

# A genetic algorithm based global search strategy for population pharmacokinetic/pharmacodynamic model selection

Mark Sale<sup>1,2</sup> & Eric A. Sherer<sup>3</sup>

<sup>1</sup>Next Level Solutions, LLC, Raleigh NC, <sup>2</sup>Modeling and Simulation, Kinetigen Inc., Research Triangle Park, NC and <sup>3</sup>Chemical Engineering, Louisiana Tech University, Ruston, LA, USA

## Correspondence

Dr Mark Sale MD, Department of Medicine, Indiana University, 1001 West 10th St, W7138, Indianapolis, IN 46202, USA.

Tel.: +1 31 7630 7868  
Fax: +1 31 7630 8185  
E-mail: mark@nextlevelsolns.com

## Keywords

genetic algorithm, NONMEM, pharmacokinetics

## Received

28 February 2013

## Accepted

4 June 2013

## Accepted Article

## Published Online

17 June 2013

The current algorithm for selecting a population pharmacokinetic/pharmacodynamic model is based on the well-established forward addition/backward elimination method. A central strength of this approach is the opportunity for a modeller to continuously examine the data and postulate new hypotheses to explain observed biases. This algorithm has served the modelling community well, but the model selection process has essentially remained unchanged for the last 30 years. During this time, more robust approaches to model selection have been made feasible by new technology and dramatic increases in computation speed. We review these methods, with emphasis on genetic algorithm approaches and discuss the role these methods may play in population pharmacokinetic/pharmacodynamic model selection.

## Introduction

Population pharmacokinetic/pharmacodynamic (PK/PD) model building is the process by which theoretical understanding of the pharmacology of a drug and empiric analysis of experimental data yields a set of equations that describes the pharmacokinetics and/or pharmacodynamics of a population of subjects taking a drug. In a more general sense, PK/PD model building is an optimization problem. The goal of this optimization problem is to find the model (e.g. number of compartments, variance terms, lag time, indirect response model, covariates, mixture models etc.) that best describes the data according to some objective (e.g. goodness of fit) or subjective (e.g. it is biologically reasonable) criteria.

Historically, the algorithm used for PK/PD modelling building begins by testing a trivial model (often a one compartment, first order absorption model, with no covariates) and then sequentially adding single features to that model and testing if that addition results in a significantly better description of the data. This process has been

used since the software package NONMEM®<sup>1</sup> (Nonlinear Mixed Effect Model) was released in the late 1970s. Figure 1 shows a diagram of the model building algorithm from a recent NONMEM manual.<sup>2</sup> This step wise, or forward addition approach to model building, is well established in linear [1] and logistic regression [2]. Mixed effect models, however, are more complex than linear models.

The model building process for mixed effects pharmacokinetic models typically starts with selecting structural model features such as compartments and lag times. With the basic structure of the model set, a similar step wise addition algorithm is then used to add covariates. The decision about whether to include a feature (a compartment, a lag time or a covariate) is typically based on a combination of objective criteria (most importantly an improvement in the goodness of fit criteria) and subjective criteria (does the addition improve some bias seen in

<sup>1</sup>NONMEM is the property of Icon PLC.

<sup>2</sup>Original drawing by the NONMEM project group, Lewis Sheiner, University of California, San Francisco, reproduced with permission from Icon PLC.

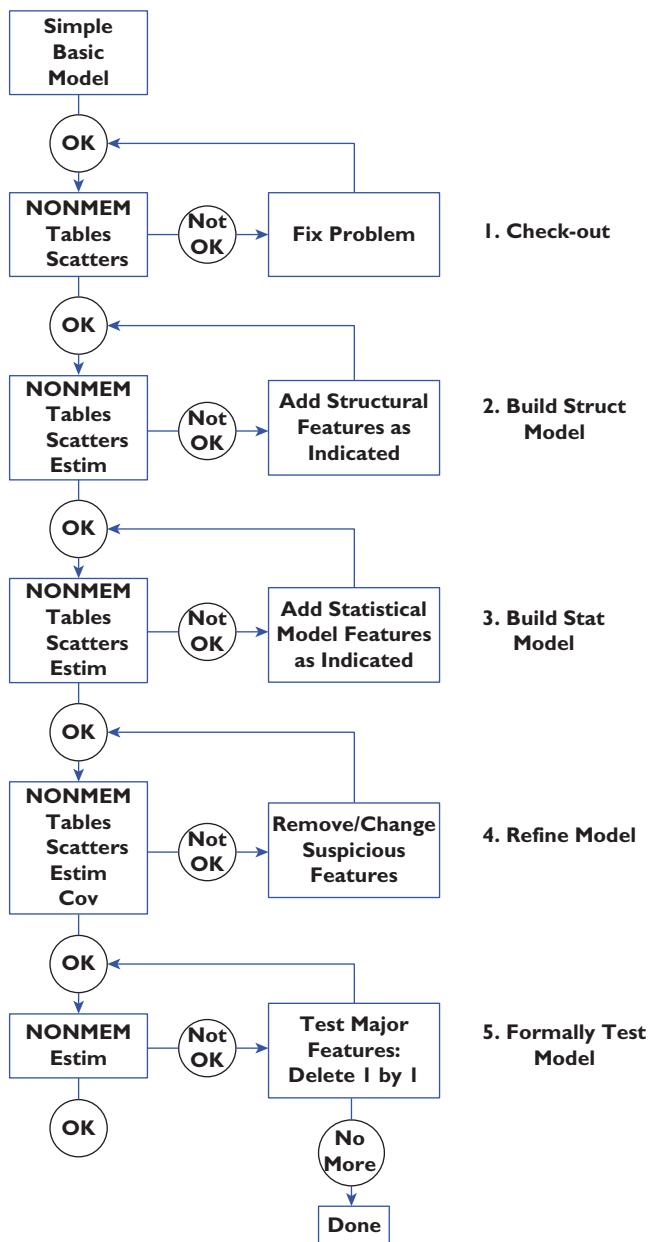

**Figure 1**

Diagram of model building algorithm from volume 5 NONMEM manuals. Reproduced with permission from Icon PLC. In the original description of the algorithm, statistical features (variance terms) were added after the structure was final for practical reasons

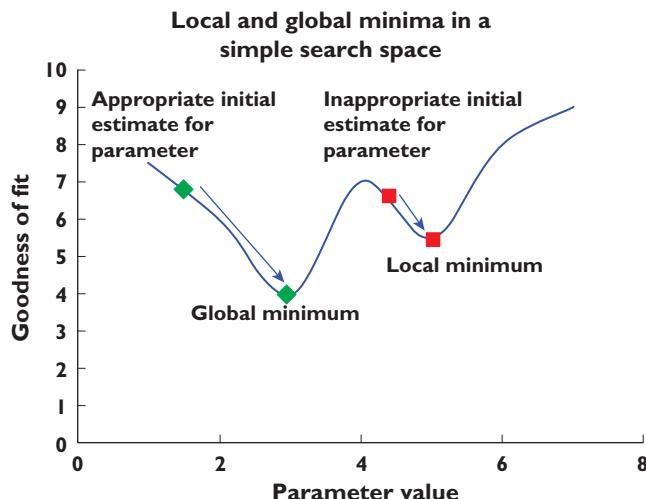
diagnostic plots) for the models with vs. without the additional features. While this sequential approach to model building has been used historically, there is no statistical basis for this particular sequence. Rather, the addition of statistical components at the end of the model building process is due to the finding that, in early versions of NONMEM, adding additional variance features added greatly to the computational time. Also, the run time costs on the

shared mainframe originally used to run NONMEM at UCSF in the 1970s were lower if the executable file was smaller. This provided a practical, rather than statistical, advantage to having fewer random effects because the size of many of the large arrays in NONMEM depends on the number of random effects.<sup>3</sup> This algorithm has remained largely unchanged for more than 30 years despite the fact that at least some of the rationale for it has long since changed.

This PK/PD model building algorithm is, in many ways, analogous to the parameter optimization algorithm used in many of the population pharmacokinetic software packages. For example, the minimization algorithm, which is used for the minimization step in NONMEM [3], starts with initial estimates for all parameters and then makes a small change in the value of each parameter. The 'goodness' (minus twice the log likelihood of the data given the model,  $-2\text{ll}$ , in this case) of the resulting model is then calculated. The search then moves in the direction of values in which the goodness of fit statistic is better. This continues until no further improvement is seen. Analogously, the traditional step wise PK/PD model building algorithm starts with an initial model which is usually very simple. The algorithm then makes a small change in the model, usually adding a single effect (e.g. a peripheral compartment, a single covariate relationship) and calculates the 'goodness' (by objective and subjective criteria) of the resulting model. Changes that significantly improve the model are retained and the process is repeated until no further significant improvement is seen. This step wise approach to model optimization is referred to as 'hill climbing' because, as in the case of parameter estimation, the optimization algorithm is constantly moving in the direction of the highest rate of improvement in the 'goodness' of the model.

It is well established that the selection of appropriate initial estimates are important in non-linear regression to avoid local minima. Because of its similarity to gradient-based parameter optimization, the minima found using 'hill climbing' model structure optimization also depends on the initial starting point. Figure 2 demonstrates the risk of arriving at a local minimum. The algorithm finds the slope of the objective function at the initial estimate and proceeds downhill, ultimately arriving at a minimum. However, whether the minimum is the global minimum or a local minimum depends on the initial guess. Any change in the sign of the gradient between the initial estimates and the global minimum will provide an effective barrier (a 'peak', or in multiple dimension, a 'ridge') to the exploration of the region around the global minimum, as crossing that region would require the algorithm to move uphill, rather than downhill. Typically (although not always the case) the closer the initial estimates are to the global minimum, the more likely that the assumption of a monotonically down-hill objective function surface is met. This assumption of a

<sup>3</sup>Personal communication, Alison Boeckmann.

**Figure 2**

Effect of initial parameter estimate on parameter identification. The quasi-Newtonian algorithm proceeds locally downhill. If inappropriate initial estimates are provided the algorithm will find a local minimum rather than a global minimum

monotonically downhill objective function surface is known as the convexity assumption [3], that no matter where you start, it will always be continuously downhill to the global minimum. For relatively simple models, especially with rich data, there are methods for obtaining initial estimates (e.g. curve stripping). However, adequate initial estimates for complex models is a challenge. Frequently the best solution is simply to try many initial estimates and chose the set that results in the best goodness of fit. For complex models, there is no way of knowing if the resulting minimum is a global or local minimum without testing every possible scenario.

Similarly, there is risk of being caught in a local minimum in a hill climbing search algorithm in a discrete search space, especially if the starting point of the search is far from the global minimum. The fundamental assumption behind a hill climbing algorithm is the independence of the different effects in the search space [4]. This independence between effects is the correlate of the convexity assumption in gradient-based optimization. For example, in the case of PK/PD models, the central assumption is that volume will be (or will not be) a function of weight regardless of presence of other features (number of compartments, variance terms, lag time, mixture models etc.), and so it does not matter when this effect is tested (e.g. before or after the number of compartments, before or after the variance terms, etc.), the result will be the same. Importantly, this assumption, like the convexity assumption in real space, has been shown to be incorrect [5]. Also, the structural model search is typically started at a point of convenience (a trivial model) rather than, as in the case of

parameter estimation, near the anticipated global minimum, resulting in a higher likelihood that the convexity assumption is not met. Unlike the common practice of using multiple initial estimates for parameter estimation, in traditional model selection, it is uncommon to start the model search from multiple initial models.

An important difference between parameter estimation in NONMEM and the process of searching for the optimal structural model is that the parameter space is generally the set of positive real numbers while the model structure space is discrete valued. That is, a model cannot include fractions of compartments (e.g. 1.462 compartments) but the number of compartments must be an integer (such as a one or two compartment model). Similarly a between subject variance term is either present or absent. Because the structural search space is discrete, there is no gradient along which to search. Therefore, different search algorithms must be used.

The discrete search space algorithms are divided into local search and global search. The distinction being based on whether any given model is compared only with model(s) similar to it or to model(s) that may be very different. Global search algorithms tend to be much more robust to local minima. The local search algorithms include:

- Hill climbing [4]
- Tabu search [6]

In the hill climbing algorithm, the assumption is that if a feature is found to be valuable in one model, it is valuable in all other models and does not need to be tested again, that is, the convexity assumption. For example, if weight is found to be a predictor of volume in a one compartment model, the hill climbing algorithm assumes that weight will be similarly predictive of volume in a two compartment model. The hill climbing algorithm is a type of 'greedy algorithm' [7]. The definition of a greedy algorithm is 'An algorithm that always takes the best immediate or local solution while finding an answer'.<sup>4</sup> Greedy algorithms proceed rapidly to the nearest local optimum. Locating the global optimum with a greedy algorithm depends entirely on starting the search within some local area that it is convex to the global minimum, that is there are no 'ridges' between the initial model and the global minimum, which would prevent the search from proceeding in that direction. If this is not the case, greedy algorithms may find less-than-optimal solutions. The hill climbing algorithm is the most efficient search algorithm. It will arrive at the final model with the fewest number of evaluations because of the assumption that each hypothesis need only be tested a single time. The computational time required for a hill climbing search increases only linearly with the size of the search space.

<sup>4</sup><http://xlinux.nist.gov/dads/HTML/greedyalgo.html>, retrieved 29 January 2013.

The list of global search algorithms commonly used to search discrete space includes:

- Exhaustive search
- Simulated annealing [8]
- Particle swarm optimization [9, 10]
- Genetic algorithm (GA) [11]

In an exhaustive search, every possible combination of features is tested. The assumption is that testing a feature in one model is completely uninformative about the value of that feature in any other model. For example, finding that weight is a predictor of volume in a one compartment model provides no information about whether it will be a predictor in a two compartment model. An exhaustive search is a completely robust search and is guaranteed to find the best solution. However, the exhaustive search is limited by computational time which increases exponentially with the size of the search space.

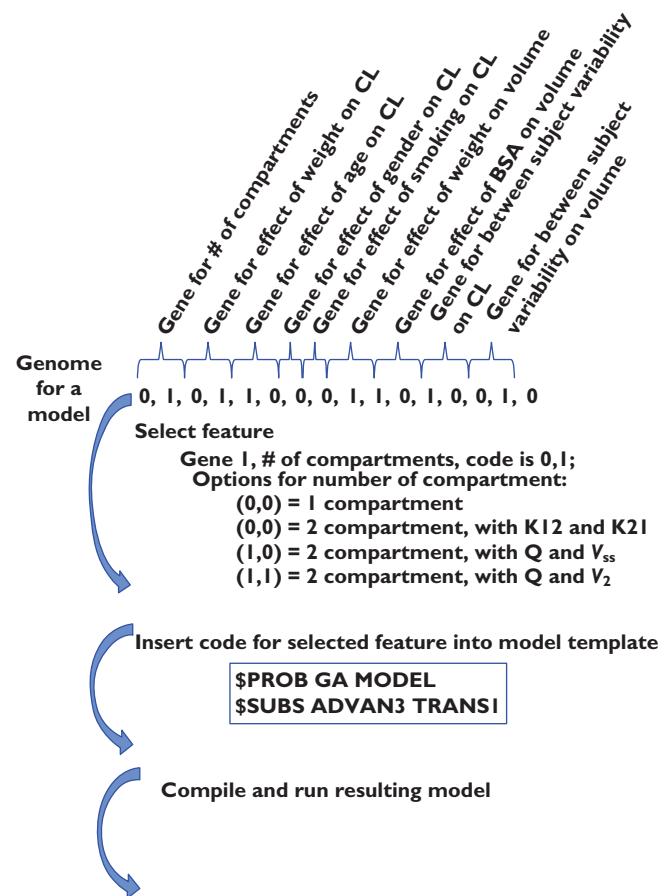
Other global search methods (e.g. simulated annealing, GA and particle swarm optimization) fall between these two extremes of hill climbing and exhaustive search in terms of the assumptions, the robustness of the search, and the computational burden. For example, if a GA finds weight to be a useful predictor of volume in one model this suggests, but does not guarantee, that weight is predictive of volume in other models. This hypothesis must be tested in many (usually thousands) of other models before being accepted or finally rejected. Among the global search algorithms, the GA was chosen as an approach for model selection in population PK/PD model building. The reasons for this selection include

- A clear method for implementing creation of NONMEM control files from the algorithm
- Easily parallelized
- The underlying algorithm is already understood by biologists
- Well documented, widely used, well researched algorithm

## Genetic algorithm

GA is an attempt to reproduce the naturally occurring processes of evolution and survival of the fittest to find a near optimal solution [11]. To implement a GA, the model is coded as a binary string. This coding for an example population pharmacokinetic model is depicted in Figure 3. In the case of population pharmacokinetics, the search space consists of all the candidate model features that would be considered feasible. These might include:

- Number of compartments (the first 'gene', two bits in Figure 3)



**Figure 3**

Coding of model features and translation into a model. If only two options are examined for a feature (e.g. the effect of gender on clearance) only 1 bit will be needed for that gene. If more than two options are examined (e.g. 4 for the basic structure, number of compartments) more than 1 bit is required for that gene. The final genome for each model is constructed by concatenating all the genes together into a bit string

- Covariates and different forms of covariate relationships (e.g. exponential, linear or power functions, the remaining 'genes', one or two bits in Figure 3)
- Different pharmacodynamic models (e.g. indirect response models)
- Variance terms including within subject variability, residual variance, and between subject variability
- Covariance between variability terms
- Mixture models
- Different initial estimates for parameters

If there are two possible values for a candidate model feature (e.g. presence or absence of an effect of gender on clearance with one set of initial estimate(s)), only a single bit is needed. There are many cases when there are more than two options. For example for a covariate relationship there may be:

**Table 1**

Code of model features and corresponding NONMEM/NMTRAN code

| Model feature                 | Feature options  | Bit string code | NONMEM code        |
|-------------------------------|--|-----------------|--------------------|
| Number of compartments        | One compartment  | 0.0             | 'ADVAN1'           |
|                               | Two compartment, parameterized as K21 and K12  | 0.1             | 'ADVAN3 TRANS1'    |
|                               | Two compartment, parameterized as intercompartmental clearance and steady-state volume of distribution | 1.0             | 'ADVAN3 TRANS3'    |
|                               | Two compartment, parameterized as intercompartmental clearance and peripheral volume                   | 1.1             | 'ADVAN3 TRANS4'    |
| Effect of weight on clearance | No effect  | 0.0             | ''                 |
|                               | Linear effect  | 0.1             | '+THETA() * WT'    |
|                               | Power model effect   | 1.0             | '*WT**THETA()'     |
|                               | Exponential effect   | 1.1             | '*EXP(THETA())*WT' |

- no covariate relationship
- a linear covariate relationship
- an exponential relationship
- a power relationship
- different initial estimates for any of the above relationships. (The inclusion of different initial estimates for parameters in the search space is an important feature of a global search algorithm because it can be very difficult to determine good initial parameter estimates for complicated pharmacokinetic and especially pharmacodynamic models. A global search of initial estimates for those parameters effectively addresses this issue.)

In these cases more than one bit will be needed. The number of bits required for a set of candidate features will be  $N\text{bits} \geq \ln N / \ln(2)$  where  $N\text{bits}$  is the number of bits required and  $N$  is the number of options. If there are five options, three bits will be needed. Table 1 gives examples of how candidate model features might be coded into a bit string and the corresponding NONMEM/NMTRAN code for those features. Once the search space is defined, a 'population' of models is created by randomizing the bits in the each genome in the population to 0 s and 1 s. Each bit string is then decoded into the corresponding NONMEM control file that represents each model, creating several hundred models with randomly selected features. These models are then run in NONMEM.

For a simple GA, two operations, crossover and mutation, are applied to these parent candidate models to create the next generation of candidate models [10]. The crossover operator creates two new candidate models from two parent models by randomly selecting a location on the bit string of the parents (same location on both parents) and swapping all of the bits after that location between the two parents to create two new candidates. As described in the next section, the selection of a parent candidate from the pool of possible candidates is based on its NONMEM output where candidate models that are more desirable have a higher likelihood of being selected for crossover. The mutation operator randomly changes a

bit value in a candidate model (i.e. 0 become 1 and vice versa).

#### Fitness

Much like the goodness of fit metric drives the parameter optimization in non-linear regression, an objective measure of model 'goodness' drives the search for the optimal model structure in global search algorithms. In GA this measure of model 'goodness' is called the fitness, from the same term used in population ecology [12] to describe how well an organism is adapted to the environment. More specifically, in population ecology the fitness is defined as the number of copies of an organism's genome that are expected to contribute to the next generation (and to subsequent generations). A fitness greater than 1 suggests that genetic material of that individual will tend to increase in frequency in the population while a fitness less than 1 suggests that genetic material of that individual will eventually be eliminated from the population.

In using a GA to search for the optimal model, the organisms are individual models and the algorithm works with a 'population' of individual models. Each organism (model) will have some objective measure of fitness (a function of overall model goodness) that will drive the number of copies of that model's genome that are contributed to the next generation of models. In GA the fitness function is problem specific. In the case of mixed effects models, there are a number of common metrics used to measure 'goodness' of a model. These include:

- Goodness of fit measures (usually  $-2\text{LL}$ )
- Parsimony, measured as the number of estimated fixed (theta) and random (omega and sigma) effect parameters
- Other 'quality of solution' metrics such as successful convergence, positive definite covariance matrix (e.g. successful covariance step in NONMEM) lack of estimation correlations between parameters and suitable condition number (ratio of largest to smallest eigenvalues)
- Cross validation  $-2\text{LL}$  [13] [14],
- Normalized prediction distribution errors (NDPE) global  $P$  value [15]

- Lack of correlation between covariates that are predictive of the same parameter
- Clinical significance of covariate effects

## Single objective GA

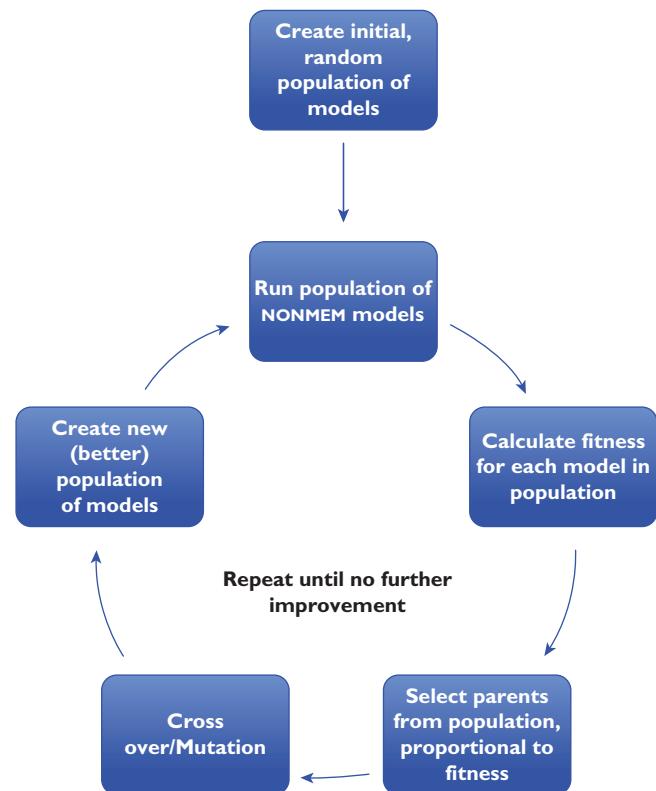
In the simplest implementation of GA, these measures of model goodness can be combined into a single value for fitness. The objective function from NONMEM ( $-2\text{II}$ ) serves as the basis for the fitness, with user defined penalties added to this for the other desired qualities of a model. For example, if the objective of the modelling exercise was simulation, rather than hypothesis testing, the penalty for additional estimated parameters (the parsimony penalty) might be small, perhaps 2 points as used in the Akaike information criteria [16]. Similarly, if the user is not concerned about having a successful covariance step, the penalty for this would be set to 0 or, in the other extreme, if there are scenarios that are not biologically plausible (e.g. a certain combination of covariates), these can simply not be included in the search space.

Figure 4 shows the overall process. The algorithm is initiated by randomly creating an initial population of models. Typically, many of these initial models will be very poor, often resulting in numerical problems, and may not converge successfully. The fitness for each model in the population of model is calculated. 'Parent' models can then be randomly selected with replacement from the population, proportional to the model fitness. The random selection with replacement is essential, as this permits especially fit (good) models to enter into the next generation multiple times, and the low fitness models to gradually be removed from the population. The 'parent' models are then paired off. Crossover between the parents and mutations are then applied, resulting in two new models for the next generation. In this way, the better models enter into subsequent generations with a higher frequency, and are recombined, and mutated to create potentially still better models.

For a search of model for NONMEM, the single objective fitness function is defined as:

$$\begin{aligned} \text{Fitness} = & -2\text{II} + N_\theta P_\theta + N_\omega P_\omega + N_\sigma P_\sigma + P_{\text{Convergence}} + P_{\text{Covariance}} \\ & + P_{\text{Correlation}} + P_{\text{Condition}\#} + P_{\text{NDPE}} + P_{\text{Parameter}} + P_{\text{Significance}} \end{aligned}$$

where  $-2\text{II}$  is the objective function from NONMEM (either from a single minimization run or cross validation),  $N_\theta$  is the number of estimated theta elements,  $P_\theta$  is the penalty for each estimated Theta element,  $N_\omega$  is the number of estimated Omega elements,  $P_\omega$  is the penalty for each estimated omega element,  $N_\sigma$  is the number of estimated sigma elements,  $P_\sigma$  is the penalty for each estimated Sigma elements,  $P_{\text{Convergence}}$  is the penalty applied if the model fails



**Figure 4**

Simple genetic algorithm. The algorithm is initialized with a random population. 'Parents' for the next generation are selected (with replacement) for the next generation proportional to the user defined 'fitness' of the model. These 'parent' models are then paired off and undergo crossover and mutation to form the next generation of models

to converge,  $P_{\text{Covariance}}$  is the penalty applied if the model fails the covariance step,  $P_{\text{Correlation}}$  is the penalty applied based on the diagonal elements of the correlation matrix,  $P_{\text{Condition}\#}$  is the penalty applied based on the condition number,  $P_{\text{NDPE}}$  is a penalty dependent on the global  $P$  value of NPDE,  $P_{\text{Parameter}}$  is a penalty dependent on the correlation coefficients of all covariates predicted a given parameter and  $P_{\text{Significance}}$  is a penalty applied for any covariate effects that fail a test of clinical significance.

All penalties except  $-2\text{II}$  are optional and the specific values of penalties are defined by the user. For example,  $P_{\text{Condition}\#}$  can be set to 100 if the condition number exceeds 1000; but the penalty and form of the condition number threshold are adjustable and are set by the user.

### Niching

A common characteristic of a simple GA is too rapid convergence. This limits the ability of the GA to explore further the global search space because of the lack of diversity in the population of models. This diversity serves as a substrate for recombination. Diversity in the population of

models is maintained by a technique called niching. In niching a penalty is applied to a group of models that are similar. A niche is then defined as the subset of the population of models that differ at less than a specified number of bits, called the niche radius. The user selects the number of niches, typically about four. The niches are determined by selecting the best model in the population as the first niche. All models that are within that radius are assigned to the first niche. Then the next best model that is not currently in a niche is selected, and all models that are within the radius are assigned to that niche. The penalty is applied to all the models in the niche, and the size of the penalty is determined such that the fitness of the best model in the niche remains better than the fitness of all models that are not in the niche. The addition of niche penalties permits the gradual elimination of the least fit models, preservation of many moderately fit models and prevents the best fitting models from rapidly dominating the population.

### *Elitism*

Because parent models are selected randomly, with a probability of selection that depends on their fitness, it is possible that the best model may actually not be chosen as a parent for the next generation. This is addressed by simply insuring that the best model will be preserved intact (without crossover or mutation) into the next generation, a method known as elitism.

### *Hybrid GA*

GA has been shown to be effective at finding generally good models. However, finding the last few changes that result in a truly optimal model can be difficult for GA. This issue is addressed by using a combination of the GA global search algorithms and a local search algorithm (see Figure 5). In this hybrid approach, GA is run for a number of (e.g. 3–10) generations after which a local search is performed, starting with the best model in each niche. The local search is similar to the traditional manual model building. A new population of models is created by systematically reversing each bit in the genome of the best model in each niche. That is, if a genome is 100 bits long, 100 models will be created, each differing from the initial model at 1 bit. If four niches are used, 400 models will result, 100 from the best model in each niche. Each model is run and the fitness is calculated.

If one or two models from the hill climbing step have a better fitness than the base models, the best model is substituted for the base model and another step in the local hill climbing search is performed. If more than two models are better than the base model, a local exhaustive search is done. For the local exhaustive search, all bit changes that result in a better model (up to 6) are examined in all combinations. If there are 6 bit changes that result in a better model, 64 ( $2^6$ ) models will result. If the local exhaustive search is performed, the best model from

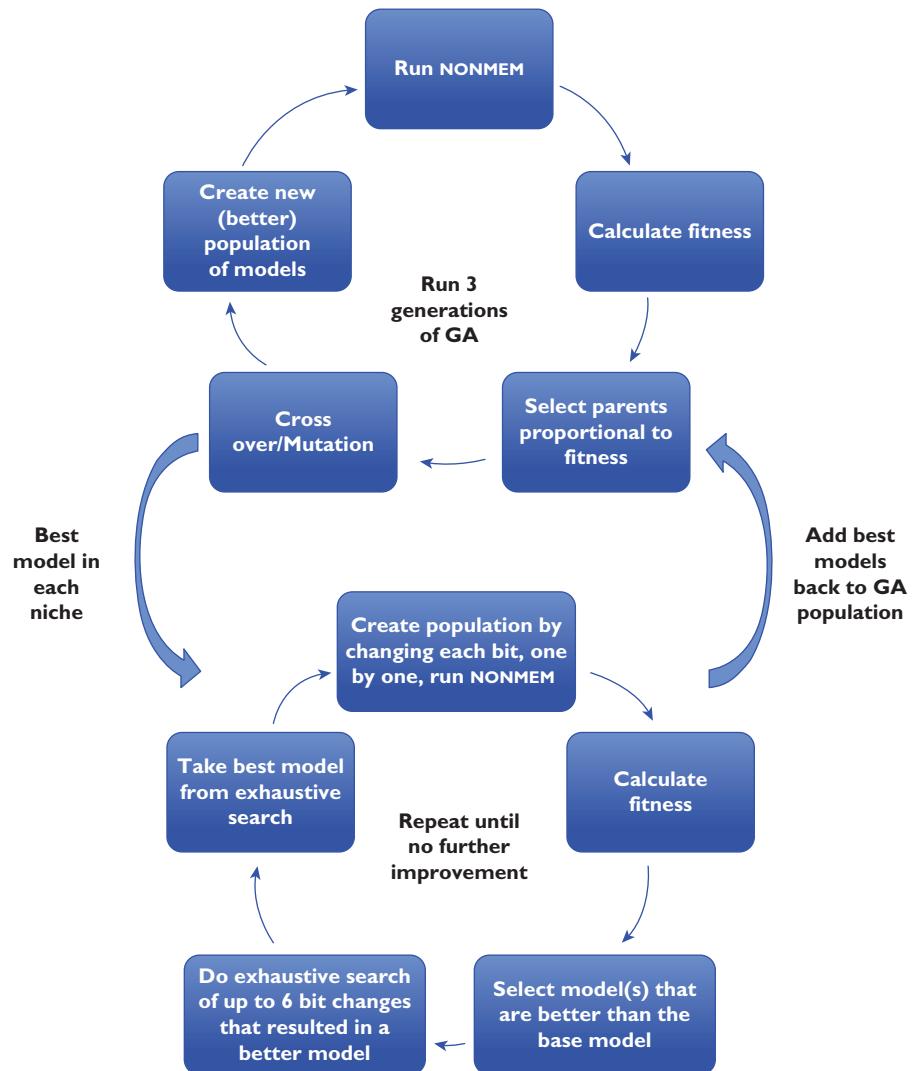
that search is then used for the next local hill climbing step. This process is repeated until no further improvement is seen. Once no further improvement is seen, the best models from the hill climbing/local exhaustive search are added to the existing population and the search returns to GA for another iteration of the global search/local search hybrid approach. In this way, a number of distinct, but fairly good regions for the search space are explored in detail every three generations. This is in contrast to the traditional hill climbing approach, which starts at a single point that is chosen for convenience (e.g. one compartment, no covariates), rather than being an particularly good model. The initiation of the hill climbing search at multiple, reasonably good points in the search space greatly improves the likelihood of finding the global minimum.

## **Implementation of single objective, hybrid GA**

The model building performance of the single objective hybrid GA was compared with traditional methods in a blinded, retrospective crossover study [17]. A summary of the results is given in Table 2. The primary criterion for evaluation of the models was the Akaike information criteria (AIC) although neither the traditional model building nor the GA method specifically targeted AIC. All analyses were designed to find 'good' models by traditional objective criteria. These results suggest that the hybrid GA algorithm was in all cases at least as effective at finding models with a lower AIC.

In addition, to the AIC results, the traditional model building method was not able to find a model that converged in four of the seven analyses without fixing the absorption rate constant. With the value for absorption rate fixed to a literature value, two of the analyses did not result in a successful covariance step. In contrast, the GA was able to select a model that had a successful convergence and successful covariance step without fixing any parameters in all cases.

In general, the single objective hybrid GA and traditional model building approaches found similar models structures and the GA approach tended to include more covariates than the traditional approach. The compartment structure was the same for six of seven analyses with one exception (risperidone) where GA found a two compartment model while the traditional approach found a one compartment model. The AIC for the GA risperidone model was 278.1 lower (better) than the risperidone model found by the traditional approach. For covariate inclusion, the GA approach found a total of 23 covariates among the seven analyses while the traditional approach found a total of 13. Seven of these covariates were common to both approaches.

**Figure 5**

Hybrid global search algorithm, combining GA, hill climbing and exhaustive search. This allows detailed exploration of a number of local regions in the search space identified by GA. This addition greatly improves the search algorithm, alternating between an efficient algorithm (hill climbing) and a completely robust algorithm (exhaustive search)

**Table 2**

Summary of results from comparison of traditional model selection vs. single objective hybrid genetic algorithm (SOHGA). A lower value for AIC suggests a more informative model. In all cases the SOHGA analysis resulted in a lower AIC than the traditional analysis

| Drug         | AIC by traditional method | AIC by SOHGA | Change in AIC (SOHGA – traditional) |
|--------------|---------------------------|--------------|-------------------------------------|
| Citalopram   | 5391.9                    | 5369.6       | -22.3                               |
| DMAG         | 9871.7                    | 9849.4       | -22.3                               |
| Escitalopram | 2737.7                    | 2737.6       | -0.1                                |
| Olanzapine   | 10365.8                   | 9895.3       | -470.5                              |
| Pephenazine  | 560.7                     | 555.9        | -4.8                                |
| Risperidone  | 5131.1                    | 4853.0       | -278.1                              |
| Ziprasidone  | 4763.2                    | 4758.7       | -4.5                                |

## Multi-objective GA

In PK/PD model building, some of the desired model characteristics (e.g. parsimony) may conflict with other desirable characteristics (e.g. minimized log-likelihood). The single objective optimization attempts to balance these individual objectives using penalty functions to weight individual objectives. A significant disadvantage to this approach is that the final model selected will depend on the weighting scheme for which the selection is *ad hoc* and subjective. Rather than combining individual objectives into a single composite objective function, multi-objective GA (MOGA) compares candidate models by each of the individual objectives and searches for non-dominated models [18].

One model dominates another when it is not worse on all objectives and better on at least one. For example, for model A to dominate model B, model A must have a  $-2\text{ll}$  at least as low, the same number or fewer parameters, the same or better quality of the solution etc. and be better on at least one criteria. For one model to dominate another means that, by objective criteria, there is no justification for not preferring that model, it is at least as good in all ways and better in at least one. Because of the tradeoffs between criteria, it is unrealistic to search for dominant models, there will never be a single model that dominates all models in the search space. For example, a model that has a  $-2\text{ll}$  value lower than all the other models will also not have fewer parameters.

Rather than searching for dominant models, we search for models that are not dominated, that is, there is no model that by objective measures is unquestionably better than this model. There will usually be many such models. A candidate model in which there is no other candidate model that is better or equal in all criteria is called Pareto optimal and the set of Pareto optimal candidates comprise the Pareto optimal set. An advantage of the Pareto optimal set is it allows the user to examine all models that are, by objective criteria, unsurpassed (i.e. non-dominated), and to select from among those on subjective criteria (biological plausibility, diagnostic plots). In PK/PD model building, objectives that can be used in the multi-objective optimization include the NONMEM objective function, number of estimated parameters, convergence, covariance, and correlation test results and global adjusted  $P$  value from NPDE.

## Implementation of multi-objective, hybrid GA

The multi-objective, hybrid GA has been implemented for the building the PK model for three different compounds and the resulting models compared with the single objective, hybrid GA and traditional methods for three compounds in a retrospective crossover study [19]. As with the traditional model building approach and the single objective GA approach, the multi-objective method could vary the ADVAN/TRANS structure of the model, the inclusion of inter-occasion variability and block structure, the inclusion of covariates the functional form and the form of the residual variability. In the multi-objective GA, candidates were evaluated along four dimensions: NONMEM objective function, number of estimated parameters, sum of the convergence, covariance step, and correlation test and global adjusted  $P$  value from NPDE. For all three compounds, the multi-objective GA found models that passed the convergence, covariance and correlation tests and with equal (1 compound) or lower (2 compounds) NONMEM objective function values than with the traditional model building approach. Compared with the single objective, hybrid GA,

the multi-objective hybrid GA Pareto optimal candidate with the same number of model parameters had similar NONMEM objective function values (within 5 points) for all three compounds.

## Finding solutions vs. generating understanding

GAs have been applied to many optimization problems including protein folding [20, 21], facial recognition,<sup>5</sup> gene expression analysis [22], and integrated circuit design [23]. In pharmacokinetics, GAs are used in complex systems for parameter estimation when greedy algorithms are not sufficiently robust [24]. However, most of these examples are simply search algorithms that are not intended to create understanding. Schmidt & Lipson [25] observed 'Despite the prevalence of computing power, the process of finding natural laws and their corresponding equations has resisted automation. A key challenge to finding analytic relations automatically is defining algorithmically what makes a correlation in observed data important and insightful'. Schmidt & Lipson went on to describe a paradigm in which natural laws could be discovered with a combination of automated search and human insight. The approach consisted of using multi-objective GA to analyze the motion of several physical systems including a single pendulum, double pendulum and spring mounted single and double masses on an air track. Data were collected on the motion of these systems. Multi-objective genetic programming was then used to assemble equations using only fundamental mathematical symbols (+, -, \*, /, cosine, sine, exponentiation) without any human intervention and to find the parameters of these equations. Multi-objective genetic programming was able to find the correct analytic solution, as systems of differential equations in all cases. In the case of the double pendulum (the most complex system), the algorithm found a Pareto set with 10 models that best described the data empirically. These 10 models were then presented to the user to evaluate for theoretical plausibility. The analytically correct equations were found among these 10 models. This approach (human selection and interpretation from among multiple models) has also proven useful in making predictions for weather [26] and has been demonstrated by Chaturvedula *et al.* for population pharmacokinetic modelling [27].

## Mode of use

It is the exception when data analysis alone is able to generate insight about natural processes [28]. The traditional approach to creating understanding is to use theoretical knowledge to create candidate models and then to

<sup>5</sup><http://www.evofit.co.uk/>, retrieved 29 January 2013.

evaluate and compare how consistent those models are with data. Despite the findings of Schmidt & Lipson [25] in a physics experiment, it is unlikely that understanding and insight can be created from pharmacokinetics data sets without human intervention. The paradigm of humans creating hypotheses based on understanding of pharmacology, then testing those hypotheses with data does not change when moving from a local search algorithm to a global search algorithm.

The incorporation of knowledge of the pharmacology of a drug into either a local or a global search algorithm starts with defining the search space. Specifically, if an effect has no possible basis in biology, it should not be examined in a traditional local search algorithm or a global search algorithm. Conversely, if a feature is well established to have an effect (e.g. creatinine clearance effect on amino glycoside clearance), potentially that effect need not be tested at all, but simply included in the model. Later in the analysis, if inclusion of a candidate feature that is biologically plausible (e.g. weight as a predictor of volume of distribution) results in infeasible parameter estimates (e.g. a negative relationship between weight and volume of distribution), the causes of such a relationship must be examined and (probably) the model modified so that the parameter value is biologically plausible.

In addition to the initial search space definition, there are opportunities to interact with the global search algorithm. The most obvious is simply if the search is consistently resulting in implausible models, the search should be stopped, the implausible models examined and modified appropriately. These modifications can then be incorporated in a new search space and the search begun again. Alternatively, if the data clearly show results inconsistent with the theoretical model, perhaps the theoretical model should be re-evaluated.

In addition, the traditional model building algorithm includes a backward elimination step. In this approach, the statistical threshold for a model feature being accepted is relatively low (often  $P < 0.05$ ). The low threshold for inclusion results in a relatively large (many features) model. In order to correct for multiple comparison, typically, the features in this model are individually removed and retested at a higher threshold (often  $P < 0.01$ ), an algorithm known as backward elimination (last step in Figure 1). Global search is well suited to this approach as first described by Chaturvedula *et al.* [27]. This approach might be called GA/backward elimination (GABE), where the traditional forward addition model building stage is replaced by the global search. As in the traditional forward addition/backward elimination, the threshold for inclusion of an effect in the model should be relatively low for the GA model selection. This will result in a larger model, some features of which may not meet statistical thresholds, be clinically important or be considered important in the model for other reasons. These features can be removed in the backward elimination step of the GABE algorithm, a

process that should include human evaluation for biological plausibility.

## Practical issues

The most important practical issue with any automated algorithm is the temptation to accept the results blindly. Global search methods do not substitute for the thoughtful examination of the data, and the incorporation of pharmacologic understanding into model selection. They simply automate the actual search part, not the intellectual part. The available software for global search in pharmacokinetics makes available the results with basic diagnostics for all models that are run presented in a large spreadsheet. Any of these models can be examined, modified and rerun prior to selecting the 'final' model.

Related to the temptation to accept blindly the algorithm results is the range of modelling exercises that can be addressed by automated methods. Automated global search algorithms are best suited when most hypotheses can be stated initially. However, every automated analysis should include examination of the results for sources of bias, and consideration of how to expand the search space to include any new hypotheses from that examination. Sometimes, hypothesis creation (especially in the case of pharmacodynamic models) is nearly iterative. A model is constructed and run. Once the model is run, diagnostics demonstrate sources of bias. Biologically sound hypotheses to explain these biases are generated and a new model is created and run, at which time diagnostics are used to generate new hypotheses. It would be difficult to use automated search algorithms in these cases, since relatively few hypotheses are available at the outset with which to construct a search space.

The second issue with global search algorithms is the high computational load. This should be examined in comparison with the high modeler effort required to actually type, run and examine the output from the models. Two thousand hours of computer time is much less expensive than 200 hours of modeler effort. The current price for Linux CPU time on Amazon compute cloud is \$0.06 h<sup>-1</sup>.<sup>6</sup> It should be noted that methods exist to automated stepwise modeling for covariates (SCM in Perl speaks NONMEM<sup>7</sup>).

Another significant issue with global search algorithms is the likelihood that examining thousands of models increases the risk of an inflated alpha error. It remains unclear how much of an issue this is, but initial results suggest that this 'over modelling' problem can be managed with appropriate parsimony penalties, and penalties for clinical significance and correlated covariates on a single parameter [29]. Cross validation is another approach

<sup>6</sup><http://physicsworld.com/cws/article/news/2009/apr/02/algorithm-discovers-physical-laws>, retrieved 29 January 2013.

<sup>7</sup><http://aws.amazon.com/ec2/pricing/>, retrieved 29 January 2013.

to managing inflated alpha error rates [13, 30] that seems especially promising and is available in the current GA application.

The application of a population PK/PD model to the drug development process has been limited by the time and expense of the analyses. Automated global search methods promise to decrease both dramatically, making it realistic to have population pharmacokinetic models available for decision making while the decisions are still being made. A typical automated analysis includes about 12 000 unique models, many more than a traditional analysis. If the run time for a model is 10 to 20 min, and 48 cores are available (eight six core machines), this suggests time for the analysis of 2 to 4 days, rather than the traditional several weeks. Other computational models where hundreds of CPUs were readily available (e.g. cloud computing) could reduce this even further.

## Conclusions

The approach to model selection in population PK/PD has changed little in the past 30 years. This approach arose out of a combination of statistical theory (forward addition/backward elimination in linear regression) and practical considerations (that the inclusion of random effects dramatically increased run time and cost). Of these two sources of the algorithm, the basis in forward addition/backward elimination is limited by the failure of the assumption of convexity, and the run time issue with variance terms has yielded to a dramatic increase in computer speed in the last 30 years. Global search algorithms not only have proven to be more robust, but, in some cases, when combined with human input to be capable of uncovering basic principles of nature [25].<sup>8</sup>

## Competing Interests

All authors have completed the Unified Competing Interest form at [http://www.icmje.org/coi\\_disclosure.pdf](http://www.icmje.org/coi_disclosure.pdf) (available on request from the corresponding author) and declare no support from any organization for the submitted work, no financial relationships with any organizations that might have an interest in the submitted work in the previous 3 years and no other relationships or activities that could appear to have influenced the submitted work.

## REFERENCES

- 1** Hocking R. The analysis and selection of variables in linear regression. *Biometrics* 1976; 32: 1–49.

<sup>8</sup><http://www.uppsala-pharmacometrics.com/software.html>, Retrieved 18 May 2013.

- 2** Hosmer D, Lemeshow S. *Applied Logistic Regression*, Second edn. New York, NY: John Wiley & Sons, Inc., 2000.
- 3** Cambini A, Martein L. *Generalized Convexity and Optimization: Theory and Applications*. Berlin Heidelberg: Springer-Verlag, 2009; 17.
- 4** Cormen TH, Leiserson CE, Rivest RL, Stein C. *Introduction to Algorithms*. Chapter 17 'Greedy Algorithms'. Cambridge, MA: MIT Press, 1990; 329.
- 5** Wade JR, Beal SL, Sambol NC. Interaction between structural, statistical, and covariate models in population pharmacokinetic analysis. *J Pharmacokinet Biopharm* 1994; 22: 165–77.
- 6** Glover F. Tabu search: a tutorial. 1990. *Interface*.
- 7** Bang-Jensen J, Gutin G, Yeo A. When the greedy algorithm fails. *Discrete Optim* 2004; 1: 121–12.
- 8** Press WH, Teukolsky SSA, Vetterling WT, Flannery BP. *Numerical Recipes in C*. New York: Cambridge University Press, 1992; 394.
- 9** Kennedy J, Eberhart R. Particle swarm optimization. *Proc IEEE Int Conf Neural Networks* 1995; IV: 1942–8.
- 10** Kim S, Li L. A novel global search algorithm for nonlinear mixed-effects models using particle swarm optimization. *J Pharmacokinet Pharmacodyn* 2011; 38: 471–95.
- 11** Goldberg D. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Boston, MA: Addison-Wesley Professional, 1989.
- 12** Crow JF, Kimura M. *An Introduction to Population Genetics Theory*. New York: Harper and Row, 1970.
- 13** Katsume T, Khandelwal A, Hooker AC, Jonsson EN, Karlsson MO. Characterization of stepwise covariate model building combined with cross-validation. 2012. PAGE meeting 2012 Venice Italy.
- 14** Gelman A, Carlin JB, Stern HS, Rubin DB. *Bayesian Data Analysis*. New York: Chapman and Hall, 1995.
- 15** Mentré F, Escolano S. Prediction discrepancies for the evaluation of nonlinear mixed-effects models. *J Pharmacokinet Biopharm* 2006; 33: 345–67.
- 16** Akaike H. A new look at the statistical model identification. *IEEE Trans Automatic Control* 1974; 19: 716–23.
- 17** Bies RR, Muldoon MF, Pollock BG, Manuck S, Smith G, Sale ME. A genetic algorithm-based, hybrid machine learning approach to model selection. *J Pharmacokinet Pharmacodyn* 2006; 33: 195–221.
- 18** Konak A, Coit DW, Smith AE. Multi-objective optimization using genetic algorithms: a tutorial. *Reliabil Eng Syst Saf* 2006; 91: 992–1007.
- 19** Sherer EA, Sale M, Manuck S, Muldoon M, Pollock BG, Bies RR. Three case studies of pharmacokinetic model building using a multi-objective genetic algorithm. The Population Approach Group in Europe Annual Meeting, June 5–8, 2012, Venice, Italy. 2012.
- 20** Willett P. Genetic algorithms in molecular recognition and design. *Trends Biotechnol* 1995; 13: 516–21.

- 21** Wong K, Leung K, Wong M. Protein structure prediction on a lattice model via multimodal optimization techniques. GECCO 2010; 2010: 155–62.
- 22** To CC, Vohradsky J. A parallel genetic algorithm for single class pattern classification and its application for gene expression profiling in *Streptomyces coelicolor*. BMC Genomics 2007; 8: 49.
- 23** Dick RP, Jha NK. MOGAC: a multiobjective genetic algorithm for hardware-software cosynthesis of distributed embedded systems. IEEE Trans Comput-Aided Design Integr Circuits Syst 1998; 17: 1–13.
- 24** Jamei M, Yang J, Yeo K, Tucker GT, Rostami-Hodjegan A. Genetic algorithms and their applications in PK/PD data analysis. 1st PharmSciFair. Nice, France, 12th–17th. 2005.
- 25** Schmidt M, Lipson H. Distilling free-form natural laws from experimental data. Science 2009; 324: 81–5.
- 26** Tolman H. State of the art lecture. American conference on pharmacometrics. 2013.
- 27** Chaturvedula A, Lee H, Sale M. Automated model selection for describing simvastatin and amlodipine interaction using single objective hybrid genetic algorithm (SOHGA). American College of Clinical Pharmacy meeting. 2012. Available at <http://www.accp1.org/pageflip/files/images/pages/page46.swf> (last accessed 20 May 2013).
- 28** Wigner EP. The unreasonable effectiveness of mathematics in the natural sciences communications on pure and applied mathematics, XIII, 1960; 001–14.
- 29** Sale M. Relationship of model selection metrics to prediction accuracy. The Second Indiana Clinical and Translational Sciences Institute (CTSI) Symposium on Disease and Therapeutic Response Modeling, 2012 Indianapolis IN.
- 30** Sale ME, Bies RR. Relationship of model selection criteria to prediction accuracy. J Pharmacokinet Pharmacodyn 2013; 40: S26.