

*A hybrid genetic algorithm for NONMEM  
structural model optimization*

Mohamed Ismail, Robert R. Bies, Mark Sale

University at Buffalo

Nuventra

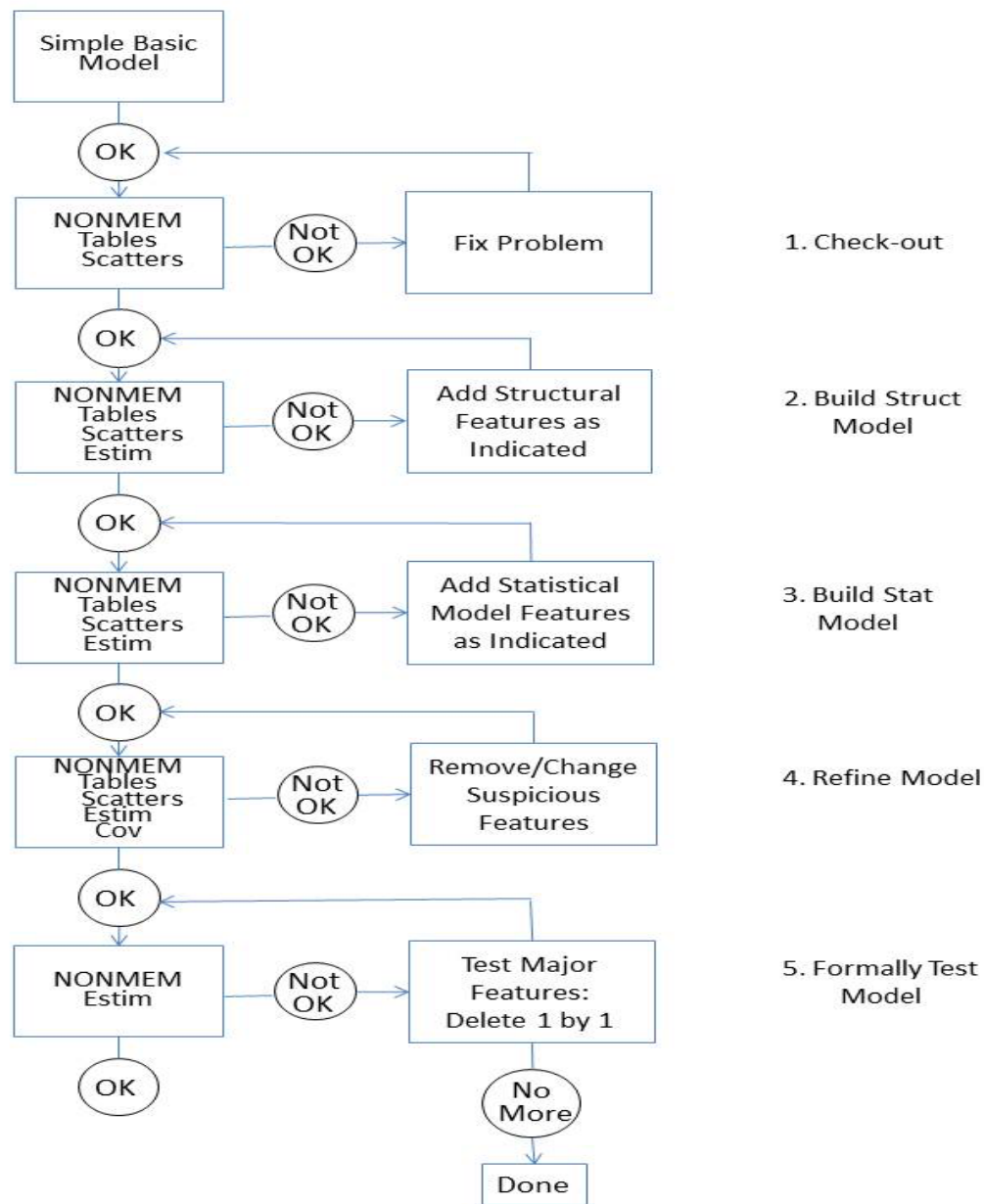


Figure 1. Diagram of model building algorithm from Volume 5 NONMEM manuals. Reproduced with permission from Icon PLC. In the original description of the algorithm, statistical features (variance terms) were added after the structure was final for practical reasons.

# Local search: “step-wise” regression

- Base (covariate free) model
  - Keep known physiology in mind
  - Compare compartment structures
    - Residual error structure to minimize systematic errors
    - Inter-individual variability where identifiable
  - Lag-time or mixture models if relevant
- Final model
  - Baseline structure
  - Single covariate forward addition
  - Single covariate backward elimination

# Local search: “step-wise” regression

- Base (covariate free) model
  - Keep known physiology in mind
  - Compare compartment structures
    - Residual error structure to minimize systematic errors
    - Inter-individual variability where identifiable
  - Lag-time or mixture models if relevant
- Final model
  - Baseline structure
  - Single covariate forward addition
  - Single covariate backward elimination

Wade JR, Beal SL, and Sambol NC.  
“Interaction between structural, statistical,  
and covariate model in population  
pharmacokinetic analysis”, J of  
Pharmacokinetics and Biopharmaceutics, 22:  
165-177, 1994.

# Genetic Algorithms

- What are they?
  - A means of evaluating factors in a model where more than one factor can be changed at a single step.
  - Partially automated to allow a more “complete” evaluation of the full grid search space for a particular candidate model.

# Genetic Algorithms

- Approach:
  - Replicate “survival of the fittest”
  - Evolutionary process is imposed on the selection and “survival” of the “best” model descriptions
  - Calculate an indicator of how “healthy” a particular individual model in the population is
  - Utilized in multiple fields e.g. placing cell phone towers, predicting stock performance etc.

# Genetic Algorithms

- “good” characteristics become more likely
- Efficient at finding “good” regions of solution space
- Slow to converge local “best”
- Adaptations
  - Elitism
    - Retain best candidate to next generation
  - Local search hybrid
    - Compare candidate with each model differing by 1 bit
    - Every 5 generations

# Genetic Algorithms

- Implementation in the context of population PK modeling (Bies and Sale 2006, JPP August, Sherer Sale and Bies 2012 JPP)
- Potential models are reduced to a bit-string (base-2 number assembly) that reflects the model “genetic” code
- Each model feature is coded as a base 2 number
  - If there are 2 options the values are 0 or 1 [(0) (1)], if more than two options then one has multiple bits eg. [(0 0), (0 1), (1 0), (1 1)]
- Features are strung together to produce aforementioned bit string
- Model can be reproduced based on the bit string that results

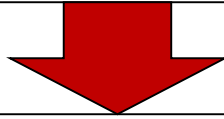


# Global optimization: genetic algorithm

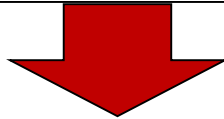
- Single-objective
  - Default composite fitness measure (initial implementation)
    - $-2 \times \log$ -likelihood
    - Penalty per model variable (**10 points**)
    - Penalties for failure to converge (400), covariance (400), and correlation (300)

# Model Selection

Compartment structure	Residual error	IIV on CL	Weight on CL	Weight on V
1 compartment <b>1 compartment w/ lag</b> 2 compartments 2 compartments w/ lag	Additive Proportional <b>Combined</b>	No relationship Additive Proportional <b>Exponential</b>	<b>No relationship</b> Additive Proportional Exponential Power-law	No relationship Additive <b>Proportional</b> Exponential Power-law



**NONMEM**



- Model evaluation criteria
  - $-2 \times \log$ -likelihood
  - Number of parameters
  - Diagnostic plots

# Basic genetic algorithm

Candidate models (N = 300 – 500)

Candidate 1.	
Compartment structure 1 compartment <b>1 compartment lag</b> 2 compartments 2 compartments lag	Residual error Additive Proportional <b>Combined</b>
Candidate 2.	
Compartment structure 1 compartment <b>1 compartment lag</b> 2 compartments 2 compartments lag	Residual error <b>Additive</b> Proportional Combined
Candidate 3.	
Compartment structure 1 compartment 1 compartment lag <b>2 compartments</b> 2 compartments lag	Residual error Additive <b>Proportional</b> Combined

# Basic genetic algorithm

Candidate models (N = 300 – 500)

Candidate 1. <b>Fitness = 1,000</b>	
Compartment structure 1 compartment <b>1 compartment lag</b> 2 compartments 2 compartments lag	Residual error Additive Proportional <b>Combined</b>
Candidate 2. <b>Fitness = 1,200</b>	
Compartment structure 1 compartment <b>1 compartment lag</b> 2 compartments 2 compartments lag	Residual error <b>Additive</b> Proportional Combined
Candidate 3. <b>Fitness = 1,050</b>	
Compartment structure 1 compartment 1 compartment lag <b>2 compartments</b> 2 compartments lag	Residual error Additive <b>Proportional</b> Combined

Evaluate fitness  
using NONMEM

# Basic genetic algorithm

Candidate models (N = 300 – 500)

<b>Candidate 1. Fitness = 1,000</b>	
Compartment structure 1 compartment <b>1 compartment lag</b> 2 compartments 2 compartments lag	Residual error Additive Proportional <b>Combined</b>
<b>Candidate 2. Fitness = 1,200</b>	
Compartment structure 1 compartment <b>1 compartment lag</b> 2 compartments 2 compartments lag	Residual error <b>Additive</b> Proportional Combined
<b>Candidate 3. Fitness = 1,050</b>	
Compartment structure 1 compartment 1 compartment lag <b>2 compartments</b> 2 compartments lag	Residual error Additive <b>Proportional</b> Combined

Evaluate fitness  
using NONMEM

**1 0**

**0 1**

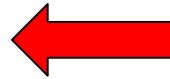
Binary representation of model decisions

# Basic genetic algorithm

Candidate 1. Fitness = 1,000	
Compartment structure	Residual error
1 compartment	Additive
<b>1 compartment lag</b>	Proportional
2 compartments	<b>Combined</b>
2 compartments lag	

**0 1**

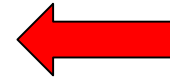
**1 0**



Candidate 3. Fitness = 1,050	
Compartment structure	Residual error
1 compartment	Additive
1 compartment lag	<b>Proportional</b>
<b>2 compartments</b>	Combined
2 compartments lag	

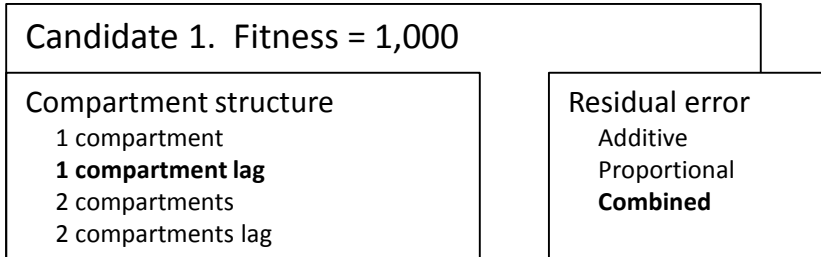
**1 0**

**0 1**



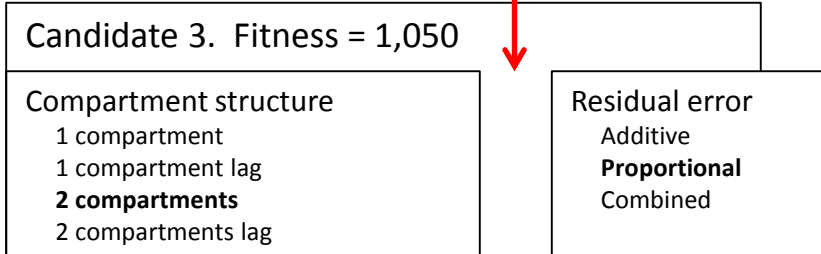
Reproduction:  
Randomly select two models from the candidate pool based on normalized fitness

# Basic genetic algorithm



**0 1**

**1 0**



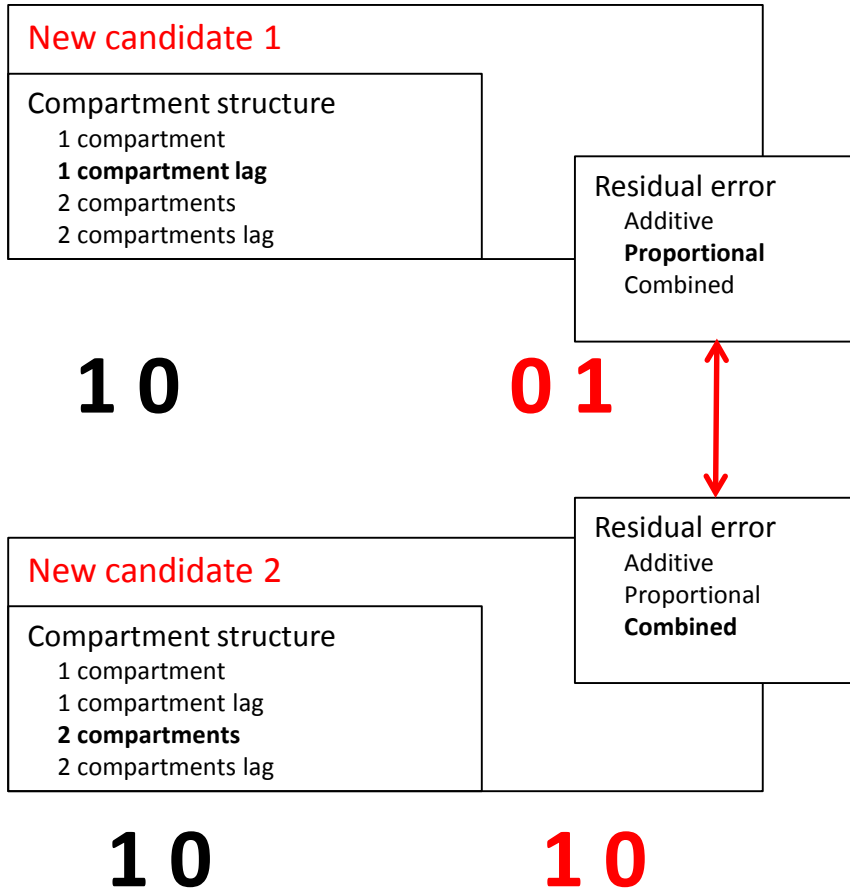
**1 0**

**0 1**

Crossover:

Randomly select a model location

# Basic genetic algorithm



## Crossover:

Randomly select a model location

Swap model information with probability  $P_{crossover}$



# Basic genetic algorithm

New candidate 1	
Compartment structure	Residual error
1 compartment	<b>Additive</b>
<b>1 compartment lag</b>	Proportional
2 compartments	Combined
2 compartments lag	

**1 0**

**0 0**

New candidate 2	
Compartment structure	Residual error
1 compartment	Additive
1 compartment lag	Proportional
<b>2 compartments</b>	<b>Combined</b>
2 compartments lag	

**1 0**

**1 0**

## Mutation:

Randomly select a model location

Change model information with probability  $P_{mutation}$

# Basic genetic algorithm

## New candidate models

New candidate 1.

Compartment structure

1 compartment  
**1 compartment lag**  
2 compartments  
2 compartments lag

Residual error

**Additive**  
Proportional  
Combined

New candidate 2.

Compartment structure

1 compartment  
1 compartment lag  
**2 compartments**  
2 compartments lag

Residual error

Additive  
Proportional  
**Combined**

Repeat reproduction,  
crossover, and mutation  
operations until a new  
candidate pool is created

Repeat process for desired  
number of 30-50  
generations

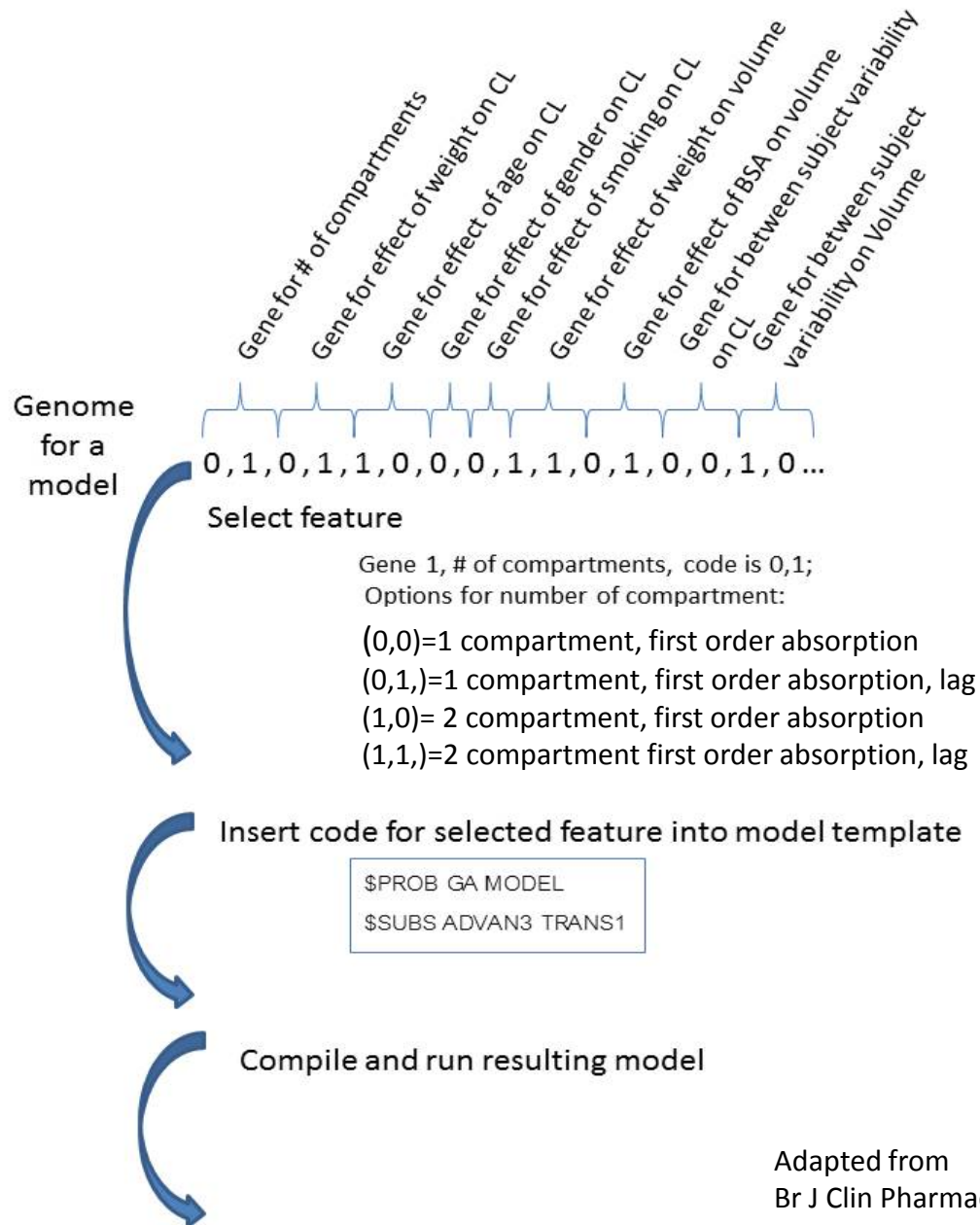


Figure 3. Coding of model features and translation into a model. If only two options are examined for a feature (e.g., the effect of Gender on Clearance) only 1 bit will be needed for that gene. If more than two options are examined (e.g., 4 for the basic structure, number of compartments) more than 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.

# Covariate Search Comparison

- Evaluation of performance of multiple methods
  - True model simulated with relatively dense sampling
  - Exponential relationship with BMI and CrCL on clearance
  - Exponential relationship BSA and Sex on volume
  - Compared:
    - Stepwise Covariate Modeling
    - LASSO (least absolute shrinkage and selection operator)
    - Single Objective Hybrid Genetic Algorithm

# Covariate Search Comparison

**Table 5** True and spurious covariate relationships identified in the simulated data by the automated stepwise covariate modeling, Lasso, and SOHGA approaches and the models fit characteristics

Method	"True" covariates		Spurious covariates		Objective function value
	Clearance	Volume of distribution	Clearance	Volume of distribution	
Original model	BMI, CRCL	BSA, Sex	–	–	6101.2
Stepwise covariate modeling (SCM): <i>p</i> value for inclusion, <i>p</i> value for elimination					
0.05, 0.05	BMI, CRCL	Sex	WT	HT, CV1	6085.9
0.05, 0.01	BMI, CRCL	Sex	–	HT, CV1	6091.1
0.10, 0.01	BMI, CRCL	Sex	–	HT, CV1	6091.1
Lasso model	BMI, CRCL	–	–	–	6254.2
Single-objective, hybrid genetic algorithm					
3.84 point penalty per parameter	BMI, CRCL	Sex	BSA	HT, CV1	6086.7
10 point penalty per parameter	BMI, CRCL	Sex	–	HT	6097.9

*BMI* body mass index, *BSA* body surface area, *CRCL* creatinine clearance, *CV1* unrelated covariate 1, *HT* height, *WT* weight

Single-objective, hybrid genetic  
algorithm (SOHGA)

vs.

step-wise approach

- Pharmacokinetic data for Risperidone
  - Identical model options / decisions
- Compare information criteria of final models
  - Compare model structures

<b>Compound</b>	<b>Administration method</b>	<b>Number of patients</b>	<b>Number of concentration measurements</b>
<b>CATIE</b>			
<b>Risperidone</b>	Oral	490	1,236

Compound	Administration method	Number of patients	Number of concentration measurements
<b>CATIE</b>			
<b>Risperidone</b>	Oral	490	1,236

	NONMEM model structures tested	First-order (FO) or first-order conditional (FOCE) estimation	Number of covariates collected
<b>Risperidone, oral</b>	ADVAN2, TRANS2 ADVAN4, TRANS4 (with 1, 2, or 3 clearance subpopulations)	FO	9



Compound	Administration method	Number of patients	Number of concentration measurements
<b>CATIE</b>			
<b>Risperidone</b>	Oral	490	1,236

	NONMEM model structures tested	First-order (FO) or first-order conditional (FOCE) estimation	Number of covariates collected
<b>Risperidone, oral</b>	ADVAN2, TRANS2 ADVAN4, TRANS4 (with 1, 2, or 3 clearance subpopulations)	FO	9

	Convergence		Covariance step (condition number)	
	Final step-wise model	Best SOHGA candidate	Final step-wise model	Best SOHGA candidate
<b>Risperidone, oral</b>	Required fixing $K_a$ early in model building process	Successful	Successful (60)	Successful ( $1.17 \times 10^6$ )

Compound	Administration method	Number of patients	Number of concentration measurements
<b>CATIE</b>			
<b>Risperidone</b>	Oral	490	1,236

	NONMEM model structures tested	First-order (FO) or first-order conditional (FOCE) estimation	Number of covariates collected
<b>Risperidone, oral</b>	ADVAN2, TRANS2 ADVAN4, TRANS4 (with 1, 2, or 3 clearance subpopulations)	FO	9

	Convergence		Covariance step (condition number)	
	Final step-wise model	Best SOHGA candidate	Final step-wise model	Best SOHGA candidate
<b>Risperidone, oral</b>	Required fixing $K_a$ early in model building process	Successful	Successful (60)	Successful ( $1.17 \times 10^6$ )

Compound	Final stepwise model	Best SOHGA candidate model	$AIC_{SOHGA} - AIC_{stepwise}$
<b>Risperidone, oral</b>	AIC = 5,131.1	AIC = 4,853.0	-278.1

# Model structure: SOHGA vs. step-wise

Compound	Final step-wise model	Best SOHGA candidate
Risperidone, oral	1 with 3 component mixture on CL	<b>2 with 2 component mixture on CL</b>

- Extra degree of freedom
  - Fix  $k_a$  based on literature due to instability
    - Risperidone ( $\Delta AIC = -278.1$ )
      - 1 covariate in final stepwise model
      - 5 covariates in best SOHGA candidate

# Example Model Search Space

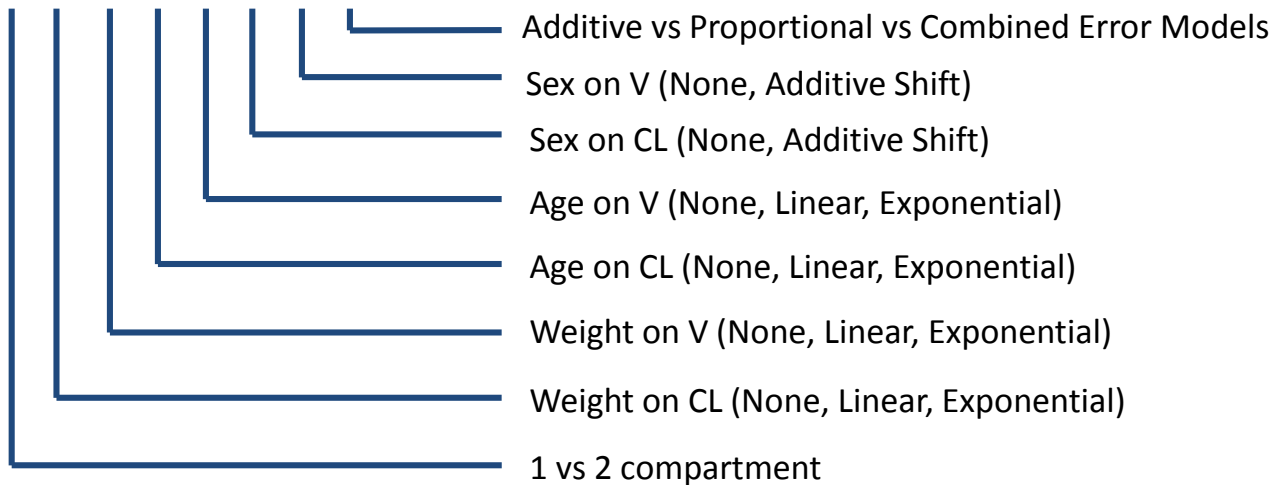
An example:

- Structure: 1, 2 compartment distribution model
- Covariates: Weight on CL, V | Age on CL, V | Sex on CL, V
  - Linear:  $TV_{Param} = THETA_A + ((Cov_i - \widehat{COV}) * THETA_B)$
  - Exponential:  $TV_{Param} = THETA_A * e^{(Cov_i - \widehat{COV}) * THETA_B}$
- Statistical: Additive, Proportional, Combined

# Example Model Search Space

- Total number of models:

- $2 \times 3 \times 3 \times 3 \times 3 \times 2 \times 2 \times 3 = 1944$  possible combinations

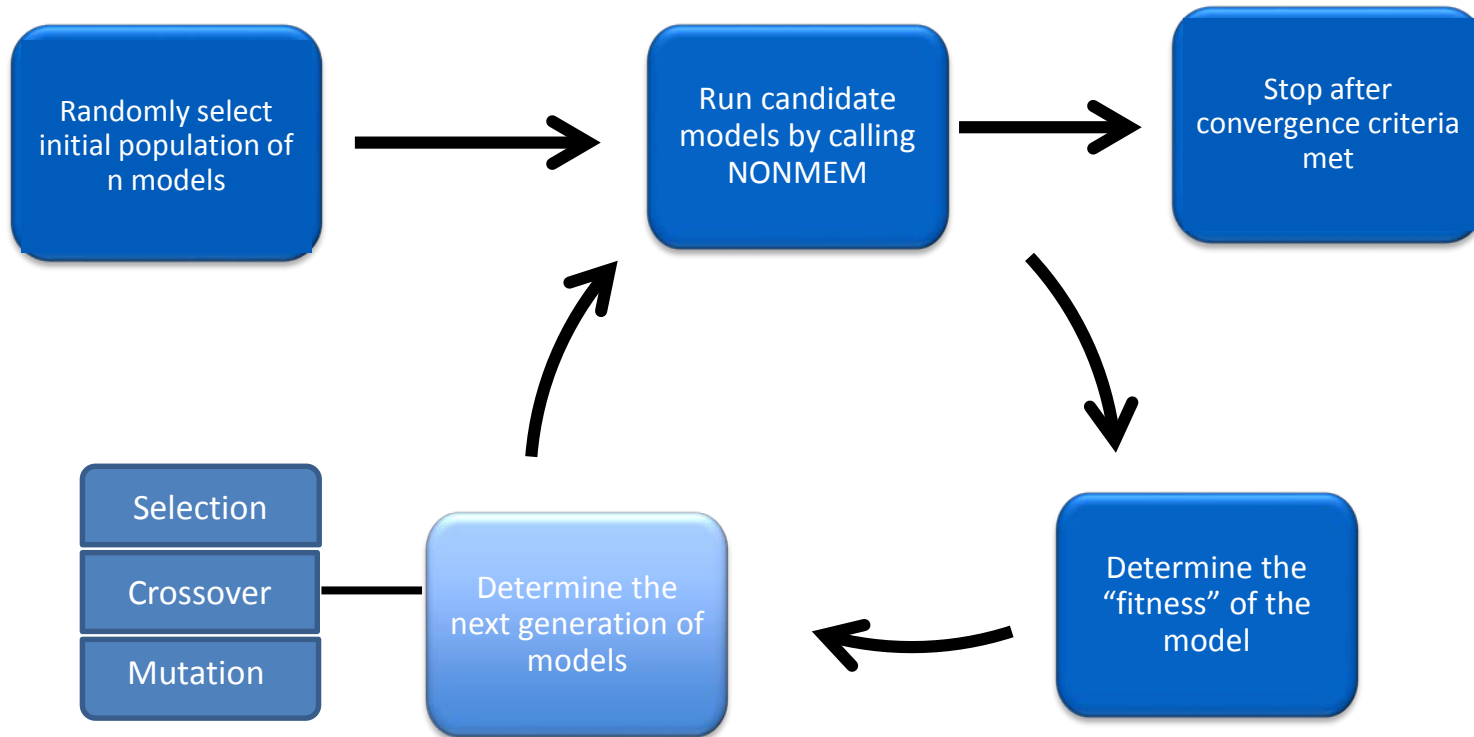


# Example Model Search Space

- Total number of models:
  - $2 \times 3 \times 3 \times 3 \times 3 \times 2 \times 2 \times 3 = 1944$  possible combinations

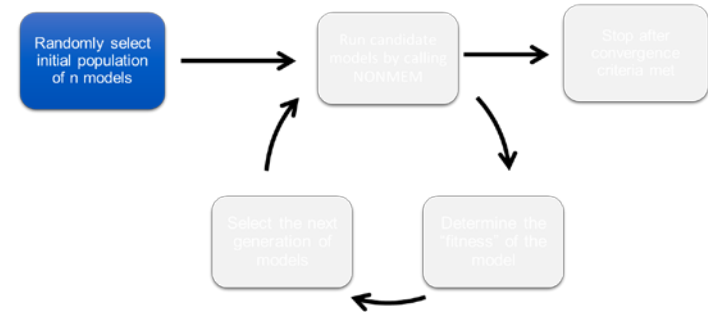
Model	$N_{\text{CMT}}$	Weight on CL	Weight on V	Age on CL	Age on V	Sex on CL	Sex on V	Error Model
1	1	None	None	None	None	None	None	Additive
2	1	Linear	None	None	None	None	None	Additive
3	1	Exponential	None	None	None	None	None	Additive
4	1	None	Linear	None	None	None	None	Additive
5	1	None	Exponential	None	None	None	None	Additive
...	...	...	...	...	...	...	...	...
1944	2	Exponential	Exponential	Exponential	Exponential	Additive	Additive	Combined

# Outline of GA



# Initial Population

- n models, or “individuals”, are randomly selected from the pool of all combinations
- Models are run simultaneously

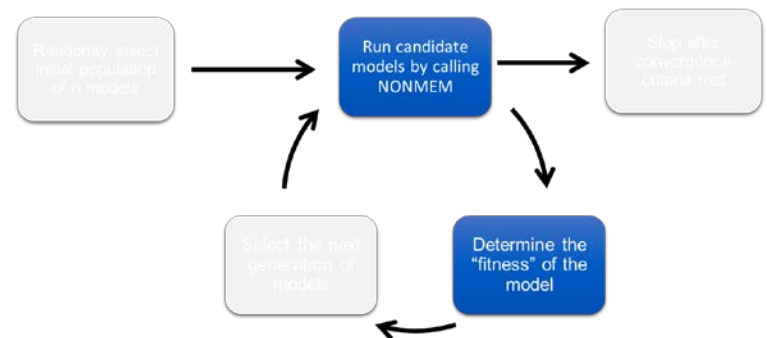


Model	N <sub>CMT</sub>	Weight on CL	Weight on V	Age on CL	Age on V	Sex on CL	Sex on V	Error Model
83	1	Linear	None	Linear	Exponential	None	Exponential	Additive
225	1	Linear	Exponential	Exponential	Linear	None	None	Proportional
343	1	Exponential	None	None	Linear	None	Linear	Proportional
800	2	None	Linear	Exponential	None	Exponential	None	Combined
1284	2	Exponential	Exponential	Linear	Exponential	None	None	Additive
1491	2	Exponential	None	None	Linear	None	Linear	Additive



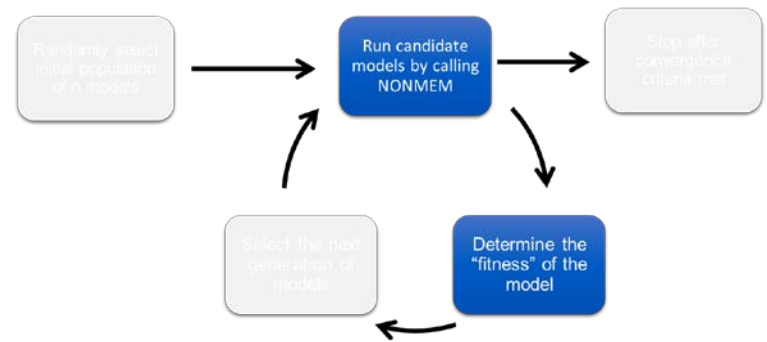
# Fitness

- How to determine how “fit” a model is?



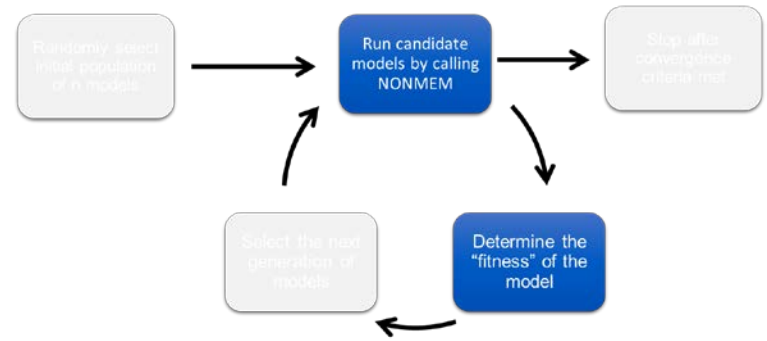
# Fitness

- How to determine how “fit” a model is?
- NONMEM objective function?



# Fitness

- How to determine how “fit” a model is?
- NONMEM objective function?
- Objective function + Penalty terms

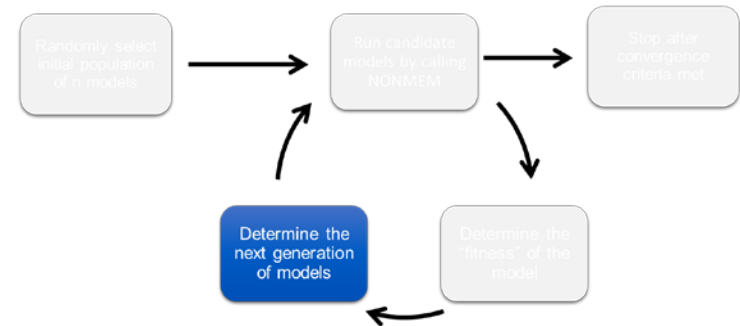


$$Fitness = \underbrace{-2LL + 2 * N_{Par}}_{AIC} + 20 * Penalty_{Converge} + 10 * Penalty_{Covar}$$

AIC

# Selection

- Tournament style selection
- Ranked selection method
  - Ideal when fitness values are close in magnitude



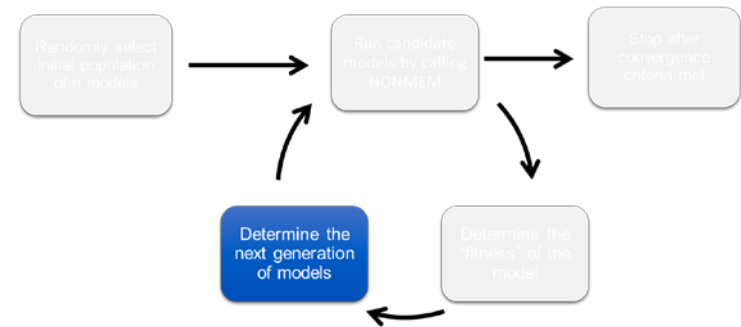
*for each model  $i$   
choose a random opponent model  $j$  (excluding  $i$ )  
the more fit model wins the tournament  
winner proceeds to the cross-over pool*

Initial Population		Crossover Pool	
Model	Fitness	Model	Fitness
83	100		
225	102		
343	98		
800	94		
1284	103		
1491	109		

A black arrow points from the first row of the 'Initial Population' table to the first row of the 'Crossover Pool' table, indicating that the model with ID 83 and fitness 100 is selected for the crossover pool.

# Selection

- Tournament style selection
- Ranked selection method
  - Ideal when fitness values are close in magnitude



*for each model  $i$   
choose a random opponent model  $j$  (excluding  $i$ )  
the more fit model wins the tournament  
winner proceeds to the cross-over pool*

Initial Population

Model	Fitness
83	100
225	102
343	98
800	94
1284	103
1491	109

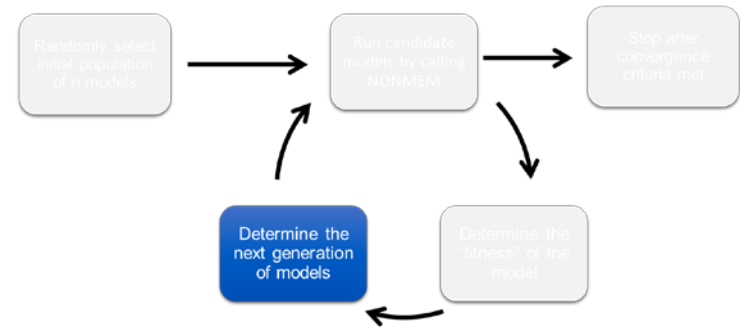
Crossover Pool

Model	Fitness
800	94



# Selection

- Tournament style selection
- Ranked selection method
  - Ideal when fitness values are close in magnitude



*for each model  $i$   
choose a random opponent model  $j$  (excluding  $i$ )  
the more fit model wins the tournament  
winner proceeds to the cross-over pool*

Initial Population

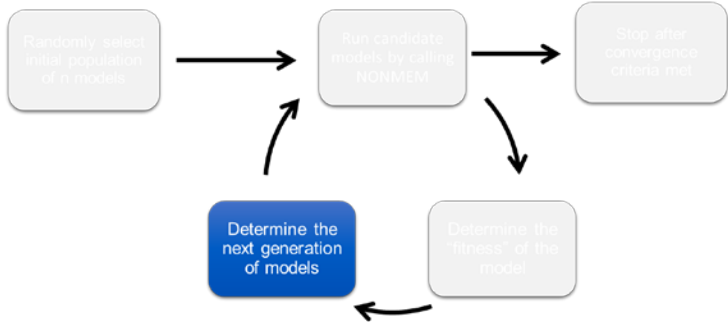
Model	Fitness
83	100
225	102
343	98
800	94
1284	103
1491	109

Crossover Pool

Model	Fitness
800	94
225	102



# Selection



- Tournament style selection
- Ranked selection method
  - Ideal when fitness values are close in magnitude

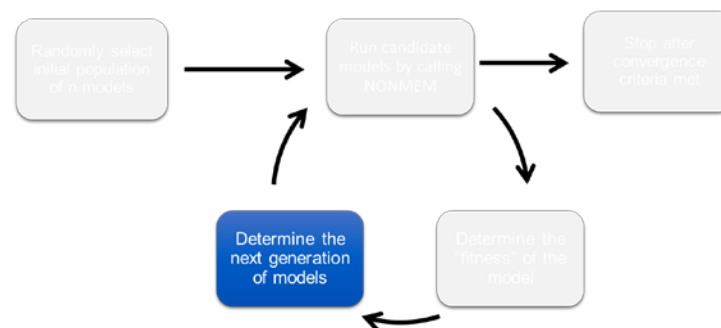
*for each model i*  
*choose a random opponent model j (excluding i)*  
*the more fit model wins the tournament*  
*winner proceeds to the cross-over pool*

Initial Population		Crossover Pool	
Model	Fitness	Model	Fitness
83	100	800	94
225	102	225	102
343	98	343	98
800	94	800	94
1284	103		
1491	109		



# Selection

- Tournament style selection
- Ranked selection method
  - Ideal when fitness values are close in magnitude



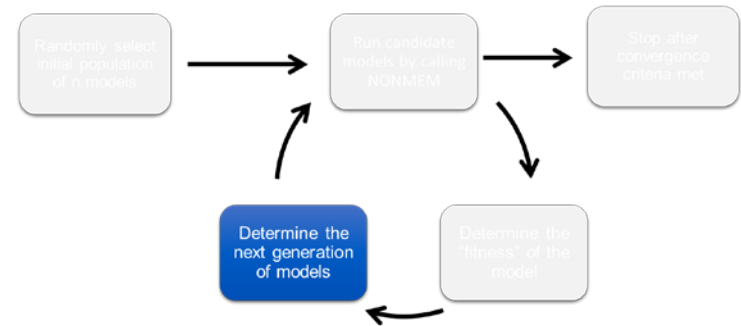
*for each model i*  
*choose a random opponent model j (excluding i)*  
*the more fit model wins the tournament*  
*winner proceeds to the cross-over pool*

Initial Population			Crossover Pool	
Model	Fitness		Model	Fitness
83	100	◀	800	94
225	102		225	102
343	98		343	98
800	94		800	94
1284	103		83	100
1491	109			



# Selection

- Tournament style selection
- Ranked selection method
  - Ideal when fitness values are close in magnitude



*for each model i*  
*choose a random opponent model j (excluding i)*  
*the more fit model wins the tournament*  
*winner proceeds to the cross-over pool*

Initial Population

Model	Fitness
83	100
225	102
343	98
800	94
1284	103
1491	109

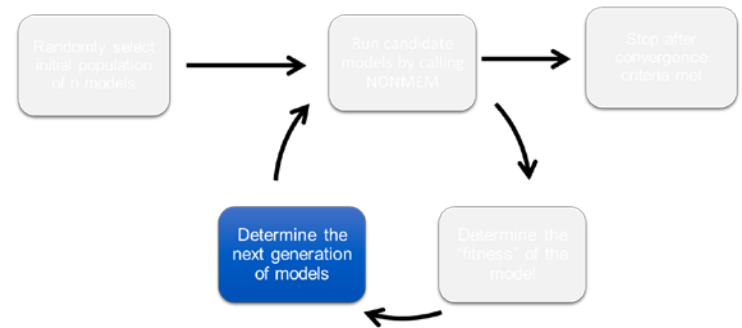
Crossover Pool

Model	Fitness
800	94
225	102
343	98
800	94
83	100
343	98



# Crossover

- Mimics biological reproduction
- Combines elements of well performing models to produce potentially more fit models
- Two-point crossover



# Crossover

## Parent Chromosomes

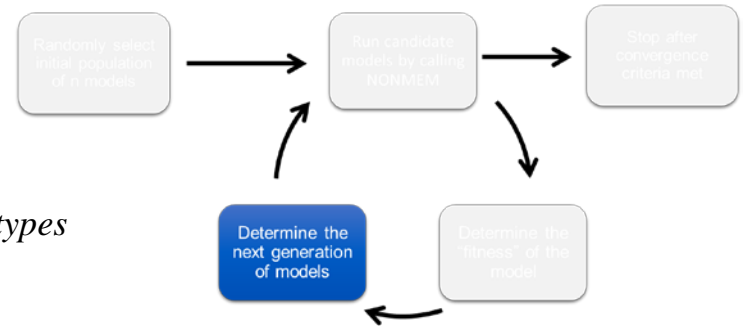
Mode I	Fitness	$N_{CM}$ T	Weight on CL	Weight on V	Age on CL	Age on V	Sex on CL	Sex on V	Error Model
800	94	2	None	Linear	Exponential 	None	Exponential 	None	Combined
343	98	▲1	Exponential 	None	None	Linear	None	Linear	Proportional

## Progeny

Mode I	Fitness	$N_{CM}$ T	Weight on CL	Weight on V	Age on CL	Age on V	Sex on CL	Sex on V	Error Model
---	---	2	None	None	None	Linear	None	None	Combined
---	---	1	Exponential 	Linear	Exponential 	None	Exponential 	Linear	Proportional

# Mutation

*for each model i*  
*for each gene j*  
*mutate gene (T/F) with probability 0.05*  
*if (mutate gene = T)*  
*newPhenotypeIndex = sample integer from 1 to length of phenotypes*  
*phenotype = phenotypes[newPhenotypeIndex]*  
*gene[j] = phenotype*

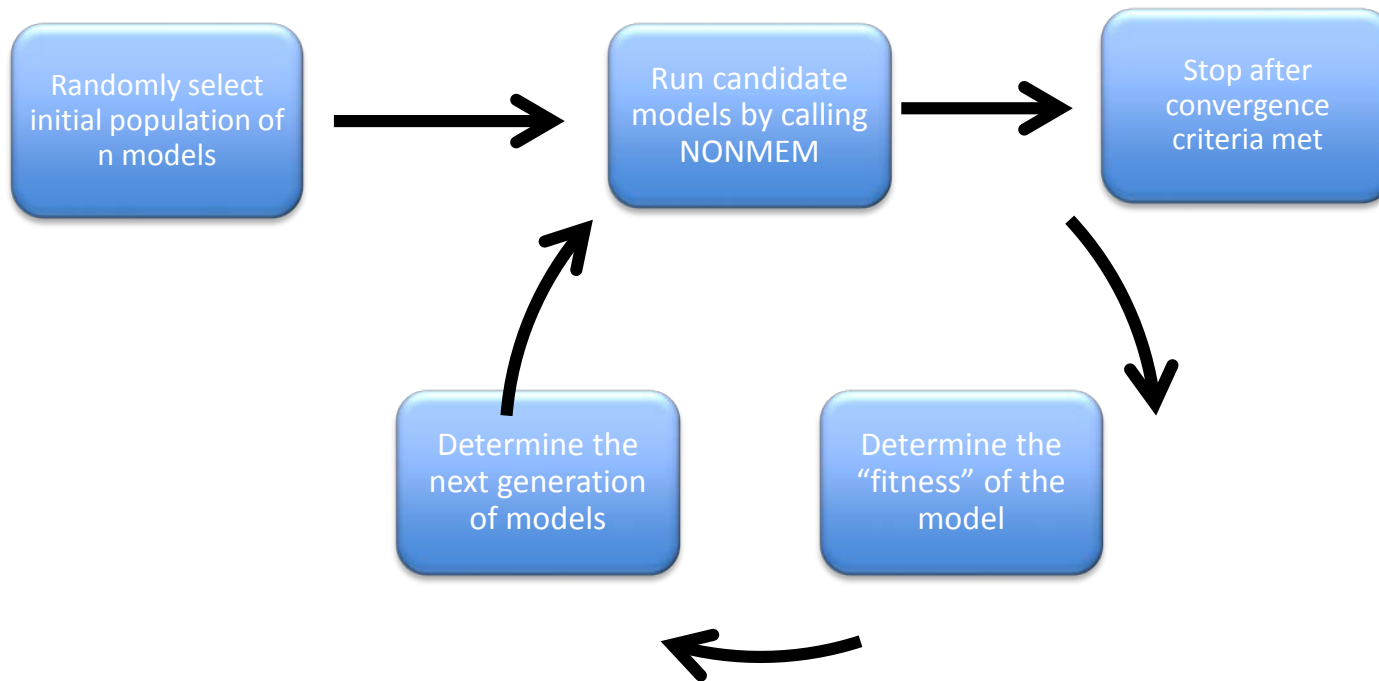


Mutate:

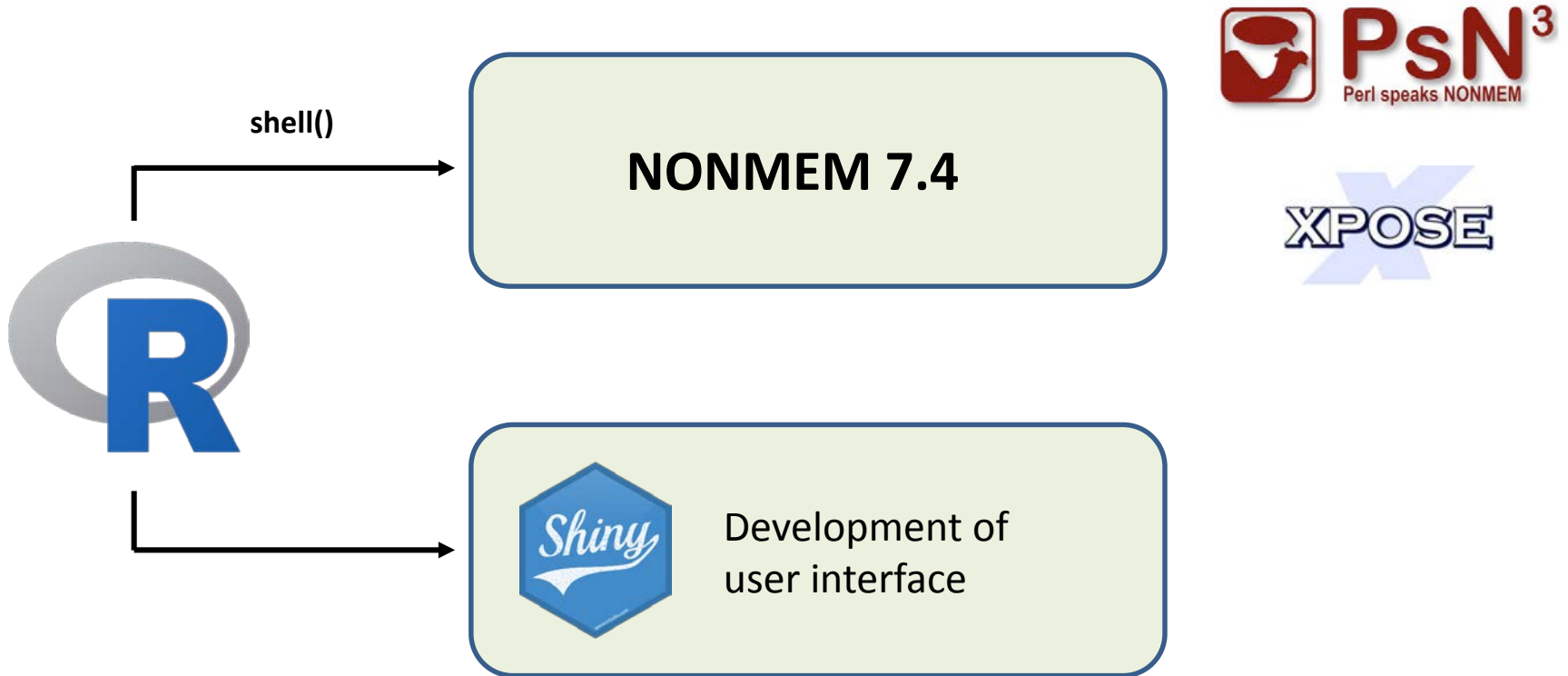
F F F F T F F

Weight on CL	Weight on V	Age on CL	Age on V	Sex on CL	Sex on V	Error Model
None	None	None	Linear	Additive	None	Combined

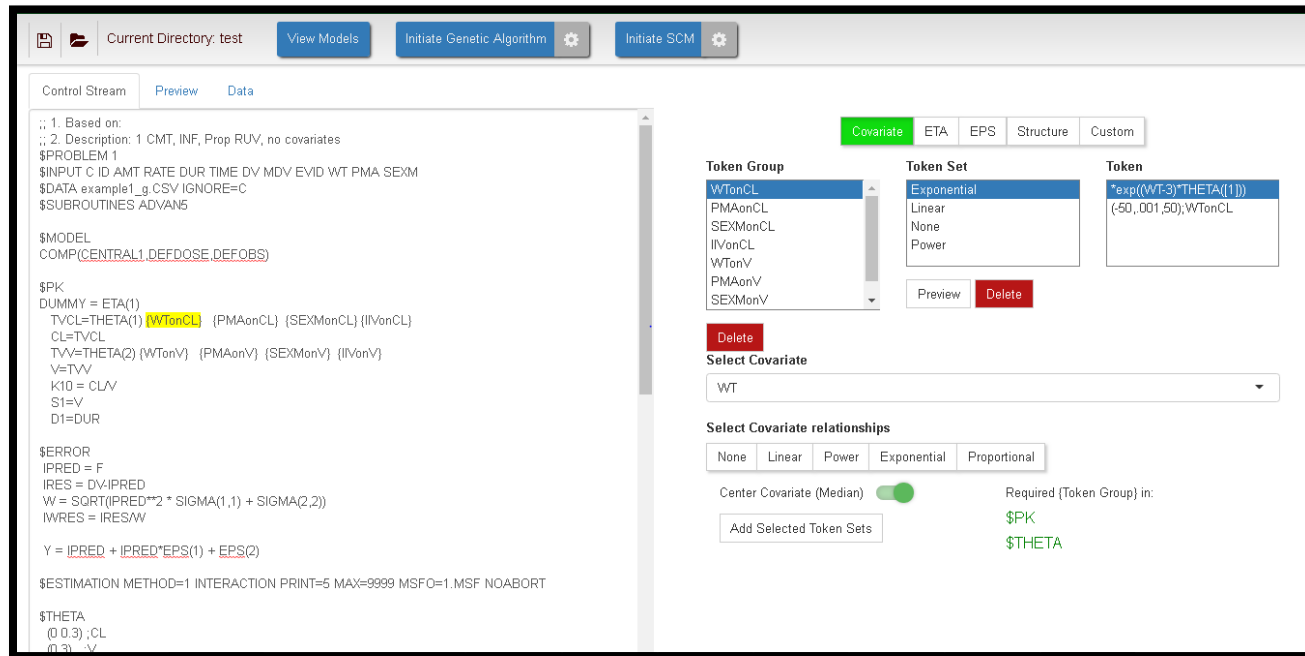
# Outline of GA



# Software



# Development of NONMEM Workbench to Implement Genetic Algorithm



The screenshot displays the NONMEM Workbench interface. The top navigation bar includes buttons for "View Models", "Initiate Genetic Algorithm", and "Initiate SCM". The main window is divided into a left pane for the Control Stream and a right pane for configuration options.

**Control Stream (Left Pane):**

```

:: 1. Based on:
:: 2. Description: 1 CMT, INF, Prop RUV, no covariates
$PROBLEM 1
$INPUT C ID AMT RATE DUR TIME DV MDV EVID WT PMA SEXM
$DATA example1_g.CSV IGNORE=C
$SUBROUTINES ADVAN5

$MODEL
COMP(CENTRAL1_DEFDOSE_DEFBOBS)

$PK
DUMMY = ETA(1)
TVCL=THETA(1) (WTonCL) (PMAonCL) (SEXMonCL) (IVonCL)
CL=TVCL
TVV=THETA(2) (WTonV) (PMAonV) (SEXMonV) (IVonV)
V=TVV
K10 = CL/V
S1=V
D1=DUR

$ERROR
IPRED = F
IRES = DV/IPRED
W = SQRT(IPRED**2 * SIGMA(1,1) + SIGMA(2,2))
IWRES = IRES/W

Y = IPRED + IPRED*EPS(1) + EPS(2)

$ESTIMATION METHOD=1 INTERACTION PRINT=5 MAX=9999 MSFO=1.MSF NOABORT

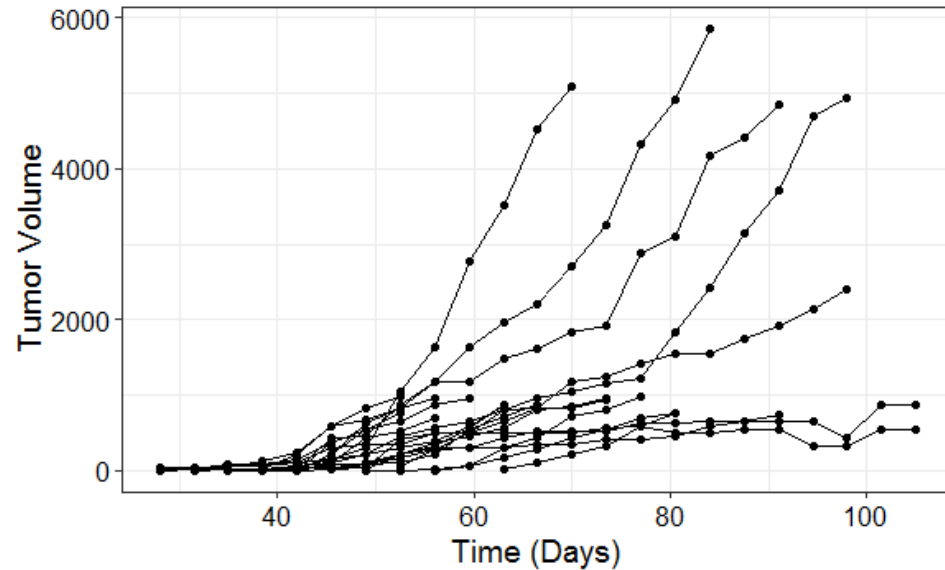
$THETA
(0.0,3);CL
(0.3);V
  
```

**Configuration Options (Right Pane):**

- Buttons:** Covariate (highlighted in green), ETA, EPS, Structure, Custom.
- Token Group:** A list containing WTonCL, PMAonCL, SEXMonCL, IVonCL, WTonV, PMAonV, and SEXMonV. WTonCL is selected.
- Token Set:** A list containing Exponential, Linear, None, and Power. Exponential is selected.
- Token:** A text field containing the expression:  $\text{exp}((W/3) \cdot \text{THETA}(1))$  with a range of (-50, 001, 50) and WTonCL.
- Select Covariate:** A dropdown menu with WT selected.
- Select Covariate relationships:** Buttons for None, Linear, Power, Exponential, and Proportional. A "Center Covariate (Median)" toggle is turned on.
- Required (Token Group) in:** A list containing \$PK and \$THETA.
- Add Selected Token Sets:** A button to apply the configuration.

# Case Study: Tumor Progression Modeling

- Unperturbed tumor growth trajectories of 22 LNCAP xenograft tumors were selected as test dataset





# Case Study: Tumor Progression Modeling

- 1584 unique models were created by the GA app with the combinations listed to the right:

Tumor Growth Model	Equation	# of $\theta$	# of IIV per $\theta^*$	# of RUV**	Number of Unique models
Exponential	$\frac{dV}{dt} = \lambda_0 \times V$	2	4	3	$4^2 \times 3 = 48$
Power	$\frac{dV}{dt} = \lambda_0 \times V^\gamma$	3	4	3	$4^3 \times 3 = 192$
Logistic	$\frac{dV}{dt} = \lambda_0 \times V \times \left(1 - \frac{V}{T_{max}}\right)$	3	4	3	$4^3 \times 3 = 192$
Gompertz	$\frac{dV}{dt} = \lambda_0 \times V \times \log\left(\frac{TUM_{max}}{V}\right)$	3	4	3	$4^3 \times 3 = 192$
Simeoni	$\frac{dV}{dt} = \frac{\lambda_0 \times V}{\left(1 + \left(\frac{\lambda_0}{\lambda_1} \times V\right)\right)^\psi}$	4	4	3	$4^4 \times 3 = 768$
Koch [1]	$\frac{dV}{dt} = \frac{\lambda_0 \times V \times 2 \times \lambda_1}{(\lambda_1 + 2 \times \lambda_0 \times V)}$	3	4	3	$4^3 \times 3 = 192$
					Sum: 1584

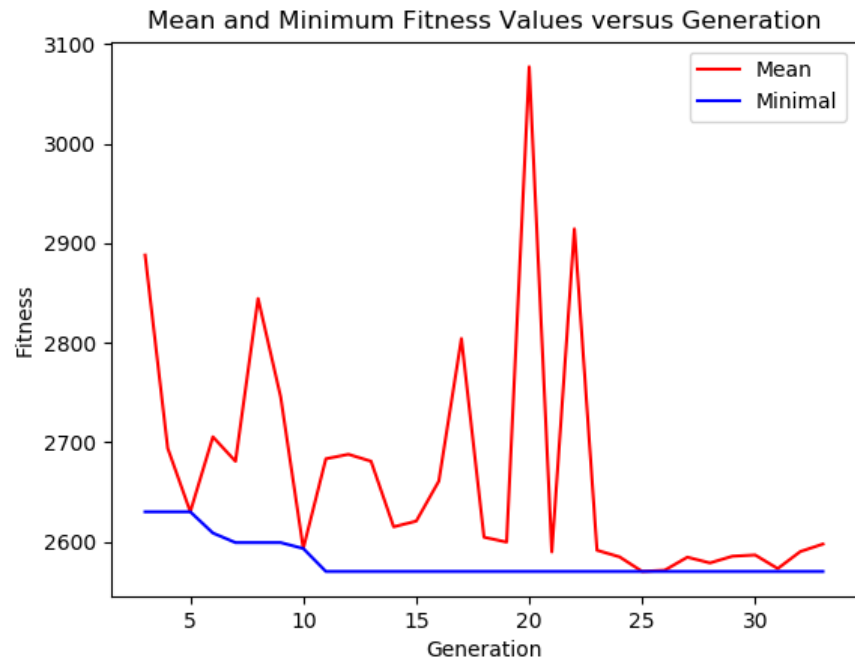
[1] Koch G1, Walz A, Lahu G, Schropp J. Modeling of tumor growth and anticancer effects of combination therapy. J Pharmacokinet Pharmacodyn. 2009 Apr;36(2):179-97.

\* The four IIV structures are: none, additive, proportional, and exponential.

\*\* The three RUV structures are additive, proportional, and additive plus proportional.

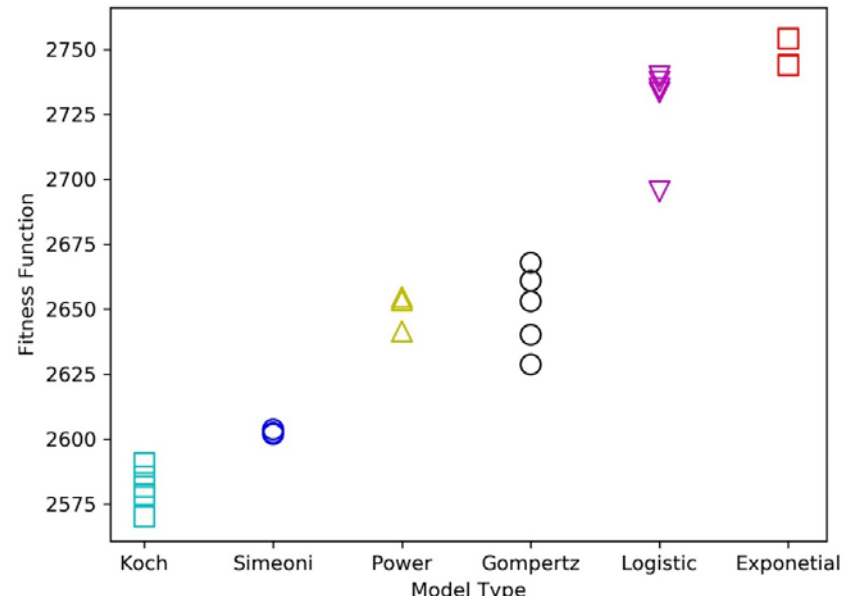
# Case Study: Tumor Progression Modeling

- Based on the available computation power (40 available cores), run 38 models simultaneously.
- It took on average 4 minutes to run a generation.
- The algorithm found the best model by the 15th generation
- To confirm model convergence, the system was allowed to continue for a total of 30 generations.
- 250 out of 1584 unique models were run by the 30th generation.



# Case Study: Tumor Progression Modeling

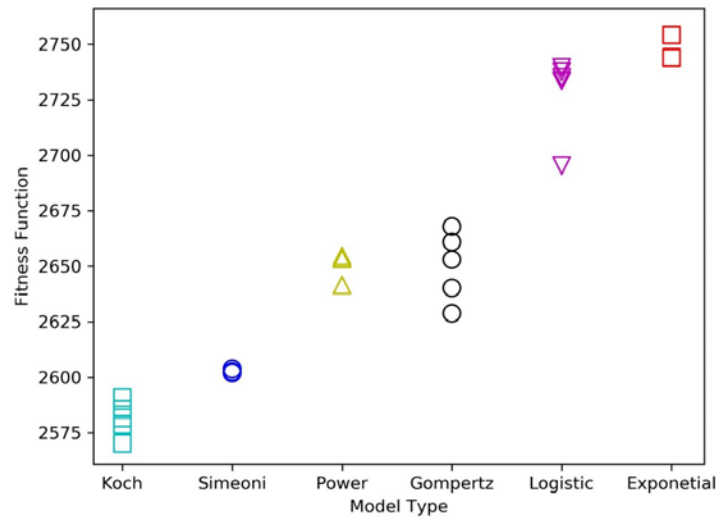
- The Koch growth model performed best for the xenograft tumor dataset.
  - Fitness value of **2572**
- The model with the best fitness had the following IIV characteristics:
  - An exponential IIV model on  $\lambda_0$
  - An exponential IIV model on  $\lambda_1$
  - An exponential IIV model on baseline.
  - The residual error model selected was additive plus proportional.
- Standard step-wise approach conducted by blinded colleague resulted in fitness value of **2748** (Simeoni structure)



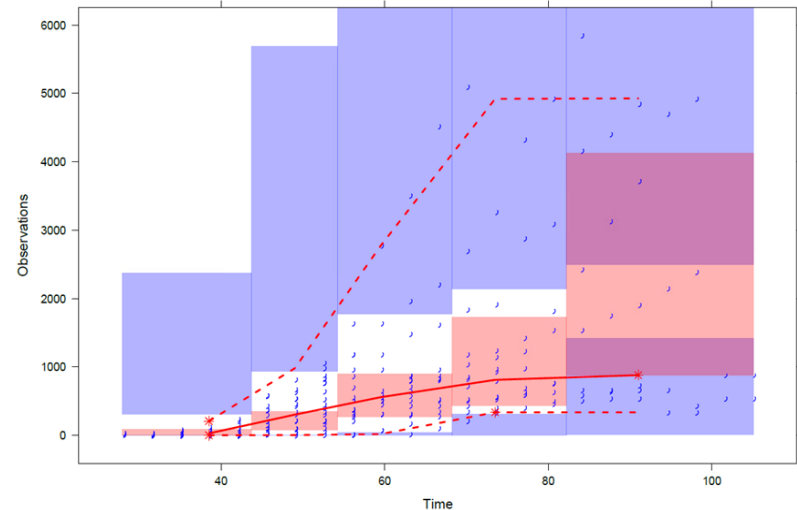
Top five fitness values for the six commonly used growth model categories

# Model Selection Results

The Koch growth model performed best for the test dataset. The model with the best fitness had the following IIV characteristics: a exponential IIV model on  $\lambda_0$ ; exponential IIV model on  $\lambda_1$ ; and exponential IIV model on baseline;. The residual error model selected was additive plus proportional.



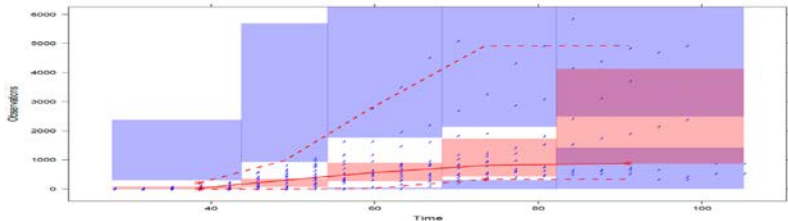
The plot of the top five fitness function for the six commonly used growth model categories (Koch, Simeoni, power, Gompertz, Logistic, and exponential).



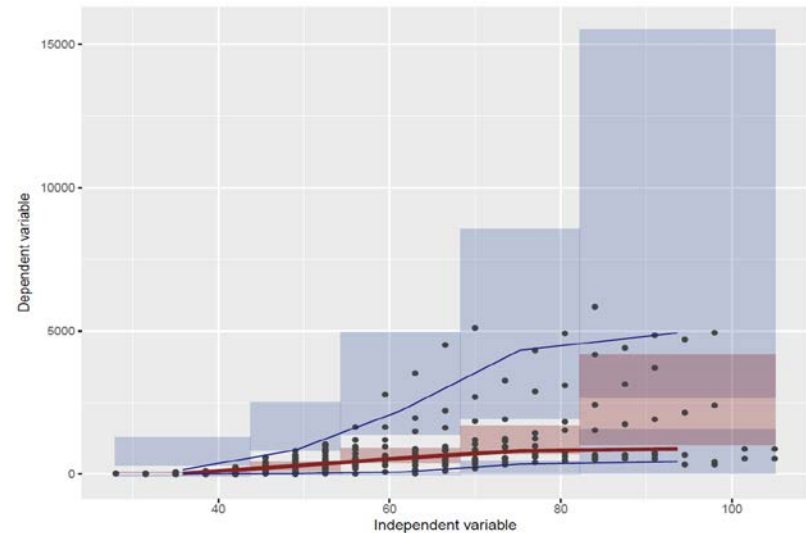
The VPC plot for the Koch model with the best fitness value of 2572. The red dashed lines are the predicted 5<sup>th</sup> and 95<sup>th</sup> percentiles.

# Model Selection Results

The best fitness function of the GA selected model is 2572 for the Koch model, while the typical approach to model building conducted by a “blinded” colleague resulted in a fitness of 2748 for a Simeoni model. In addition, the best Simeoni model found by GA gets a fitness function of 2602.



The VPC plot for the Koch model with the best fitness value of 2572. The red dashed lines are the predicted 5<sup>th</sup> and 95<sup>th</sup> percentiles.



The VPC plot for the manual picked Simeoni model with the fitness value of 2748. The blue solid lines are the predicted 5<sup>th</sup> and 95<sup>th</sup> percentiles.

# Limitations of SOHGA

- Only post-hoc visual predictive checks
- Single-objective
  - Ad hoc (user defined) weighting scheme
    - i.e., 10 points / parameter is  $\chi^2 = 0.0016$
- Equally valid yet very different candidate models are possible
- Does not consider feasibility
  - Could modify weighting scheme

# Conclusions

- The genetic algorithm identified a mixed effect model for risperidone PK and tumor trajectories that had substantially better OFV (and converted fitness) compared with the standard model search strategy.
- The current app can improve the accuracy and efficiency of model development. An automated solution for population PK/PD modeling will allow modelers to focus on hypothesis generation and model evaluation rather than text processing and model execution.

# Acknowledgements

- Mark Sale, Nuventra
- Mohamed Ismail, AbbVie and University at Buffalo
- Sihang Liu, University at Buffalo
- Nikhil Pillai, University at Buffalo
- Eric Sherer, Louisiana Technological University
- Risperidone
  - Bruce G. Pollock, University of Toronto
  - Jeffrey A. Lieberman, Columbia University
  - Stephen R. Marder, UCLA
- Tumor Trajectories
  - Beth Pflug, Roswell Park Cancer Center, Buffalo



Current Directory: test

View Models

Initiate Genetic Algorithm

Initiate SCM

Control Stream Preview Data

```
:: 1. Based on:
:: 2. Description: 1 CMT, INF, Prop RUV, no covariates
$PROBLEM 1
$INPUT C ID AMT RATE DUR TIME DV MDV EVID WT PMA SEXM
$DATA example1_g.CSV IGNORE=C
$SUBROUTINES ADVAN5

$MODEL
COMP(CENTRAL1,DEFDOSE,DEFOPS)

$PK
DUMMY = ETA(1)
TVCL=THETA(1) {WTonCL} {PMAonCL} {SEXMonCL} {IIVonCL}
CL=TVCL
TVV=THETA(2) {WTonV} {PMAonV} {SEXMonV} {IIVonV}
V=TVV
K10 = CL/V
S1=V
D1=DUR

$ERROR
IPRED = F
IRES = DV-IPRED
W = SQRT(IPRED**2 * SIGMA(1,1) + SIGMA(2,2))
IWRES = IRES/W

Y = IPRED + IPRED*EPS(1) + EPS(2)

$ESTIMATION METHOD=1 INTERACTION PRINT=5 MAX=9999 MSFO=1.MSF NOABORT

$THETA
(0 0.3) ;CL
(0 3) ;V
```

Covariate ETA EPS Structure Custom

Token Group

- WTonCL
- PMAonCL
- SEXMonCL
- IIVonCL
- WTonV
- PMAonV
- SEXMonV

Delete

Token Set

- Exponential
- Linear
- None
- Power

Preview

Delete

Token

\*exp((WT-3)\*THETA((1)))  
(-50,.001,50);WTonCL

Select Covariate

WT

Select Covariate relationships

None Linear Power Exponential Proportional

Center Covariate (Median)

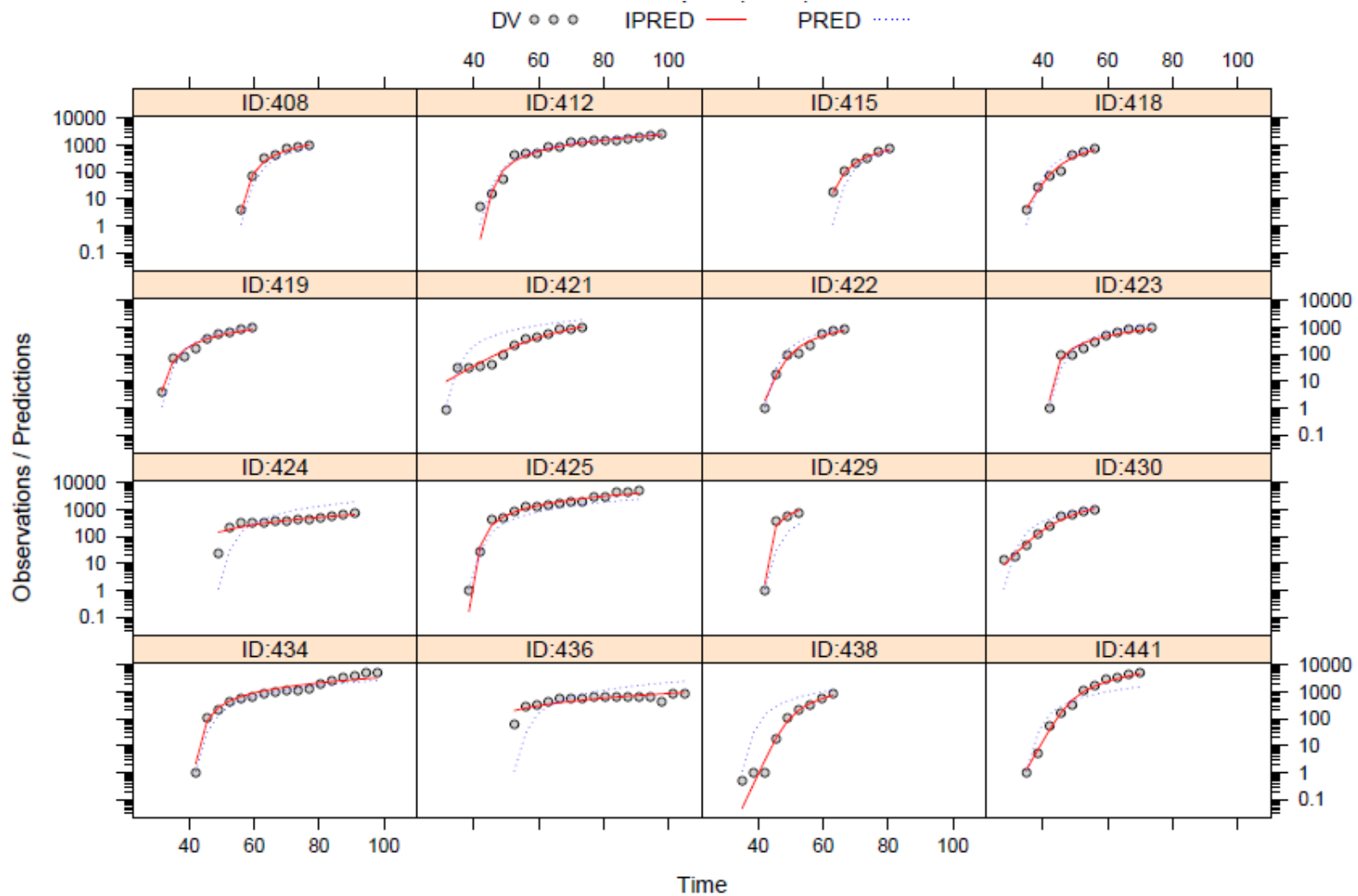
Required (Token Group) in:

Add Selected Token Sets

\$PK

\$THETA

# Case Study: Tumor Progression Modeling



# NONMEM Workbench

## Built in helpers for testing common model features

- Covariate effects
  - None, Linear, Power, Exponential, Proportional
- Interindividual Variability
  - None, Normal, Log-normal

The screenshot displays the NONMEM Workbench interface for configuring model features. At the top, there are tabs for 'Covariate', 'ETA' (highlighted in green), 'EPS', 'Structure', and 'Custom'. Below the tabs, the interface is divided into three main sections: 'Token Group', 'Token Set', and 'Token'. The 'Token Group' section contains a list of options: WTonCL, SEXMonCL, PMAonCL, IIVonCL (highlighted in blue), WTonV, SEXMonV, and PMAonV. The 'Token Set' section contains two options: 'Logarithmic' (highlighted in blue) and 'None'. The 'Token' section contains the expression '\*exp(ETA([1]))' (highlighted in blue) and the value '(0.01)'. Below these sections, there are 'Preview' and 'Delete' buttons. A red 'Delete' button is also present below the 'Token Group' list. At the bottom, there is a section for 'Select IIV relationships' with buttons for 'None', 'Normal', 'Logarithmic' (highlighted), and 'Normal (Proportional)'. A final button 'Add Selected Token Sets' is located at the bottom of the interface.

# NONMEM Workbench

## Built in helpers for testing common model features

- Covariate effects
  - None, Linear, Power, Exponential, Proportional
- Interindividual Variability
  - None, Normal, Log-normal
- Residual Error
  - Additive, Proportional, Combined

The screenshot displays the NONMEM Workbench interface for configuring the EPS token. At the top, there are tabs for 'Covariate', 'ETA', 'EPS' (which is highlighted in green), 'Structure', and 'Custom'. Below these tabs, the interface is divided into three main sections: 'Token Group', 'Token Set', and 'Token'. The 'Token Group' section contains a list of groups: PMAonCL, IIVonCL, WTonV, SEXMonV, PMAonV, IIVonV, and ERROR (which is selected). Below this list is a red 'Delete' button. The 'Token Set' section shows a list of sets: Additive (selected), Additive+Proportional, and Proportional. Below this list are 'Preview' and 'Delete' buttons. The 'Token' section displays the formula 'Y=F+EPS([1])' and the parameters '(0.01); Additive'. Below the 'Token Set' and 'Token' sections, there is a 'Select Residual Error Models' section with three buttons: 'Additive', 'Proportional', and 'Additive+Proportional'. At the bottom, there is an 'Add Selected Token Sets' button.

# NONMEM Workbench

## Built in helpers for testing common model features

- Covariate effects
  - None, Linear, Power, Exponential, Proportional
- Interindividual Variability
  - None, Normal, Log-normal
- Residual Error
  - Additive, Proportional, Combined
- Structure
  - Number of distribution compartments
  - +/- lag time
  - +/- saturable clearance

The screenshot displays the NONMEM Workbench interface with the 'Structure' tab selected. The interface is organized into several sections:

- Navigation:** Tabs for 'Covariate', 'ETA', 'EPS', 'Structure' (highlighted), and 'Custom'.
- Token Group:** A list of options including NCMT, WTonCL, SEXMonCL, PMAonCL, IIVonCL, WTonV, and SEXMonV.
- Token Set:** A list containing '+1cmt' and 'None', with 'Preview' and 'Delete' buttons below.
- Token:** A list of model parameters including 'COMP(CENTRAL2)', 'TVQ12 = THETA([1])', 'Q12=TVQ12', 'TVV12=THETA([2])', 'V12=TVV12', 'K12=Q12/V', and 'K21=Q12/V12'.
- Actions:** A red 'Delete' button is located below the Token list.
- Select Structural Component:** Tabs for 'Compartments' (selected), 'Tlag', and 'Michaelis-Menten'.
- Additional Compartments:** A row of three numbered buttons (1, 2, 3), with button '1' selected.
- Diagram:** A compartmental model diagram showing two circles. The left circle is labeled 'CENTRAL'. An arrow points from the left circle to the right circle, labeled 'K12'. A return arrow points from the right circle back to the left circle, labeled 'K21'.
- Footer:** An 'Add Selected Token Sets' button.

# NONMEM Workbench

## Built in helpers for testing common model features

- Covariate effects
  - None, Linear, Power, Exponential, Proportional
- Interindividual Variability
  - None, Normal, Log-normal
- Residual Error
  - Additive, Proportional, Combined
- Structure
  - Number of distribution compartments
  - +/- lag time
  - +/- saturable clearance
- Initial Estimates
  - Specify a range of initial estimates
- Custom token sets

The screenshot displays the NONMEM Workbench interface with the 'Custom' tab selected. The interface is organized into three main columns: 'Token Group', 'Token Set', and 'Token'. The 'Token Group' column shows a list with 'Custom' selected. The 'Token Set' column shows a list with 'Custom1' selected. The 'Token' column is currently empty. Below these lists are buttons for 'Delete', 'Preview', and 'Delete'. A dashed box highlights the configuration area for the selected 'Custom' token group, which includes a 'Token Group Name' field containing 'Custom'. Below this is a 'Token Set Name' field containing 'Custom1'. Underneath, there are two 'Token' fields labeled 'Token 1' and 'Token 2', both of which are empty. At the bottom of the dashed box are three buttons: 'Add Token Set', 'Add Token', and 'Save'. The top navigation bar includes tabs for 'Covariate', 'ETA', 'EPS', 'Structure', and 'Custom' (which is highlighted in green).

# Post-processing

- View final parameter estimates and precision
- Integrated with Xpose
- Automatically generate diagnostics plots
  - Individual plots
  - DV vs PRED
  - DV vs IPRED
  - RES vs TIME, DV
- Run PsN modules
  - VPC
  - Log-likelihood profiling

The screenshot shows the 'Parameters' tab in the PsN software. It displays three sections: 'Structural (THETA)', 'IIV (OMEGA)', and 'RUV (SIGMA)'. Each section contains a table with columns for 'Estimate' and 'RSE (%)'. The 'Structural (THETA)' section lists parameters CL, V, THETA3, THETA4, THETA5, and THETA6. The 'IIV (OMEGA)' section lists 'Prop'. The 'RUV (SIGMA)' section lists 'Add'.

Structural (THETA)		
	Estimate	RSE (%)
CL	0.30	7.09
V	3.85	5.61
THETA3	0.06	18.37
THETA4	1.36	6.96
THETA5	-2.65	19.55
THETA6	0.08	9.45

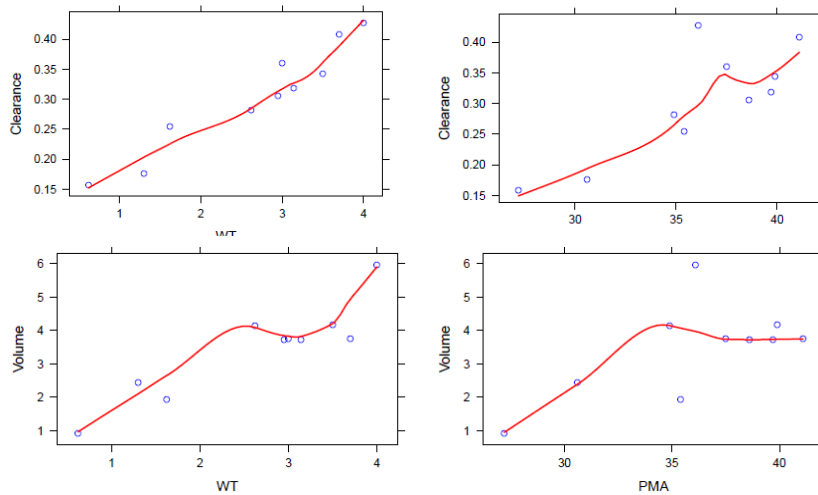
IIV (OMEGA)		
Prop	0.19	21.57

RUV (SIGMA)		
	Estimate	RSE (%)
Add	0.00	NA

# Post-Processing

- Linear regression between:
  - Covariates and post-hoc parameters
  - Covariates and random effects (ETAs)



Plots Parameters

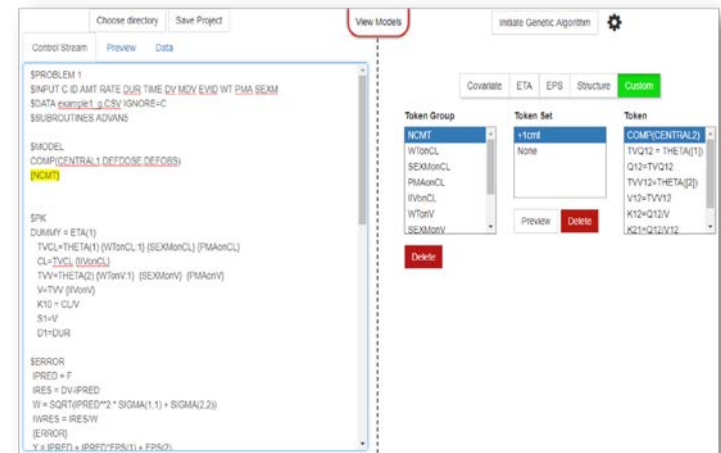
Covariate Model PsN

R	Model	Estimate	Pval.x
Linear	WTonCL	0.051	0.0014
Exp	WTonCL	0.188	0.00107
Power	WTonCL	0.353	0.00118
Linear	PMAonCL	0.013	0.00338
Exp	PMAonCL	0.046	0.00203
Power	PMAonCL	1.58	0.00178
Linear	WTonV	0.668	0.00821
Exp	WTonV	0.258	0.00665
Power	WTonV	0.501	0.00433
Linear	PMAonV	0.12	0.099
Exp	PMAonV	0.055	0.0388
Power	PMAonV	1.903	0.0315
Linear	WTonETA1	0	NaN
Linear	PMAonETA1	0	NaN
Linear	WTonETA2	0.188	0.00107
Linear	PMAonETA2	0.046	0.00203



# Software Application

- Free and open source
- Cross platform
- Extendable
- Features:
  - Implements robust search for globally optimal model solution
  - Organizes and displays models in tabular interface, allowing user to sort, filter, edit, create, and delete models seamlessly
  - Displays run results, parameter estimates, and precision
  - Integrated with Xpose and PsN
  - Linear regression models between covariates and post-hocs

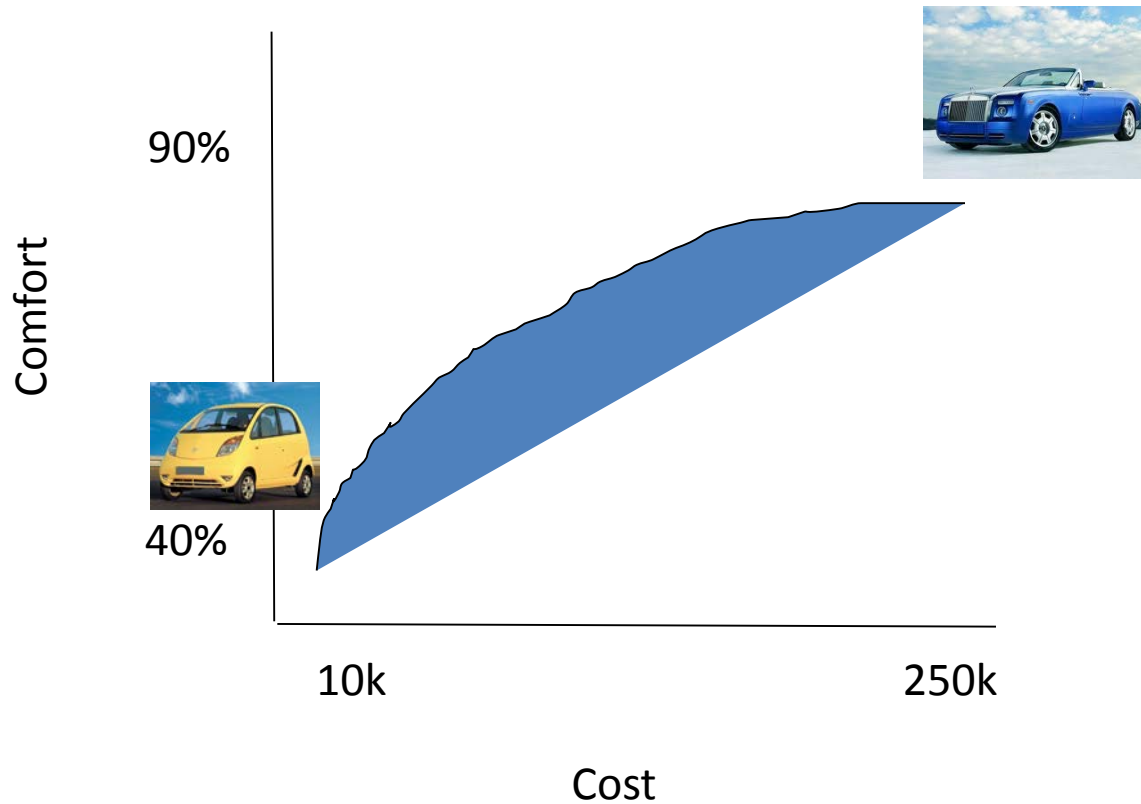


# Multi-objective optimization

- Single objective GA – composite function.
- Arbitrary (how much is each component of a model worth in terms of a penalty on fitness)
- Decisionmakers
  - preferred not to be presented with a single “best” option
  - wanted subjective elements captured
- NSGA-II (non-dominated sorted genetic algorithm)
  - Possible dimensions
    - Number of parameters
    - -2 Log Likelihood
    - NPDE P-value
    - “Quality” score
    - AIC
  - Uses elitism, diversity and mutation operators
  - Sorts on “non-dominated” solutions
    - Solutions that are at least as good as all others but better in one dimension

# Multi-objective optimization

- Optimize over many criteria



# MOGA Options

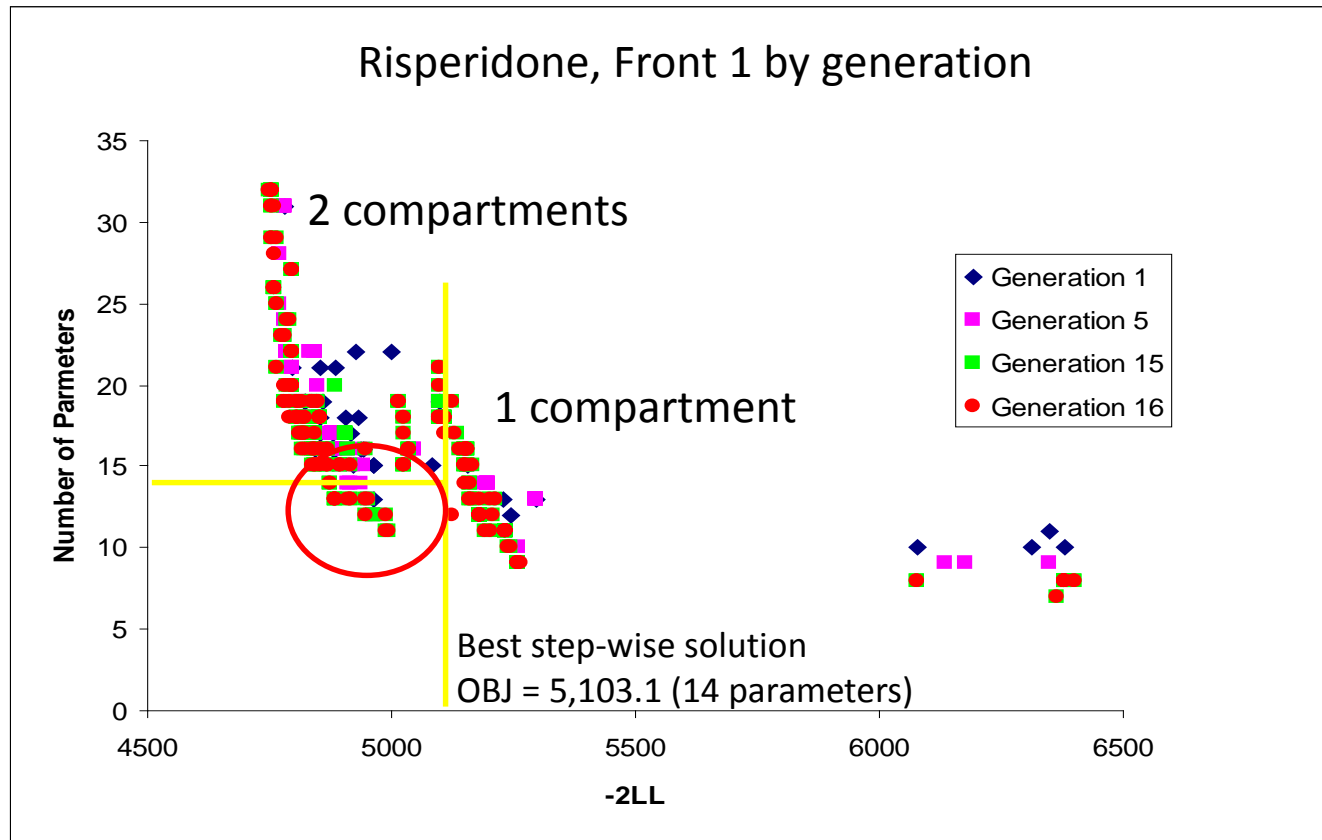
The screenshot shows the 'Options' dialog box for MOGA, with the 'Basic' tab selected. The dialog is divided into two main sections: 'Objectives' and 'User R Code'. The 'Objectives' section contains several checkboxes, with 'Use AIC', 'Use NPDE', 'Use Condition Number', 'Quality', 'Convergence', 'Covariance', and 'Correlation' checked. The 'User R Code' section contains a 'Random seed' section with 'Use Default' selected and a 'Number of Bins' field set to 16. Below these are several other options, including 'DownHill q3 Generations', 'Include ga for non diagonal OMEGA', 'Save best?', 'Make XPose Plots', 'VPC plot semilog', and 'Keep Xpose Data Files'. At the bottom, there are several numerical input fields for 'Cross over/genome', 'Mutation rate', 'Population size', 'Generation limit', 'Lower limit for non-crash', 'Minimization timeout (minutes)', 'Update time limit (minutes)', and 'Covariance time (minutes)', each with a 'Default' checkbox.

Option	Value / State
OBJ	<input type="checkbox"/>
Use AIC	<input checked="" type="checkbox"/>
Number of parameters	<input type="checkbox"/>
Use NPDE	<input checked="" type="checkbox"/>
Use Condition Number	<input checked="" type="checkbox"/>
User Supplied Rcode (not implemented)	<input type="checkbox"/>
"Quality"	<input checked="" type="checkbox"/>
Convergence	<input checked="" type="checkbox"/>
Covariance	<input checked="" type="checkbox"/>
Correlation	<input checked="" type="checkbox"/>
Random seed	Use Default (selected)
Use Clock	<input type="checkbox"/>
User Defined	<input type="checkbox"/>
DownHill q3 Generations	<input checked="" type="checkbox"/>
Include ga for non diagonal OMEGA	<input checked="" type="checkbox"/>
Save best?	<input checked="" type="checkbox"/>
Make XPose Plots	<input checked="" type="checkbox"/>
Number of Bins	16
VPC plot semilog	<input type="checkbox"/>
Keep Xpose Data Files	<input type="checkbox"/>
Cross over/genome	0.7
Mutation rate	0.01
Population size	300
Generation limit	110
Lower limit for non-crash	-10000
Minimization timeout (minutes)	60
Update time limit (minutes)	15
Covariance time (minutes)	60

- There are fewer MOGA options, since penalties do not need to be specified.

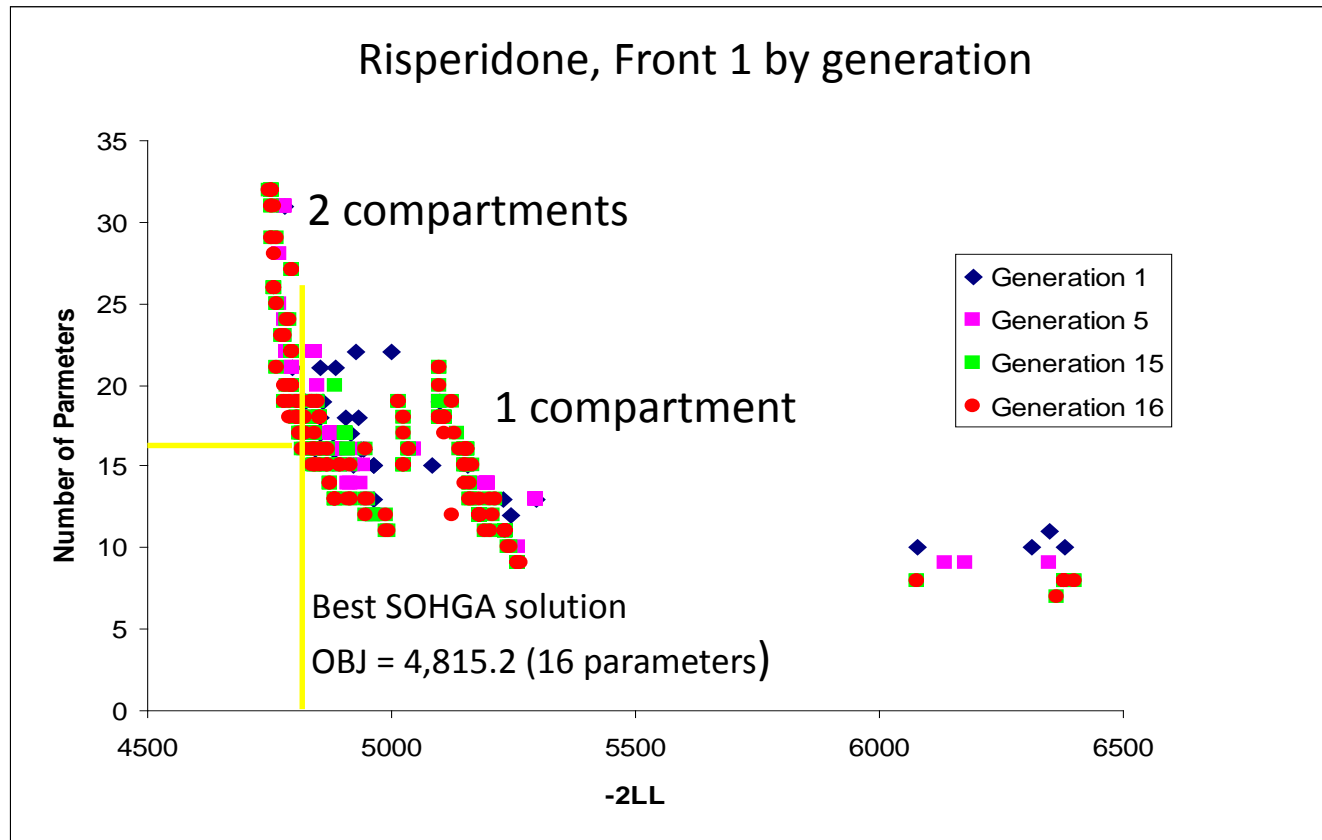
# Multi-objective genetic algorithm

- Front in  $-2 \times LL$  vs. # parameters space

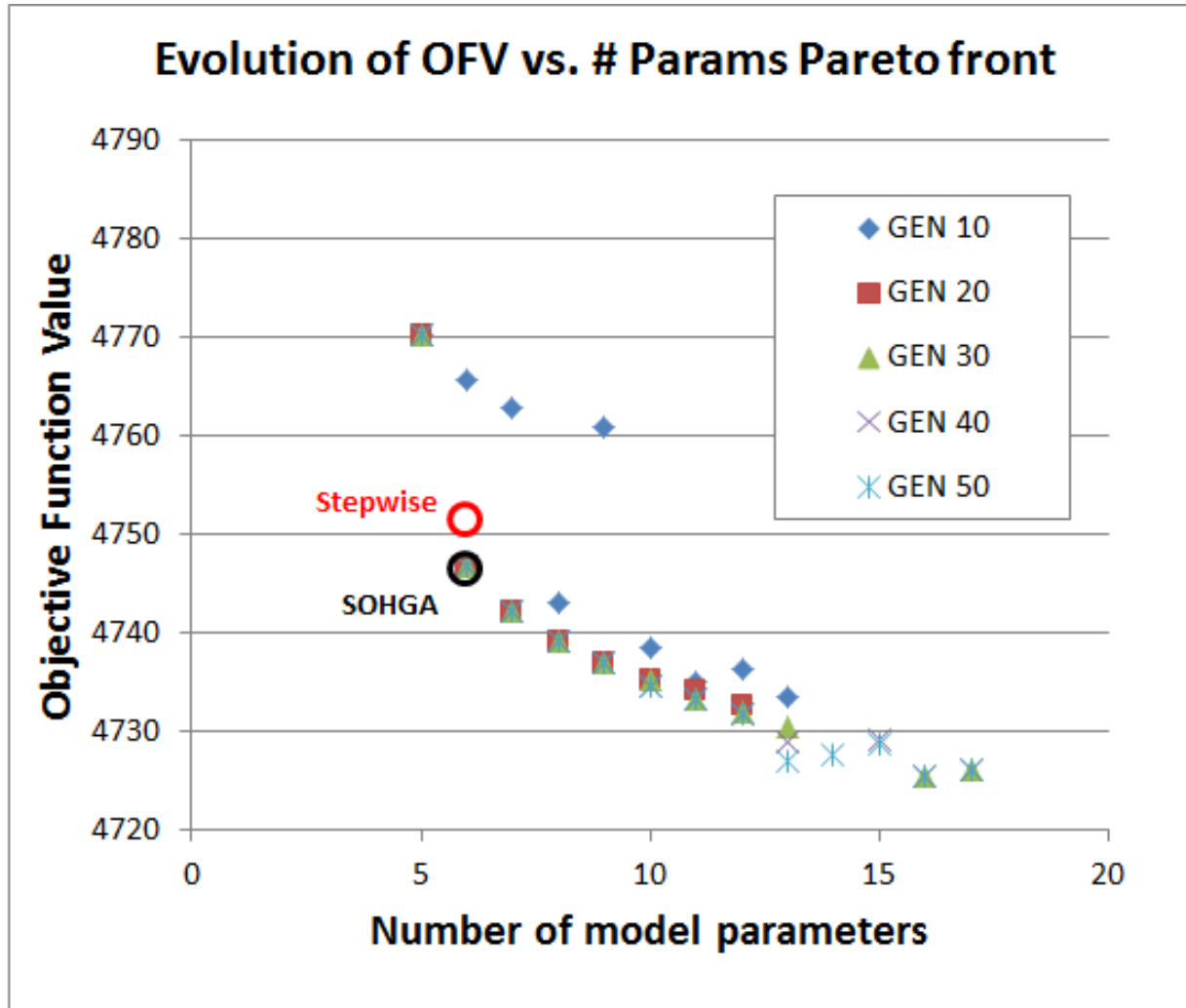


# Multi-objective genetic algorithm

- Front in  $-2 \times LL$  vs. # parameters space



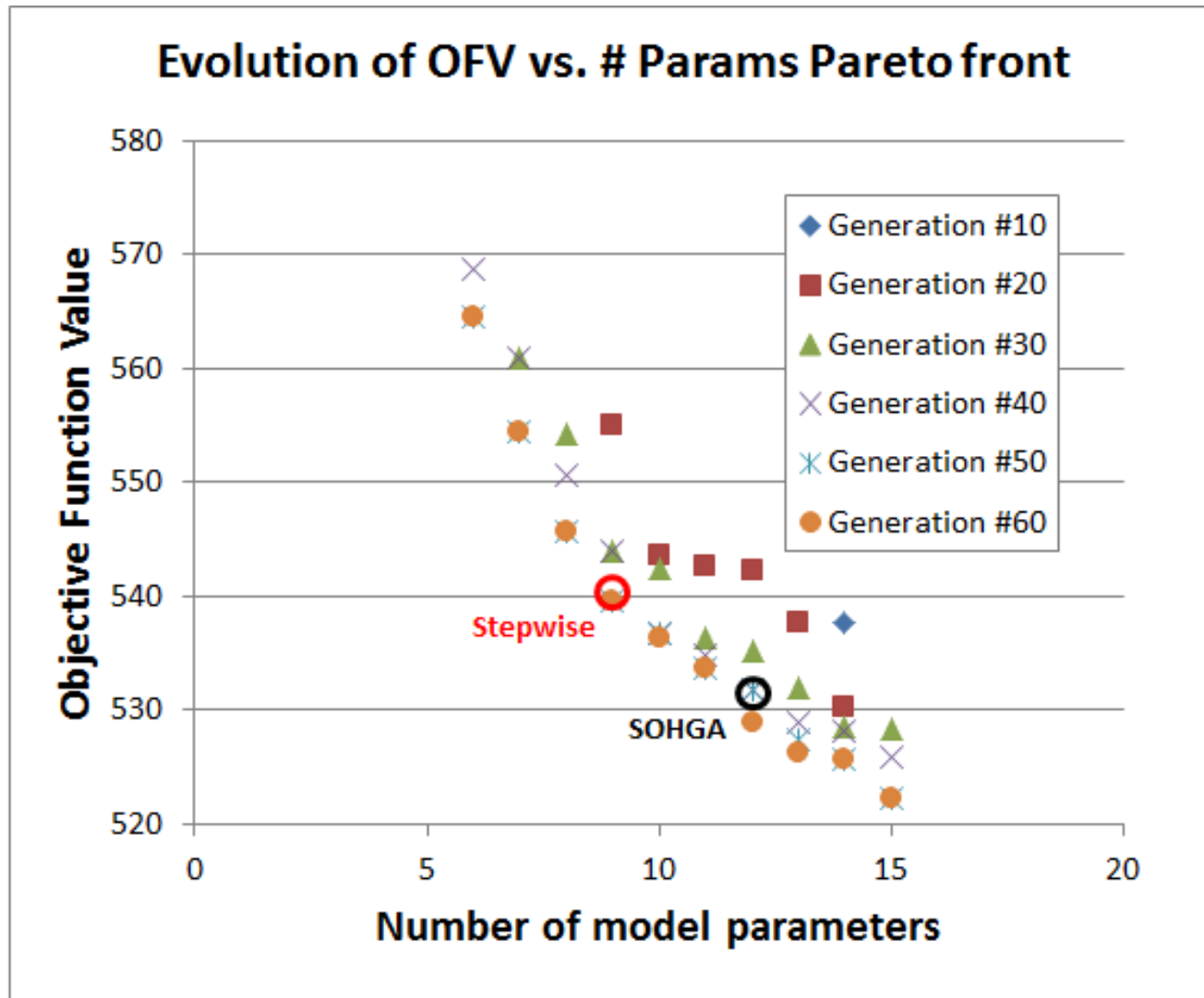
# Ziprasidone MOGA





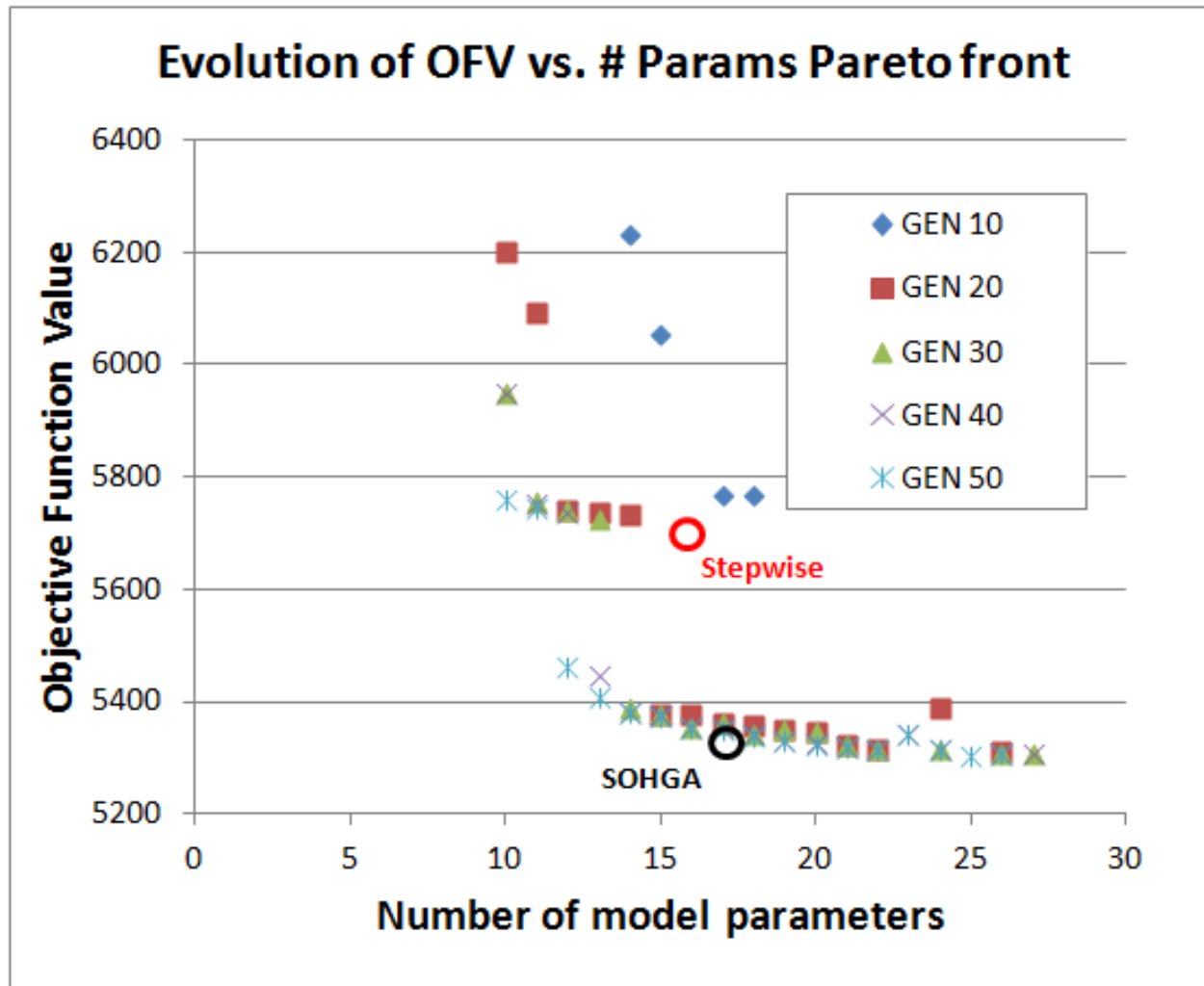


# Perphenazine MOGA





# Citalopram (IV) MOGA



# Citalopram (IV) MOGA

Number of Parameters	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Objective Function	5759.6	5745.7	5463.8	5409.7	5380.4	5374.5	5353.1	5350.6	5339.7	5329.9	5323.5	5318	5314.5	5343.5	5314.1	5303.3	5308.9
							stepwise - 5695.5 SOHGA - 5335.6										
IIV - CL																	X
IIV - V1			X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
IIV - Q	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
IIV - V2	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
COVARIANCE				X	X	X	X	X	X	X	X	X	X	X	X	X	X
ADDITIVE																	
PROPORTIONAL	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
COMBINED																	
CL - BMI	X	X	X	X	X		X	X			X	X		X	X	X	X
CL - WT		X	X			X			X			X	X	X	X	X	X
CL - SEX							X	X						X			X
CL - FAT										X	X	X	X	X	X		X
CL - AGE												X					X
V1 - BMI						X		X	X	X	X	X	X	X	X	X	X
V1 - SEX							X	X	X	X	X	X			X	X	X
V1 - WT									X				X				X
V1 - FAT																	X
Q - SEX					X	X			X	X			X	X	X	X	X
Q - WT									X	X	X	X	X				X
Q - FAT															X	X	X
Q - AGE														X			X
Q - FFM															X	X	X
V2 - FFM	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
V2 - SEX							X	X		X	X	X	X	X	X	X	X
V2 - WT										X	X	X	X		X	X	X
V2 - FAT																	X
V2 - AGE											X		X	X			X
V2 - BMI														X	X	X	X

# The Control Stream Template

- Resembles NONMEM control stream
- With additional {placeholder} text strings
  - Termed token groups

```
$PK
TVCL = THETA(1) {WTonCL}
CL = TVCL * EXP(ETA(1))

$THETA
(0, 10) ; CL
{WTonCL}
```

# The Control Stream Template

```
$PK
TVCL = THETA(1) {WTonCL}
CL = TVCL * EXP(ETA(1))

$THETA
(0, 10) ; CL
{WTonCL}
```

Placeholders are replaced with text strings to produce syntactically correct control streams

<u>Token Group</u>	<u>Token Set</u>	<u>Tokens</u>
WTonCL	Linear	+THETA(n)*WT (0.001); WTonCL
WTonCL	Exponential	*exp(THETA(n)*WT) (0.001); WTonCL

# Summary

- Implemented genetic algorithm method to more completely search the model space
- Single-objective, hybrid genetic algorithm
  - Arbitrary fitness function
  - Post-hoc assessment of predictive performance
- Multi-objective views additional dimensions
  - Generate non-dominated pareto front to evaluate trade-offs across models for given characteristics