USA
[2]Department of Computer Science, Florida State University, Tallahassee, FL, USA
[3]Department of Statistics, University of Texas-Dallas, Dallas, TX, USA
[4]Institute of Biomedical Engineering, Oxford University, Oxford, UK
[5]Department of Statistics, University of North Carolina-Chapel Hill, Chapel Hill, NC, USA
[6]Department of Mathematics, University of Arizona, Tucson, AZ, USA
[7]Department of Statistics, University of Wisconsin-Madison, Madison, WI, USA
[8]Center for Research in Scientific Computation, North Carolina State University, Raleigh, NC, USA

*** Correspondence:**
Theodore Rieger
ted.rieger@pfizer.com

Abstract

Quantitative systems pharmacology (QSP) models aim to mechanistically describe disease progression and predict the effects of existing therapies or novel compounds on disease biomarkers or outcomes. For most drug development applications, it is important to predict not only the mean response to the intervention but also the inter-patient variability. In addition, given the complexity of QSP models and the sparsity of relevant human data, the parameters of QSP models are often not well-determined. One approach to overcome these limitations is to develop Virtual Populations (VPops), which allow for the exploration of parametric uncertainty and reproduce the variability in response to perturbation. Here we evaluate approaches to improve the efficiency of generating VPops. We aimed to generate these populations without sacrificing diversity of the Virtual Patients' (VPs) pathophysiologies and phenotypes. To do this, we used our previously published approach (Allen, Rieger et al. 2016) together with two established algorithms (genetic algorithm and Metropolis-Hastings) as well as developing a novel approach, which we call "nested simulated annealing." Each method improved our previously-published algorithm in at least one aspect. For example, all of the methods tested required significantly fewer plausible patients (PP), the precursors to VPs, to create a reasonable VPop, but we found there may be tradeoffs in terms of parametric and phenotypic diversity of the VPs. However, the improved methodologies introduced here may be appropriate in many applications.

1 Introduction

Physiologically-based mathematical models are often used to describe and predict the response of a patient to an existing therapy or novel agent. These models, frequently referred to as quantitative systems pharmacology (QSP) models, are used to simulate clinical trials in drug development (Musante, Ramanujan et al. 2017). In these applications, it is important that they not only capture the mean patient response to treatment but also interpatient variability and how that variability may evolve over time. In addition, due to the novel nature of many therapies; the complexity of human physiology; and generally limited human data, QSP models are rarely fully determined by data. One approach to these challenges is to develop alternate parameter sets to capture the variability in the real clinical trial population and sample as much uncertainty in the model parameters as possible (Gadkar, Budha et al. 2014, Hallow, Lo et al. 2014, van de Pas, Rullmann et al. 2014).

Previously, we published an algorithm for the generation and selection of these alternative value sets (Allen, Rieger et al. 2016). The flow of the algorithm was to use simulated annealing to generate as large a population of "plausible virtual patients" as was practical. Simulated annealing was used with a given cost functional that optimized solutions to be biologically feasible. These plausible patients, and as a collection plausible population, were termed plausible since each generated parameter set simulated a patient that was physiologically reasonable, and could be in a clinical trial, but there was not yet any selection for how *likely* it was for that patient to have been in a particular clinical trial. We then used our novel selection technique to choose those patients from the plausible population that most resembled a desired clinical population. These selected patients were then termed virtual patients (VPs), and as a collection they were called a virtual population (Vpop).

Since the original algorithm created a plausible population that was naïve to the targeted virtual population distribution, significant computational effort was expended in generating spurious plausible patients outside of the target distribution. Here we propose additional algorithms to improve the generation of the plausible population for more efficient generation of the virtual population. The common extension of our previous approach in each of the tested algorithms is to use information about the target distribution in generating the plausible population. We explored this idea in three ways: (1) a novel method we have termed "nested simulated annealing" (NSA), which repeatedly targets a simulated annealing method to fill portions of the output-space at the correct frequency to approximate the target distribution; (2) a population-based global optimization approach, in this case MATLAB's genetic algorithm (GA); and lastly, (3) an important-sampling technique, which for this application we chose the Metropolis-Hastings (MH) method (Figure 1). We also re-simulated the original method, which uses simulated annealing, for direct comparison to the three new approaches.

2 Methods

The focus of this paper is the evaluation of three approaches for the generation of virtual populations that match empirical distributions of clinical cohorts or populations. This section is organized to describe these three methods (NSA- nested simulated annealing, GA- modified genetic algorithm, and MH- modified Metropolis-Hastings), and how to apply them to generate virtual patients (VPs). This is followed by a description of how the results were analyzed, including a novel metric for assessing the uniqueness of a collection of parameter sets.

2.1 Mathematical Model and Data

Following our previous approach, we test our proposed methods using a publicly-available lipoprotein metabolism model (van de Pas, Woutersen et al. 2012) and clinical data from the National Health and Nutrition Examination Survey (NHANES 2011-2012). In brief, the van de Pas model is a model of cholesterol production by the liver and its transit through the plasma. The focus of the model is on calculating dynamic changes to high-density lipoprotein cholesterol (HDL_c) and "non-high-density lipoprotein cholesterol", which we assume to be equivalent to low-density lipoprotein cholesterol (LDL_c). The goal of this paper is to assess the model's ability to generate physiologically reasonable patients at baseline (i.e., at steady state, before the application of cholesterol therapies). We compare the distributions of LDLc, HDLc, and total cholesterol (TC, or LDL_c+ HDL_c) in our PPs and VPs to the multivariate distributions available in the NHANES database.

2.2 Nested Simulated Annealing

The nested simulated annealing (NSA) approach is a novel method we developed to use information about the target data distribution in the process of generating the plausible population. Previously we optimized the steady-state solutions, x^*, to within biologically reasonable ranges rather than to a specific point using the following cost functional

$$g(\boldsymbol{p}) = \sum_{i=1}^{N} \max\left[\left(x_i^*(\boldsymbol{p}) - \frac{l_i+u_i}{2}\right)^2 - \left(\frac{u_i}{2} - \frac{l_i}{2}\right)^2, 0\right], \qquad (1)$$

where N is the number of states, and l_i and u_i represent the biologically reasonable lower and upper bounds for the ith state variable's steady-state solution. Note that if $x_i^*(\boldsymbol{p})$ is within the biological bounds, then $g(\boldsymbol{p}) = 0$, and the parameter set $\boldsymbol{p}$ is called a plausible patient. This optimization is repeated until a plausible population is created.

Here we modified the cost functional, equation (1), in order to include information about the desired distribution, thus decreasing the number of plausible patients required per VP. Since the NHANES data is well approximated by a multivariate log-

normal distribution, which is roughly an ellipsoidal space, we modified the hypercube to an ellipsoidal shape, in the dimensions for which we have data, that encapsulates the data (with the most extreme data being on the surface of the ellipse). However, an immediate problem with this approach was points at the edge of the ellipse were accepted and generated as frequently as those near the centroid of the data. Hence, to fully use information from the target distribution, we considered multiple ellipsoids nested within one another such that we can control the number of plausible patients generated in each ellipse.

Note that while biological upper and lower bounds exist for all of the nine model outputs, we only have distributional information regarding LDL_C, HDL_C, and TC. As a result, it is necessary to still minimize the original cost functional (1) for the remaining six dimensions to ensure the optimized parameter sets do not produce biologically implausible solutions.

We use the following equation for the contour of an ellipsoid that will encompass the data:

$$(\boldsymbol{X} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{X} - \boldsymbol{\mu}) = c^2, \qquad (2)$$

where $\boldsymbol{X}$ is a 3 × 1 vector representing a point in the log-scaled observable space, $\boldsymbol{\mu}$ and Σ are maximum likelihood estimation of the mean and covariance matrix of the multivariate normal distribution, and c^2 controls the size of the ellipsoid. By letting $\boldsymbol{X}$ be the data point furthest from the mean in (2), we can explicitly calculate the minimum value for c^2 that allows the data to be encompassed by the ellipsoid.

First consider only one ellipsoid region. Then the modified cost functional is

$$h(\boldsymbol{p}) = g(\boldsymbol{p}) - \gamma((\boldsymbol{X} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{X} - \boldsymbol{\mu}) - c^2), \qquad (3)$$

where $g(\boldsymbol{p})$ is given by equation (1) for N = 6, and is calculated for the 6 state variables for which we do not have distributional information. We can interpret this cost functional as forcing these 6 state variables to be within plausible biological bounds while forcing the remaining 3 log-scaled observables to be within the smallest ellipsoid region that encompasses the clinical data. Notice we have an additional parameter, γ, which is simply a scaling factor that is a free parameter of the method.

Now consider several nested ellipsoid regions. The cost functional for each nested ellipsoid is given in (3) but with modified c^2 values in order to control the distribution of the plausible population. The number of nested ellipsoids, R, is another free parameter of this method. Denote the ellipsoids as $E_1 \subset E_2 \subset \cdots \subset E_R$, with the kth ellipsoid centered at the mean $\boldsymbol{\mu}$ and defined as $E_k = \{\boldsymbol{X}: (\boldsymbol{X} -$
$\boldsymbol{\mu}T\Sigma-1\,\boldsymbol{X}-\boldsymbol{\mu}\leq ck2\}.$

We choose the c_k values such there is a k/R probability of the data being observed within the kth ellipse:

$$\int_{E_k} \Phi(x)dx = \frac{k}{R} \quad , \qquad (4)$$

where $\Phi(x)$ is the multivariate normal distribution, and $k = 1 \dots R$. We find an approximate solution, c_k , to this integral, by using a Monte Carlo approach. Equivalently,

$$\int_{E_1} \Phi(x)dx = \frac{1}{R} \qquad (5)$$

$$\int_{E_k \setminus E_{k-1}} \Phi(x)dx = \frac{1}{R}. \quad (6)$$

For this method, we are going to populate each ellipsoid such that the proportion of plausible patients within each ellipsoid matches the target distribution. We therefore need to calculate how many plausible patients are required for each ellipsoid, given a desired total number of plausible patients. Define q_k as the proportion of the total plausible population within the kth ellipsoid. Then, since we assume the observables are approximately uniformly distributed throughout each ellipsoid, we want to solve a system of R equations for each q_k obtained by solving

$$\frac{1}{R} = \sum_{k=j}^{R} \frac{V_j - V_{j-1}}{V_k} q_k, \qquad (7)$$

for $j = 1, \dots, R$. Where V_j is the volume of the jth ellipse. We define $V_0 = 0$. Note that $V_j \propto c_j^n$, where n is the dimension of the multi-dimensional distribution. Then, equation (7) can be re-written as

$$\frac{1}{R} = \sum_{k=j}^{R} \frac{c_j^n - c_{j-1}^n}{c_k^n} q_k, \qquad (8)$$

which can be solved recursively for the q_k (starting with $k = R$, and defining $c_0^n = 0$).

Alternatively, since the NHANES data provided individual level data, the target distributions c_k (and q_k) were calculated empirically (see code, Supplementary Materials).

With the ellipses defined we can generate the plausible population by randomly generating a parameter set in the ranges we defined from a uniform distribution. Then, for a plausible population of size m, for $l = 1$ to m, we optimize from our initial parameter estimate to our final plausible patient via minimizing

$$h_k(\boldsymbol{p}) = g(\boldsymbol{p}) - \gamma((\boldsymbol{X} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{X} - \boldsymbol{\mu}) - c_k^2), \quad (9)$$

where $g(\boldsymbol{p})$ is given by equation (1) for N = 6, and is calculated for the 6 state variables for which we do not have distributional information.

This method requires choosing the number of nested ellipsoids. The greater the number of ellipsoids, the closer the distribution of the plausible population will match that of the clinical population, thus reducing the number of plausible patients required per virtual patient. However, an increase in the number of ellipsoids increases computation time for the plausible population. We then use rejection sampling to select the virtual patients from the plausible population.

2.3 Modified Genetic Algorithm

A commonly used population-based approach for optimizing nonlinear models is a genetic algorithm (GA), (Golberg 1989, Conn, Gould et al. 1991). For our problem, we created our plausible population using MATLAB's *ga* function (MATLAB 2016). This algorithm first creates an initial population, where each patient is generated from a uniform distribution bounded by biologically reasonable values. The algorithm then assigns a fitness value to each patient using a cost (fitness) functional. Similar to the nested simulated annealing methods, we modify equation (1) by incorporating information about the desired distribution. Specifically, the cost functional is given by

$$H(\boldsymbol{p}) = \begin{cases} g(\boldsymbol{p}) & \text{if } g(\boldsymbol{p}) > 0 \\ -\gamma(\boldsymbol{p}) & \text{otherwise,} \end{cases} \quad (10)$$

where $\gamma(\boldsymbol{p})$ is the likelihood of a given log-scaled observable value $\boldsymbol{X}$ given by

$$\gamma(\boldsymbol{p}) = \frac{1}{\sqrt{(2\pi)^3 |\Sigma|}} \exp\left(-\frac{1}{2}(\boldsymbol{X} - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{X} - \boldsymbol{\mu})\right). \quad (11)$$

This cost functional can be interpreted as forcing all state variables to be within plausible biological bounds and additionally assessing a penalty as the log-scaled observables deviate from $\boldsymbol{\mu}$.

In implementing this algorithm, we start with an initial population of individuals, where each individual is created by random selection of parameter values from a uniform distribution with biologically-reasonable bounds. As the algorithm progresses, children are created for each generation; those with a cost functional value $H(\boldsymbol{p}) < 0$ become plausible patients. Since many plausible patients are created each generation, there is no way to specify the exact number of plausible patients generated. Thus we must preset the minimum number of plausible patients desired, but in practice we tended to generate slightly more than sought (see Supplementary Code). Once the plausible population is created, we follow the remaining steps in Allen et al. and use rejection sampling to determine the virtual

population from the plausible population.

2.4 Modified Metropolis-Hastings

First, recall the original Metropolis-Hasting (MH) algorithm to approximate a desired distribution, for a review see for example (Robert 2015). Let $\pi(\boldsymbol{p})$ be our desired target multivariate probability distribution for the vector $\boldsymbol{p}$ (in this case $\boldsymbol{p}$ is a parameter set of the model). The MH algorithm generates a sequence of $\boldsymbol{p}$, such that the distribution of this sequence, $\{\boldsymbol{p_0}, \dots, \boldsymbol{p_N}\}$ converges to $\pi(\boldsymbol{p})$ as $N \to \infty$. Let $Q(\boldsymbol{p}, \boldsymbol{q})$ be some symmetric proposal distribution, which is interpreted as generating a proposed value $\boldsymbol{q}$ from $Q(\boldsymbol{p}, \boldsymbol{q})$ when the process is at value $\mathbf{p}$. Then the original Metropolis Hastings algorithm is as follows:

1. Generate an initial vector $\boldsymbol{p_0}$, set $i = 1$.
2. Generate a proposed vector $\boldsymbol{p}^* \sim Q(\boldsymbol{p_{i-1}}, \boldsymbol{p}^*)$.
3. Calculate the probability $\boldsymbol{p}^*$ is accepted, $\alpha = \min\left(1, \frac{\pi(\boldsymbol{p}^*)}{\pi(\boldsymbol{p_{i-1}})}\right)$.
4. Generate $y \sim U(0,1)$, if $y \leq \alpha$, set $\boldsymbol{p_i} = \boldsymbol{p}^*$. If $y > \alpha$, set $\boldsymbol{p_i} = \boldsymbol{p_{i-1}}$.
5. Repeat steps 2 to 4 for $i = 1, \dots, N$ to collect $\{\boldsymbol{p_0}, \dots, \boldsymbol{p_N}\}$ as a sampling from the target distribution.

This MH algorithm approximates the target distribution $\pi(\boldsymbol{p})$ by randomly sampling from it. At first glance, this approach appears immediately applicable to the problem at hand and will generate a plausible population that will converge to the virtual population as $N \to \infty$. However, this algorithm requires modification because we do not know $\pi(\boldsymbol{p})$ *a priori*; i.e., we do not know how the parameters sets should be distributed such that the model, when simulated using those parameters, matches the data.

We rewrite our target distribution as $T(\boldsymbol{X_p})$, where $\boldsymbol{X}$ is the observable outcomes generated by the model M, using a parameter set (which in this case is in the log-space, so $\boldsymbol{X_p} = \log M(\boldsymbol{p})$). Then our algorithm becomes

1. Generate an initial vector $\boldsymbol{p_0}$, set $i = 1$.
2. Generate a proposed vector $\boldsymbol{p}^* \sim Q(\boldsymbol{p_{i-1}}, \boldsymbol{p}^*)$, write $Q_m(\boldsymbol{p_{i-1}}, \boldsymbol{p}^*) = M(Q(\boldsymbol{p_{i-1}}, \boldsymbol{p}^*))$.
3. Calculate the probability $\boldsymbol{p}^*$ is accepted, $\alpha = \min\left(1, \frac{T(\boldsymbol{X_{p^*}})Q_m(\boldsymbol{p}^*, \boldsymbol{p_{i-1}})}{T(\boldsymbol{X_{p_{i-1}}})Q_m(\boldsymbol{p_{i-1}}, \boldsymbol{p}^*)}\right)$,

   assume $\alpha \sim \min\left(1, \frac{T(\boldsymbol{X_{p^*}})}{T(\boldsymbol{X_{p_{i-1}}})}\right)$.
4. Generate $y \sim U(0,1)$, if $y \leq \alpha$, set $\boldsymbol{p_i} = \boldsymbol{p}^*$. If $y > \alpha$, set $\boldsymbol{p_i} = \boldsymbol{p_{i-1}}$.
5. Repeat steps 2 to 4 for $i = 1, \dots, N$ to collect $\{\boldsymbol{p_1}, \dots, \boldsymbol{p_N}\}$ as a sampling from the target distribution.

In the canonical version of the MH algorithm α is independent of the proposal distribution Q because it is symmetric and cancels out of the equation. In the modified version above, Q_m is unknown and, in fact, is unlikely to be symmetric. In order to proceed we assume that Q_m is approximately symmetric $Q_m(\boldsymbol{p}^*, \boldsymbol{p}_{i-1}) \sim Q_m(\boldsymbol{p}_{i-1}, \boldsymbol{p}^*)$, so that we can calculate α as above. Because of this approximation it is still necessary, following our previously published algorithm, to apply acceptance-rejection sampling to determine the virtual population from the plausible population.

2.5 Assessing Similarity Between Virtual Patients

It is desirable to examine the parameter space in the virtual population to ensure heterogeneity of the virtual patients (while still reproducing available data). In our method this was previously ensured by generating VPs independently and from different initial parameter estimates. However, this is not necessarily the case for the GA and MH methods.

To assess the diversity of a virtual population we devised a test metric $d(\boldsymbol{p}_i, \boldsymbol{p}_j)$ which scores how similar two VPs are. By bootstrapping sampling pairs of VPs from a given virtual population, we built up a distribution for d and could compare the resultant cumulative density function (CDF) for each method. The test metric d is simply the normalized dot-product of $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ after they are scaled and shifted:

$$d(\boldsymbol{p}_i, \boldsymbol{p}_j) = \frac{\hat{\boldsymbol{p}}_i \hat{\boldsymbol{p}}_j}{|\hat{\boldsymbol{p}}_i||\hat{\boldsymbol{p}}_j|} \qquad (12)$$
$$\hat{\boldsymbol{p}} = \mathbf{diag}(\boldsymbol{V}^{-1}(\boldsymbol{p} - \boldsymbol{l})) - \frac{1}{2}$$

where $\boldsymbol{V}$ is a diagonal matrix such that $\boldsymbol{v}_{ii} = \boldsymbol{u}_i - \boldsymbol{l}_i$. Hence, $\mathbf{diag}(\boldsymbol{V}^{-1}(\boldsymbol{p} - \boldsymbol{l}))$ uses the defined upper and lower bound for each parameter (the elements of $\boldsymbol{u}$ and $\boldsymbol{l}$ respectively), to scale each parameter in $\boldsymbol{p}$ to be between 0 and 1. To ensure that $d \in [-1,1]$ we further subtract ½ from each element. This means that, in principle, $\hat{\boldsymbol{p}}$ can be orientated in any direction in m-dimensional space (where m is the number of parameters). This also means that if the elements of $\boldsymbol{p}$ are sampled uniformly between the upper and lower bounds that, by symmetry, the expected value of the distribution should be zero (i.e., the CDF crosses 0.5 at $d = 0$). This is the optimal parameter set in terms of diversity, but may not be achievable given the constraints applied to the model. Conversely, if we generate virtual populations from very similar parameter sets then the distribution will be right-shifted towards d=1.

2.6 Assessing Goodness of Fit

The goodness of fit (GoF) to the empirical target distribution was assessed by using the Kolmogorov-Smirnov statistic for each marginal distribution in one dimension:

$$\text{GoF} = \sum \sup |F_i(x) - D_i(x)| \qquad (13)$$

where $F_i(x)$ is the empirical cumulative distribution function for the ith model observable that we are fitting the observed cumulative distribution of the data, $D_i(x)$. Note that for a perfect fit GoF=0, and that GoF $[0, n]$, where n is the number of distributions being fitted.

2.7 Source code

All algorithms were implemented in MATLAB 2016b (v9.1.0.441655) using the Global Optimization Toolbox (v3.4.1) where a pre-packaged routine was available (e.g., GA, SA). The ODE model was implemented in SimBiology (v5.5). The K-S Test for GoF utilizes MATLAB's Statistics and Machine Learning Toolbox (v11.0). The full source code is available for download from a GitHub repository (Rieger 2017).

All simulations were performed sequentially on a MacBook Pro with an Intel Core i7 2.9 GHz processor and 16 GB of RAM.

3 Results

3.1 Comparing the various algorithms

To compare the three proposed algorithms (NSA, GA and MH) we evaluated four metrics to measure performance of the algorithm compared to the original simulated annealing method. For each algorithm, we evaluated:

1. **Efficiency:** How many PPs were needed to achieve a certain GoF of the final Vpop to the observables?
2. **Computational Cost:** How fast was the generation of PPs and VPs?
3. **Diversity:** Are the VPs parametrically similar or do they maintain the parametric heterogeneity of the PPs?
4. **Convergence**: Do the methods benefit from the acceptance/rejection step or can a VPop be generated directly?

3.2 Comparison of algorithms for efficiency of yield

For each algorithm, we targeted generation of between 100 and 10,000 total PPs; those PPs were then converted into VPs through the acceptance/rejection algorithm. The GoF of the resulting VPop was calculated as discussed in Methods. By comparing the GoF achieved for the VPops with varying PPs (Figure 2) we find that as the number of PPs→10,000+, all of the algorithms generated essentially indistinguishable GoFs for the final VPop (albeit with different VPs in each VPop). However, the three new algorithms were more efficient than the original SA method, especially when the number of PPs < 1,000. In fact, VPops generated with as few as 100 PPs could have similar fits to the observable data as the SA method with 500+ PPs.

3.3 Comparison of algorithms for computational cost

Even if an algorithm can generate the same GoF through far fewer PPs than the original SA algorithm, this does not necessarily mean the process was computationally more efficient. We further compared each method based on the clock time (evaluated via MATLAB's *tic/toc* functions) required to generate a VPop from 10,000 plausible patients (Figure 3). While the NSA method was arguably superior based on yield, this algorithm required approximately the same amount of time to execute as the SA method. Based on time, the MH and GA were the fastest algorithms and the SA remains among the least efficient.

3.4 Comparison of algorithms for parametric diversity of the final virtual population

An advantage of the SA algorithm is its ability to generate VPops that maintain most of the parametric diversity of the original PPs (Supplementary Figure S1-2). This diversity in the VPop is an essential feature for QSP models since they are often utilized in simulation of clinical trials involving novel therapies. If the underlying parameters of VPs are highly similar/correlated, clinical trial simulations performed with the VPop may incorrectly predict a very narrow range of therapeutic response. Therefore, we need to ensure that as we introduce new algorithms, we do not trade parametric diversity for computational gains. We measured the diversity of the VPops generated by each algorithm by uniformly sampling pairs of VPs and calculating the dot product between each set of parameters (see Methods). As a reference point, we included a set of uniformly, randomly generated model parameters (Figure 4). The closer each algorithm's final VP parameter distribution is to the random reference, the more diverse we considered the set of VPs in the Vpop. For this criterion, the SA method was found to have the most diversity, closely followed by NSA and MH. The GA method showed distinct rightward shifts in its distribution, indicating that fewer independent parameter sets were identified in the generation of the VPops. Supplementary figures show the violin plot for each method for both the PPs and VPs (Figures S2, S4, S6, and S8).

3.5 Convergence of the algorithms to the data

In contrast to the original SA method, each of the methods tested use information about the desired population distribution and thus requires fewer plausible patients to achieve an acceptable fit for the VPop (Figure 2). To evaluate if the final selection step was still required as part of the algorithm workflow, we compared the GoF for each of the methods before and after the acceptance/rejection step, starting with ~10,000 PPs in each case (Figure 5). As expected, the largest improvement in VPop fit from the PPs to VPs was for the SA method; however, all of the methods showed at least a 3-fold improvement in GoF through the final selection step. The NSA method showed the best initial fit for its plausible patients to the NHANES data, reasonably reproducing both the 1-dimensional and 2-dimensional histograms

before the selection step (Figure S3A-F). These fits were approximately equivalent to the final VPop fit for the SA method starting with 1,000 PPs.

4 Discussion

By design, most quantitative systems pharmacology models are not identifiable from available data. While it would be desirable to have parameters well determined and characterized, the uncertainty in parameter values in QSP models often reflects our current knowledge (or lack thereof) of human (patho)-physiology. Therefore, with this perspective, we can use these models in a hypothesis-generation/testing mode to explore how these knowledge gaps translate to uncertainties in clinical outcomes and clinical trial design. In our opinion, the most thorough and robust way of doing this is by generating the most diverse virtual populations given the available data.

Exploring the parameter space of under-determined quantitative systems pharmacology models remains a challenge but it is essential for safety and efficacy predictions for novel compounds. Here we presented three methods for generating diverse parameter sets in a QSP model. While this exercise is by no means an exhaustive exploration of global optimization techniques, each algorithm improved at least one of the testing metrics compared to our previously published SA method. The seemingly simple question of which method is "the best" cannot be definitively answered here but it is important to be aware of the pros and cons for each and potential steps to improve performance.

We have previously discussed the advantages and disadvantages of using the SA method for this application (Allen, Rieger et al. 2016). In comparison to other methods tested it was the slowest (or tied with NSA), required the most plausible patients for a quality fit but on the plus side it also generated the most (or tied with NSA) diverse virtual population with the fewest imposed correlations. The ease of implementation was also an advantage for SA. As implemented, the algorithm required no prior knowledge of the final virtual population distribution and there was a minimal set of tuning parameters required, most of which were default MATLAB options (i.e., no arbitrary decision about number of ellipses, number of generations). As such, the algorithm remains relevant as a "first try" for generating VPs. Furthermore, it is the only method that can be run (at least in part) without prior knowledge of the target distribution. This relaxed requirement can make it an attractive choice for pre-computing plausible populations or for exploring how the parameter space relates to the model output (for example, identifying parameters in the model that can give rise to sub-populations of interest).

The NSA approach iterates on the SA method by utilizing prior knowledge of the final parameter distribution and forcing the algorithm to regions with the most desired patient density. This method is likely most efficient when the target distribution is approximately ellipse-shaped. Fortunately, for the example here, the

target distribution was well approximated as a multivariate lognormal. The viability of this method for more eccentric or bimodal distributions would need to be studied by applying it to other case studies. However, such distributions are less commonly observed in clinical trials. For our case study, the computational cost to generate a plausible population was comparable between the NSA and SA methods; however, the NSA approach demonstrated vastly improved yield (VP per PP), which facilitated an overall more efficient VPop generation process. The algorithm was as good, or better, than the other methods tested for direct generation (pre-selection/rejection) of a reasonable Vpop, without the imposed correlations found in VPops generated by GA. While essentially the same as the SA method to implement, there is a problem-specific choice for the number of ellipses to use. Here we used five regions, regardless of the number of plausible patients being generated. Fine-tuning based on the model/number of plausible patients may potentially improve performance.

Metropolis-Hastings is unique amongst the approaches we tried in that it is a Markov Chain, which should imply some degree of correlation between the plausible patients. The advantage of this technique was an increase in speed and compared to the two methods based on SA; however, there also was a right-shift in the dot product cumulative distribution, implying a slightly less diverse final population. Methods have been published to attempt to reduce this correlation (Santoso, Phoon et al. 2011) and to improve performance in higher dimensions (Betancourt 2017), but we chose to evaluate only the common form of the algorithm and to leave further exploration for future improvements. While straightforward to implement, MH requires the choice of a proposal distribution. As noted in the Methods, because we indirectly sample the distribution of the observables by first sampling the parameter space and then generate the observables through model simulation we do not have direct control over the choice of a proposal distribution. The implications of this for direct convergence of this method will depend on the symmetry of the observable distribution (induced by the parameter sampling) around every point on the Markov chain. In this case, the approximation we assumed (see Methods) appeared to hold sufficiently for the MH algorithm to approximate the empirical distributions. However, the final Vpop was improved by acceptance/rejection sampling.

The genetic algorithm was very similar to MH in that, compared to the SA-based methods, a 10-fold improvement in computational speed was achieved at the cost of some lost heterogeneity in the final virtual population. The GA is easy to implement for QSP models and the supplied routine with MATLAB's Global Optimization Toolbox was sufficient for our purposes. GA requires some problem-specific decisions, which may affect overall performance; for example, the size of the population, number of generations, and mutation rate can be adjusted as needed.

The curse of under-determined models has led to a long history of using different global optimization techniques for generating parameter sets within the bounds of the data (van de Pas, Woutersen et al. 2012, Gadkar, Budha et al. 2014, Hallow, Lo et

al. 2014). Use of global optimization techniques often feels like more of an art than a science due to how problem-specific their application can be. For this reason, we examined several algorithms with different approaches for exploring constrained, multidimensional parameter spaces. Requiring only minimal tuning, each of these algorithms successfully explored the range of our 23-dimensional parameter space and generated reasonable PPs. The choice of algorithm to use for a new problem, particularly one with higher dimensions and a less Gaussian set of observations, will need to be evaluated on a case-by-case basis. For example, for models that are slow to simulate the most constraining factor is computational cost. In this case the MH or GA approaches may be the most successful; however, as we have shown, without adaptation these methods come at a cost of diversity in the final Vpop.

We hope that the results presented here will provide a guide to selection and implementation of these algorithms to facilitate the generation of robust Vpops in mathematical models of (patho)-physiology.

5 Abbreviations

GA – Genetic Algorithm
GoF – Goodness of Fit
MH – Metropolis-Hastings
NSA – Nested Simulated Annealing
SA – Simulated Annealing
PP – Plausible Patient
VP – Virtual Patient
VPop – Virtual Population

6 Author contributions

RA conceived of the original algorithm. CM, RA, and TR conceived of the updated algorithm objectives. AG, GC, LB, RW, YC1, YC2, and YL selected and coded the new algorithms and performed the initial proof-of-concept testing. CM, RE, HB, RA, and TR supervised and advised the initial work. RA, RE, RW, TR, HB, and CM drafted the manuscript. All authors reviewed, revised, and approved of the final manuscript.

7 Conflict of interest

TRR, RJA, and CJM were employees of Pfizer Inc. during the completion and analysis of this study.

8 Funding

This research was supported in part by the National Institute on Alcohol Abuse and Alcoholism under grant number 1R01AA022714-01A1, in part by the Air Force Office of Scientific Research under grant number AFOSR FA9550-15-1-0298, and in part by the US Department of Education Graduate Assistance in Areas of National Need (GAANN) under grant number P200A120047.

Pfizer Inc. supported the research of TRR, RJA, and CJM.

9 Acknowledgements

The authors thank Ilse Ipsen and Thomas Gehrmann for organizing the 2016 SAMSI Workshop at North Carolina State University, which initially facilitated this collaboration.

10 References

Allen, R. J., T. R. Rieger and C. J. Musante (2016). "Efficient Generation and Selection of Virtual Populations in Quantitative Systems Pharmacology Models." CPT Pharmacometrics Syst Pharmacol **5**(3): 140-146.

Betancourt, M. (2017) "A Conceptual Introduction to Hamiltonian Monte Carlo." ArXiv e-prints **1701**, 60.

Conn, A. R., N. I. Gould and P. Toint (1991). "A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds." SIAM Journal on Numerical Analysis **28**(2): 545-572.

Gadkar, K., N. Budha, A. Baruch, J. D. Davis, P. Fielder and S. Ramanujan (2014). "A Mechanistic Systems Pharmacology Model for Prediction of LDL Cholesterol Lowering by PCSK9 Antagonism in Human Dyslipidemic Populations." CPT Pharmacometrics Syst Pharmacol **3**: e149.

Golberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. Reading, MA, Addison-Wesley Publishing Co.

Hallow, K. M., A. Lo, J. Beh, M. Rodrigo, S. Ermakov, S. Friedman, H. de Leon, A. Sarkar, Y. Xiong, R. Sarangapani, H. Schmidt, R. Webb and A. G. Kondic (2014). "A model-based approach to investigating the pathophysiological mechanisms of hypertension and response to antihypertensive therapies: extending the Guyton model." AJP: Regulatory, Integrative and Comparative Physiology **306**(9): R647-R662.

MATLAB (2016). MATLAB and Global Optimization Toolbox Release 2016b. Natick, Massachussetts, USA.

Musante, C. J., S. Ramanujan, B. J. Schmidt, O. G. Ghobrial, J. Lu and A. C. Heatherington (2017). "Quantitative Systems Pharmacology: A Case for Disease Models." Clin Pharmacol Ther **101**(1): 24-27.

NHANES. (2011-2012). "National Health and Nutrition Examination Survey (NHANES) Data." 2011-2012. from http://www.cdc.gov/nchs/nhanes.htm.

Rieger, T. (2017). "Source code repository for vpop-gen, code for evaluating methods to fit virtual populations." from https://zenodo.org/record/999552.

Robert, C. P. (2015). "The Metropolis-Hastings algorithm." ArXiv e-prints **1504**: arXiv:1504.01896.

Santoso, A., K. Phoon and S. Quek (2011). "Modified Metropolis–Hastings algorithm with reduced chain correlation for efficient subset simulation." Probabilistic Engineering Mechanics **26**(2): 331-341.

van de Pas, N. C. A., J. A. C. Rullmann, R. A. Woutersen, B. van Ommen, I. M. C. M. Rietjens and A. A. de Graaf (2014). "Predicting individual responses to pravastatin using a physiologically based kinetic model for plasma cholesterol concentrations." Journal of Pharmacokinetics and Pharmacodynamics **41**(4): 351-362.

van de Pas, N. C. A., R. A. Woutersen, B. van Ommen, I. M. C. M. Rietjens and A. A. de Graaf (2012). "A physiologically based in silico kinetic model predicting plasma cholesterol concentrations in humans." The Journal of Lipid Research **53**(12): 2734-2746.

11 Figures

11.1 Flowchart of algorithm

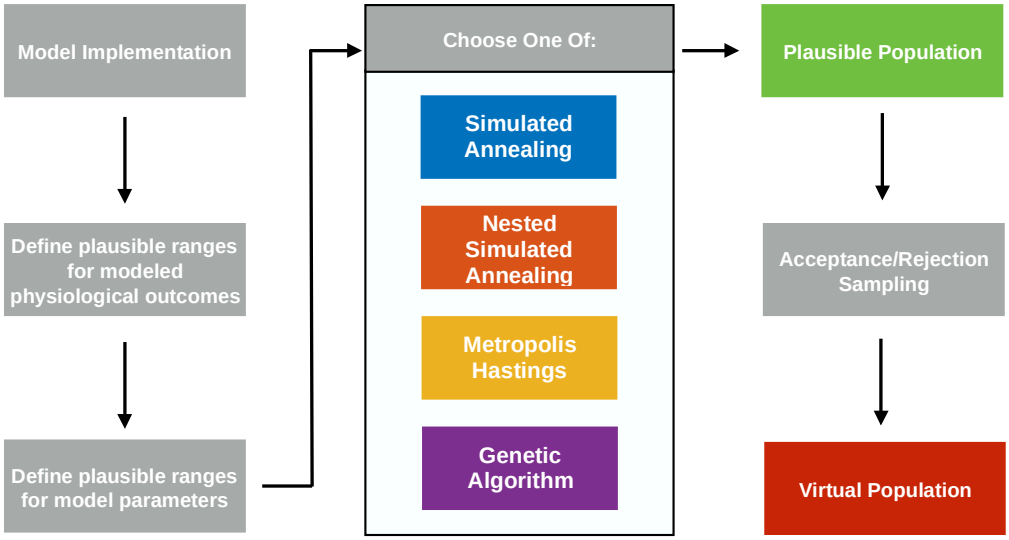

Figure 1. Flowchart of algorithm. The initial problem setup is shared by each of the methods attempted, where biologically plausible ranges are placed on model parameters and states. Once the problem is setup, the generation of plausible patients is carried out by one of four algorithms. Post-plausible patient generation, each algorithm follows the same acceptance/rejection sampling steps to create the virtual population.

11.2 Goodness-of-fit vs. number of plausible patients

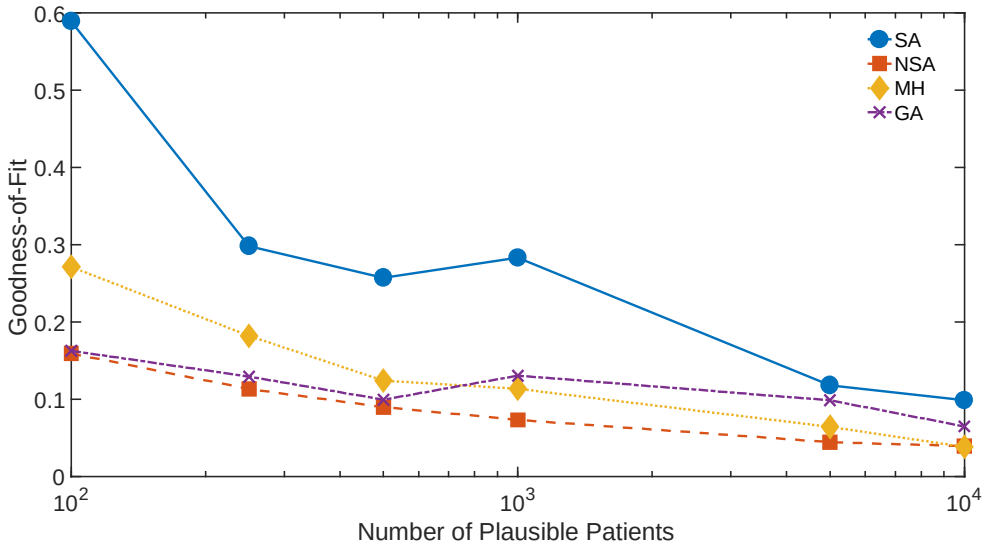

Figure 2. Goodness-of-fit (lower number is better) of the final virtual population vs. the number of plausible patients generated for each method. SA = simulated annealing (blue circles), NSA = nested simulated annealing (red squares), MH = Metropolis-Hastings (yellow diamonds), GA = genetic algorithm (purple x's).

11.3 Comparison of VP and PP generation time for each method

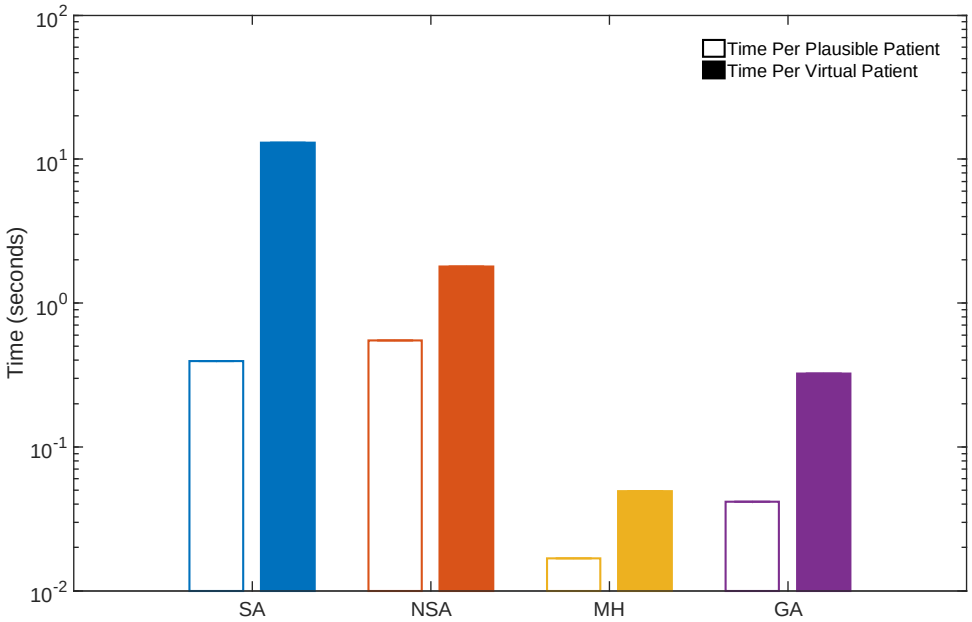

Figure 3. Comparison of the time/plausible patient (open bars) or time/virtual patient (filled bars) for each method. For each method, ~10,000 plausible patients were generated and then a virtual population was selected from those plausible patients. Time was calculated via the functions *tic/toc* in MATLAB. SA = simulated annealing (blue), NSA = nested simulated annealing (red), MH = Metropolis-Hastings (yellow), GA = genetic algorithm (purple).

11.4 Diversity of VPs in final Vpops

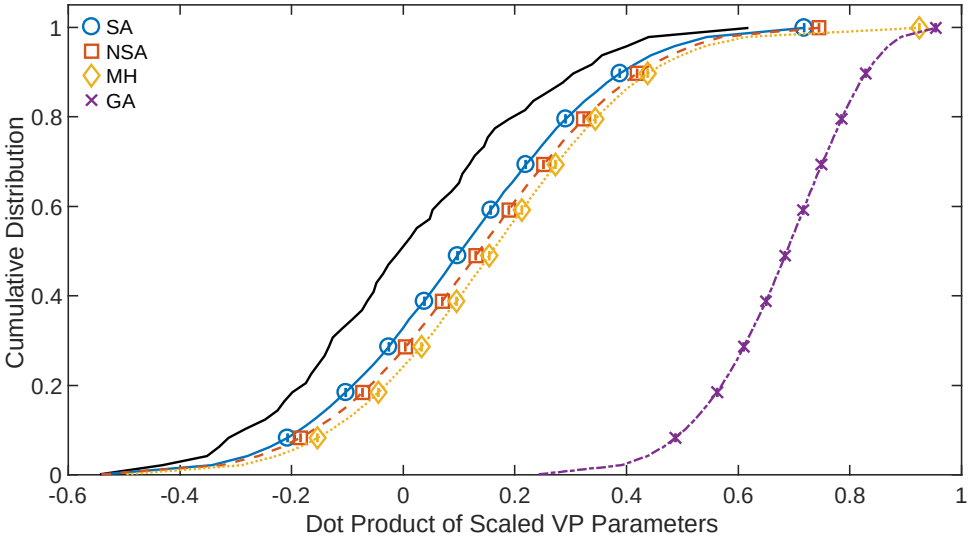

Figure 4. Cumulative distribution vs. the dot-product of the vector of virtual patients' (VPs) parameters to assess the diversity of parameter values in the virtual population. For each method, 10,000 dot-products of randomly chosen VPs were calculated and the cumulative distribution plotted. As a positive control, a set of parameters from a uniform distribution was generated (solid, black line). Distributions closer to the uniform random control indicate a more diverse set of VPs. Distributions skewed towards the right indicates a more uniform set of VPs. SA = simulated annealing (blue circles, solid), NSA = nested simulated annealing (red squares, dashed), MH = Metropolis-Hastings (yello diamonds, dotted), GA = genetic algorithm (purple x's, dashed).

11.5 Efficiency of the acceptance/rejection algorithm

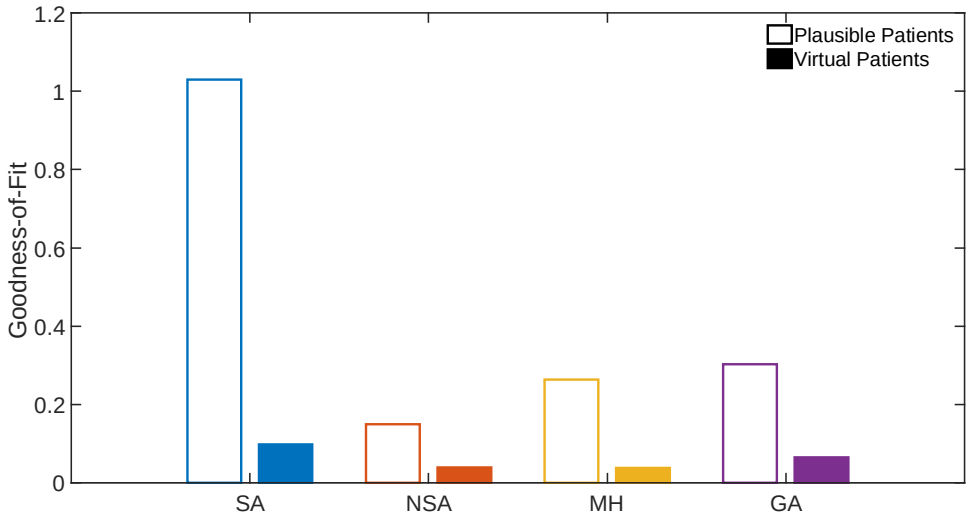

Figure 5. Improvement of goodness-of-fit (lower number is better) from the plausible patients (open bars) → virtual patients (filled bars) for each method starting from ~ 10,000 plausible patients. SA = simulated annealing (blue), NSA = nested simulated annealing (red), MH = Metropolis-Hastings (yellow), GA = genetic algorithm (purple).