Leveraging machine learning strategies for nonlinear mixed effects model selection

Robert Bies

University at Buffalo

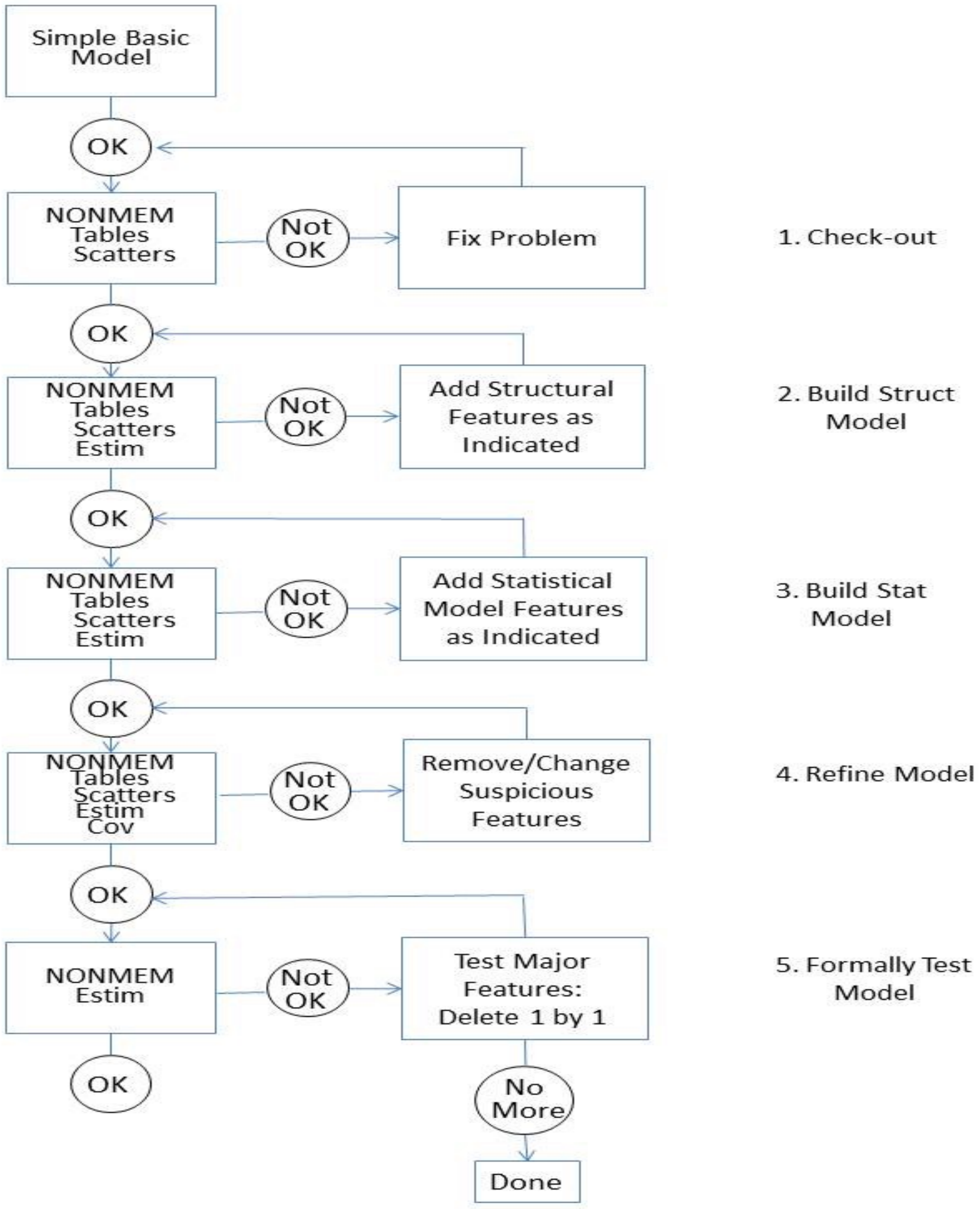
Local search:
“step-wise” regression

- 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 eliminati

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.



Br J Clin Pharmacol 2013 Jun 17 Epub ahead of print

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.

Potentially large solution space for a Pop PK model example

N	Initial conditions for NONMEM
2	Compartment structure 1 or 2 compartments
2	lag-time (yes/no)
4x2	Weight and age covariate on clearance None, additive, proportional, power function
4x2	Weight and age covariate on volume None, additive, proportional, power function
2x2	Sex covariate on clearance and volume of distribution
3x2	Between subject variability on CL, V and Ka absent or exponential
3	Residual error structure additive, proportional, combined
98304	

Potentially large solution space for a Pop PD model example

Total number of models:

4*4*4*4*4*4*4*4*4*4*3 = 3145728 possible combinations

- Additive vs Proportional vs Combined Error Models
- IIV on Tau of group 2 (None, Additive, Proportional, Exponential)
- IIV on K of group 2 (None, Additive, Proportional, Exponential)
- IIV on Tau of group 3 (None, Additive, Proportional, Exponential)
- IIV on K of group 3 (None, Additive, Proportional, Exponential)
- IIV on Tau of group 4 (None, Additive, Proportional, Exponential)
- IIV on K of group 4 (None, Additive, Proportional, Exponential)
- IIV on Lambda0 (None, Additive, Proportional, Exponential)
- IIV on Lambda1 (None, Additive, Proportional, Exponential)
- IIV on V0 (None, Additive, Proportional, Exponential)
- 4 model structure combination:
  Simeoni Growth + Cell Distribution Killing
  Koch Growth + Cell Distribution Killing
  Simeoni Growth + Signal Distribution Killing
  Koch Growth + Signal Distribution Killing

Ismail et al, JPKPD 2002 49(2)243-256

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

Holland, J.H. (1984). Genetic Algorithms and Adaptation. In: Selfridge, O.G., Rissland, E.L., Arbib, M.A. (eds) Adaptive Control of Ill-Defined Systems. NATO Conference Series, vol 16. Springer, Boston, MA. https://doi.org/10.1007/978-1-4684-8941-5_21
Holland, J. H. [1975]. “Adaptation in Natural and Artificial Systems,” University of Michigan Press, Ann Arbor.

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.

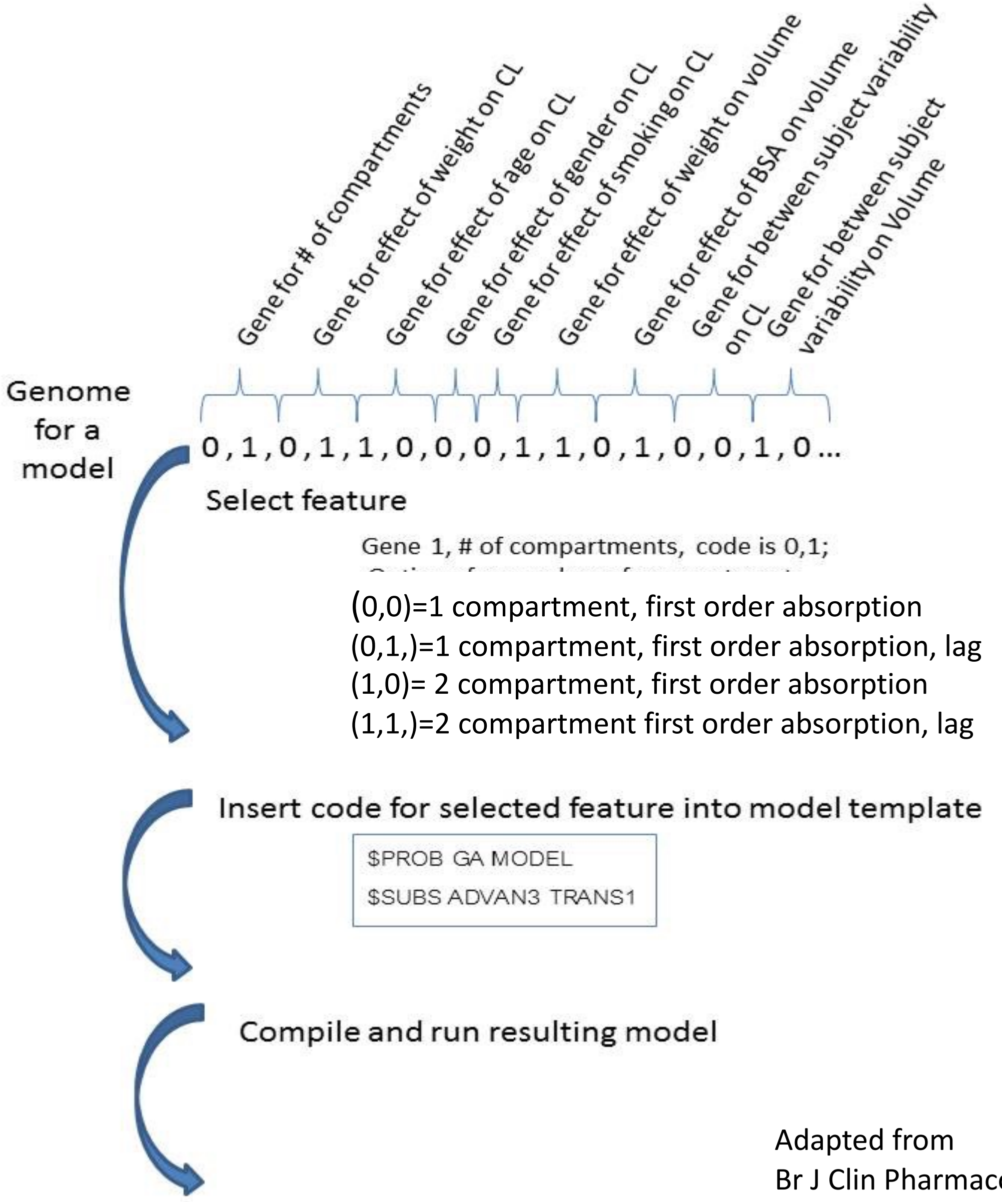
Holland, J.H. (1984). Genetic Algorithms and Adaptation. In: Selfridge, O.G., Rissland, E.L., Arbib, M.A. (eds) Adaptive Control of Ill-Defined Systems. NATO Conference Series, vol 16. Springer, Boston, MA. https://doi.org/10.1007/978-1-4684-8941-5_21
Holland, J. H. [1975]. “Adaptation in Natural and Artificial Systems,” University of Michigan Press, Ann Arbor.

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 ‘n’ generations

Genetic algorithms

- Implementation in the context of population PK/PD/Disease Progress 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 (e.g., number of compartment, covariate relationship) 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



Adapted from
Br J Clin Pharmacol 2013 Jun 17 Epub ahead of print

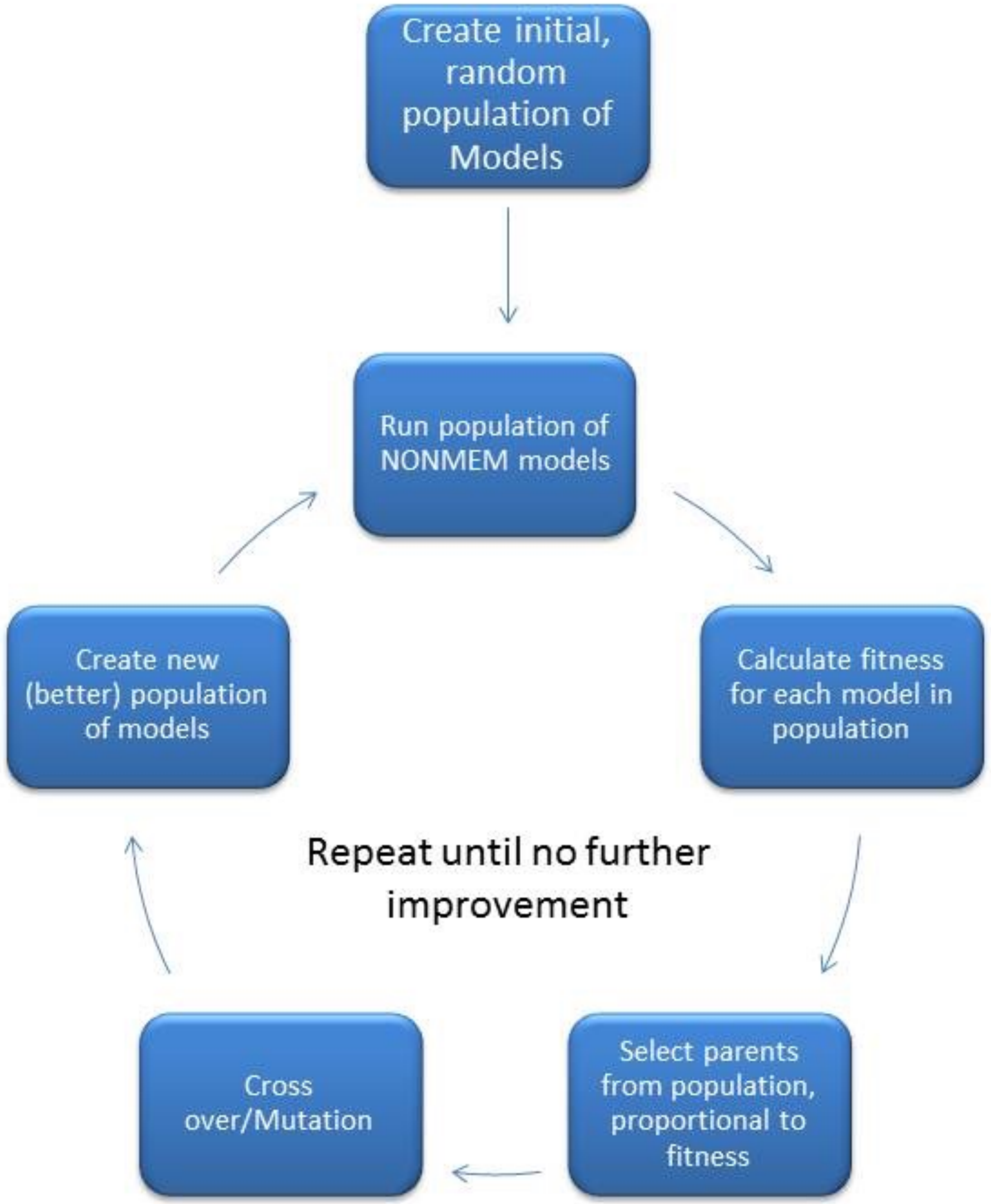
Figure 3. Coding of model features and translation into a model. If only two options are examined for a feature (e.g., the effect of Gender on Clearance) only 1 bit will be needed for that gene. If more than two options are examined (e.g., 4 for the basic structure, number of compartments) more than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

Model Feature	Feature Options	Bit string code	NONMEM code
Number of Compartments	1-cmt, 1st order absorption	0,0	Advan2 Trans2
	1-cmt, 1st order absorption, lag	0,1	Advan2 Trans2, alag
	2-cmt, 1st order absorption	1,0	Advan4, Trans4
	2-cmt, 1st order absorption, lag	1,1	Advan4, Trans4, alag
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)”

Adapted from: Br J Clin Pharmacol 2013 Jun 17 Epub ahead of print

Genetic algorithms

- Single-objective – user defined criteria, with defaults
  - Default composite fitness measure
    - -2 x log-likelihood
    - Penalty per model variable (**10 points**)
    - Penalties for failure to converge (400), covariance (400), and correlation (300)
- Multi-objective
  - Pareto front for pairs of objectives
  - Eg. -2xLL vs. # parameters



Br J Clin Pharmacol 2013 Jun 17 Epub ahead of print

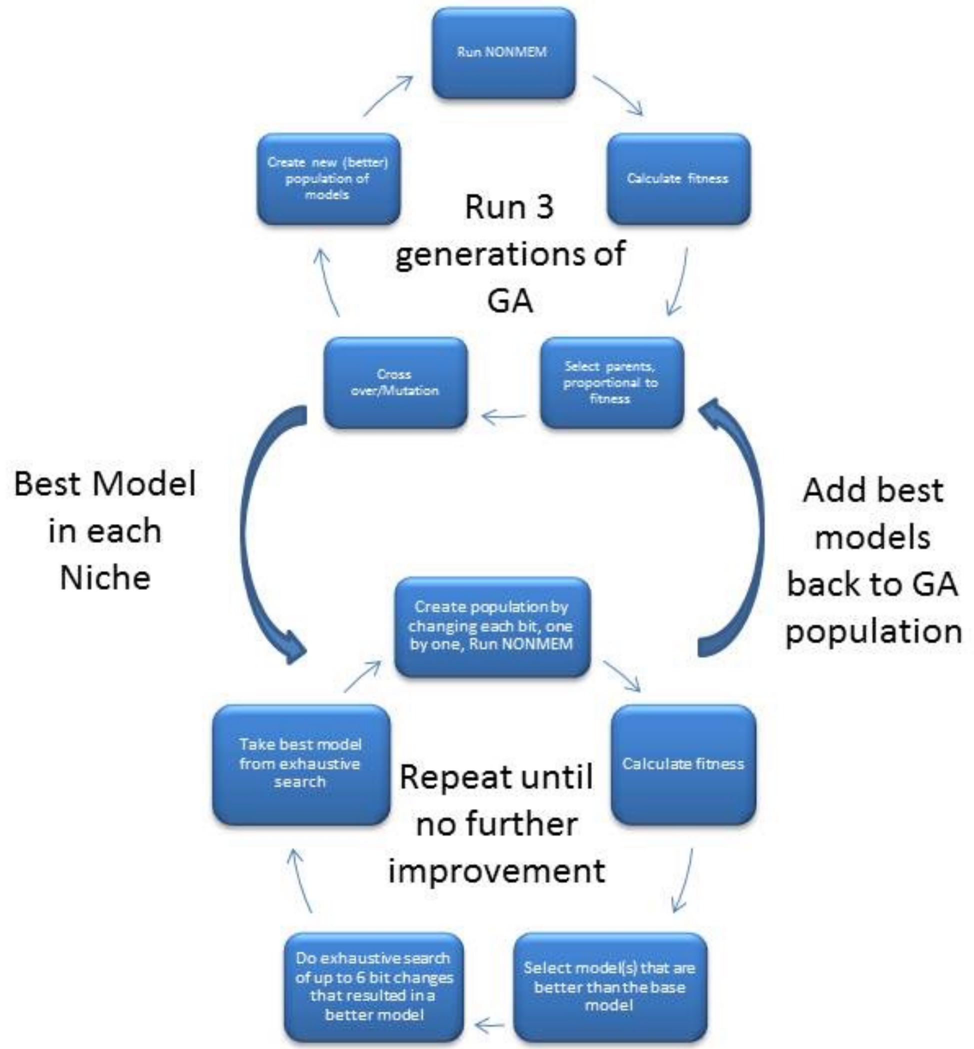


Figure 4. Simple Genetic algorithm. The algorithm is initialized with a random population. "Parents" for the next generation are selected (with replacement) for the next generation proportional to the user defined "fitness" of the model. These "parent" models are then paired off and undergo cross over and mutation to form the next generation of models.
215x279mm (96 x 96 DPI)

Single-objective, hybrid genetic algorithm (SOHGA) vs. step-wise approach

- Pharmacokinetic data for 7 compounds
  - Identical model options / decisions
- Compare information criteria of final models
  - Compare model structures

Sample sizes

Compound	Administration method	Number of patients	Number of concentration measurements
Citalopram	IV	331	1,324
DMAG	IV	67	1,148
Escitalopram	Oral	172	473
CATIE			
Olanzapine	Oral	523	1,527
Perphenazine	Oral	156	421
Risperidone	Oral	490	1,236
Ziprasidone	Oral	233	568

Model structure and covariate options

Compound	NONMEM model structures tested	First-order (FO) or first-order conditional (FOCE) estimation	Number of covariates collected
Citalopram, IV	ADVAN3, TRANS4	FOCE with interaction	7
DMAG, IV	ADVAN3, TRANS4 ADVAN11, TRANS4 (with potential for inter-occasion variability)	FOCE with interaction	10
Escitalopram, oral	ADVAN2, TRANS2 ADVAN4, TRANS4	FOCE with interaction	7
Olanzapine, oral	ADVAN2, TRANS2 ADVAN4, TRANS4	FOCE with interaction	9
Perphenazine, oral	ADVAN2, TRANS2 ADVAN4, TRANS4	FOCE with interaction	9
Risperidone, oral	ADVAN2, TRANS2 ADVAN4, TRANS4 (with 1, 2, or 3 clearance subpopulations)	FO	9
Ziprasidone, oral	ADVAN2, TRANS2	FOCE with interaction	9

Model convergence: SOHGA vs. step-wise

	Convergence	
	Final step-wise model	**Best SOHGA candidate**
Citalopram, IV	Successful	Successful
DMAG, IV	Successful	Successful
Escitalopram, oral	Successful	Successful
Olanzapine, oral	Required fixing K_a early in model building process	Successful
Perphenazine, oral	Required fixing K_a early in model building process	Successful
Risperidone, oral	Required fixing K_a early in model building process	Successful
Ziprasidone, oral	Required fixing K_a early in model building process	Successful

Model convergence: SOHGA vs. step-wise

	Convergence		Covariance step (condition number)	
	Final step-wise model	Best SOHGA candidate	Final step-wise model	Best SOHGA candidate
Citalopram, IV	Successful	Successful	Unsuccessful (N/A)	Successful (2,830)
DMAG, IV	Successful	Successful	Successful (20)	Successful (25)
Escitalopram, oral	Successful	Successful	Successful (39)	Successful (9)
Olanzapine, oral	Required fixing K_a early in model building process	Successful	Successful (12)	Successful (50)
Perphenazine, oral	Required fixing K_a early in model building process	Successful	Unsuccessful (N/A)	Successful (212)
Risperidone, oral	Required fixing K_a early in model building process	Successful	Successful (60)	Successful (1.17×10^6)
Ziprasidone, oral	Required fixing K_a early in model building process	Successful	Successful (3)	Successful (5)

Fits to data: SOHGA vs. step-wise

- 4 of 7 compounds have >10 point improvement with genetic algorithm approach
  - 10 point penalty for 1 parameter in SOHGA

Compound	Final stepwise model	Best SOHGA candidate model	$AIC_{SOHGA} - AIC_{stepwise}$
Citalopram, IV	AIC = 5,391.9	AIC = 5,369.6	-22.3
DMAG, IV	AIC = 9,871.7	AIC = 9,849.4	-22.3
Olanzapine, oral	AIC = 10,365.8	AIC = 9,895.3	-470.5
Risperidone, oral	AIC = 5,131.1	AIC = 4,853.0	-278.1

Fits to data: SOHGA vs. step-wise

- 4 of 7 compounds have >10 point improvement with genetic algorithm approach
- 3 of 7 have <10 point change

Compound	Final stepwise model	Best SOHGA candidate model	$AIC_{SOHGA} – AIC_{stepwise}$
Citalopram, IV	AIC = 5,391.9	AIC = 5,369.6	-22.3
DMAG, IV	AIC = 9,871.7	AIC = 9,849.4	-22.3
Escitalopram, oral	AIC = 2,737.7	AIC = 2,737.6	-0.1
Olanzapine, oral	AIC = 10,365.8	AIC = 9,895.3	-470.5
Perphenazine, oral	AIC = 560.7	AIC = 555.9	-4.8
Risperidone, oral	AIC = 5,131.1	AIC = 4,853.0	-278.1
Ziprasidone, oral	AIC = 4,463.2	AIC = 4,758.7	-4.5

Fits to data: SOHGA vs. step-wise

- 4 of 7 compounds have >10 point improvement with genetic algorithm approach
- 3 of 7 have <10 point change
- 0 of 7 are worse

Compound	Final stepwise model	Best SOHGA candidate model	$AIC_{SOHGA} – AIC_{stepwise}$
Citalopram, IV	AIC = 5,391.9	AIC = 5,369.6	-22.3
DMAG, IV	AIC = 9,871.7	AIC = 9,849.4	-22.3
Escitalopram, oral	AIC = 2,737.7	AIC = 2,737.6	-0.1
Olanzapine, oral	AIC = 10,365.8	AIC = 9,895.3	-470.5
Perphenazine, oral	AIC = 560.7	AIC = 555.9	-4.8
Risperidone, oral	AIC = 5,131.1	AIC = 4,853.0	-278.1
Ziprasidone, oral	AIC = 4,463.2	AIC = 4,758.7	-4.5

Model structure: SOHGA vs. step-wise

- Compartment structure
  - 6 of 7 (86%) agree

Compound	Final step-wise model	Best SOHGA candidate
Citalopram, IV	2	2
DMAG, IV	3	3
Escitalopram, oral	1	1 with estimated Ka
Olanzapine, oral	1	1 with estimated Ka
Perphenazine, oral	1	1 with estimated Ka
Risperidone, oral	1 with 3 component mixture on CL	**2 with 2 component mixture on CL**
Ziprasidone, oral	1	1

pyDarwin - Impetus

- Challenges with visual basic coding (sunset of code) for original SOHGA implementation
- Re-coded the GA in R using a R-shiny gui (Ismail 2022)
  - Issues with
    - model search space limitations
  - Better at search of space for tumor growth (n=1) and response model, worse at pop PK model example (n=1)
  - github.com/mhismail/nmga

Ismail et al, JPKPD 2002 49(2)243-256

pyDarwin - Impetus

- FDA HHS grant announcement
  - Development of a model selection method for population pharmacokinetics analysis by deep-learning based reinforcement learning (RFA-FD-21-027)
- GA is a “brute force” global optimization algorithm
- Opportunity to explore other ML algorithms for model identification
- https://github.com/certara/pyDarwin

pyDarwin - Model Selection

	Model Selection	Parameter Estimation
Search Space	Discrete	Continuous
Start	Trivial Model	Initial Estimate

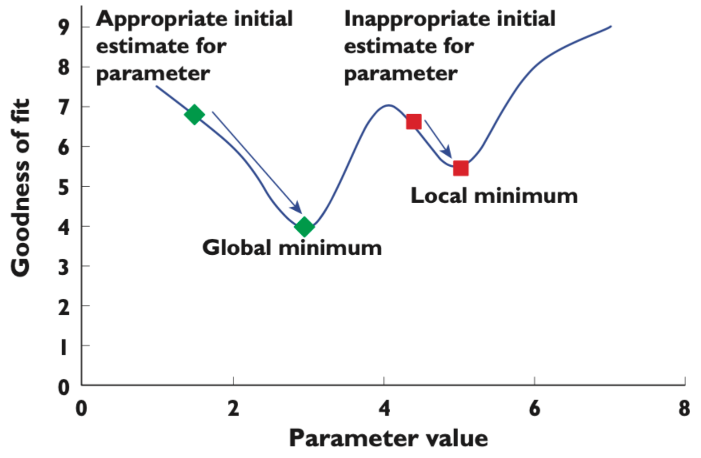
Traditional PK/PD Model Selection (Downhill Method):

- Start from the base model, then add features (COM#, covariate, etc.)
- Doesn't consider the complex interaction between the structural, covariate, and random effects
- Assumes the optimal solution is continuously downhill from every other point in the search space

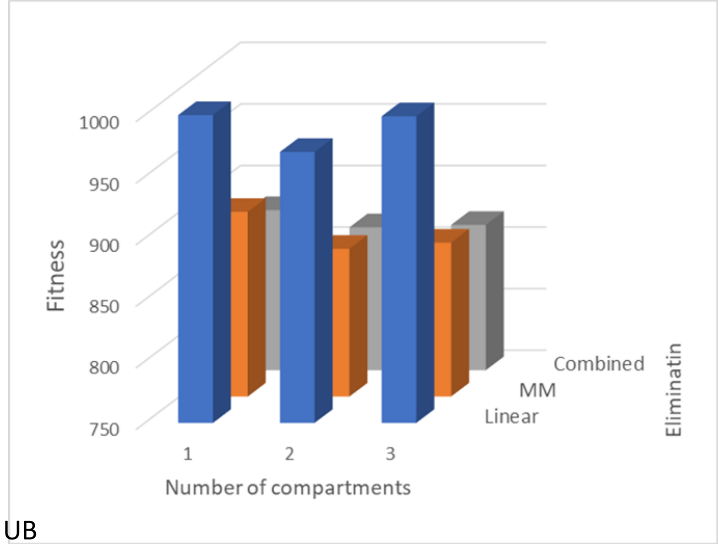
Machine Learning Model Selection:

- Start from multiple random models, test the models, have better idea about the model structure, update the information to the next generation and repeat this procedure

a. Local and global minima in a continuous simple search space



b. Discrete search space



Sale M, Sherer EA.. Br J Clin Pharmacol. 2015 Jan;79(1):28-39
Wade JR, Beal SL, Sambol NC. *J Pharmacokinet Biopharm*. 1994;22(2):165-177.
Chen X, Hamdan A, Wang S, et al. 2022. PAGE 30 (2022) Abstrt 10091
pyDarwin Handout by Certara

Slide courtesy of Xinnong Li, Ph.D. candidate UB

pyDarwin - Algorithms

- **Genetic Algorithm**[1]
- **Gaussian Process**[2]
- **Random Forest**[2]
- **Exhaustive Search**
- **Random Tree with Gradient Boosting**[2]
- Particle Swarm Optimization[3]

1. Implementation with DEAP, 2. Implementation with scikit-optimize, 3. Implementation with pySwarms

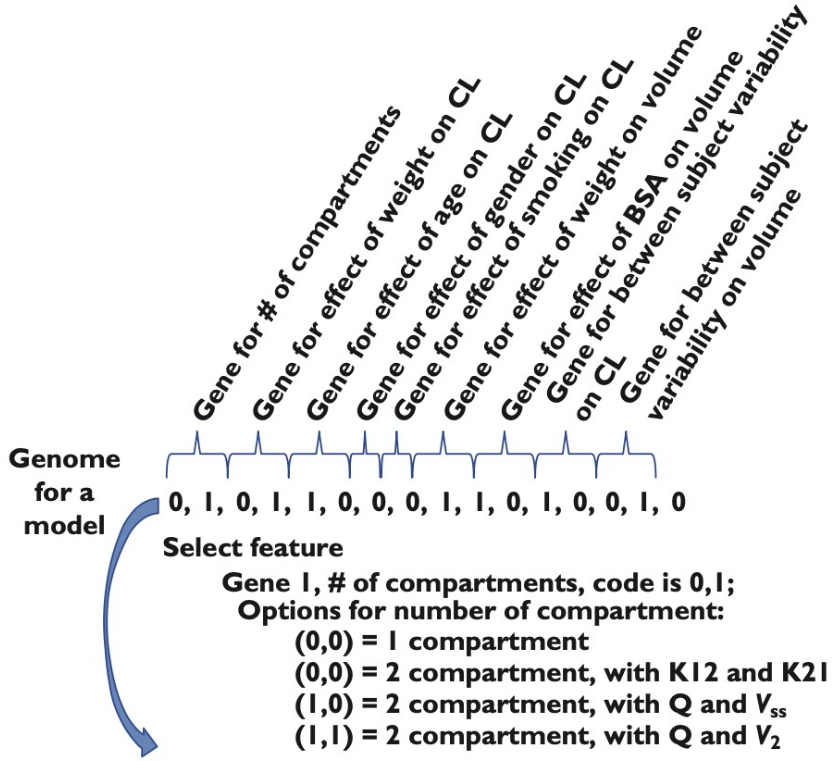
Slide courtesy of Xinnong Li, Ph.D. candidate UB

https://certara.github.io/pyDarwin/html/Overview.html

pyDarwin-Genetic Algorithm

Workflow:

- Randomly generate initial candidates
- Select the best parents (using **tournament selection**) from the candidate pool
- Using '**mutation**' and '**crossover**' to create next candidate pool
- Repeat the process until no further improvement was observed



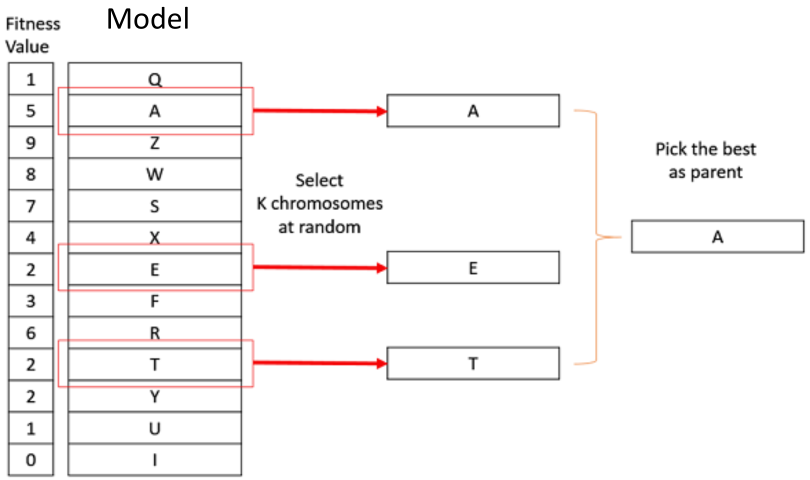
Slide courtesy of Xinnong Li, Ph.D. candidate UB

Sale M, Sherer EA.. Br J Clin Pharmacol. 2015 Jan;79(1):28-39

pyDarwin Genetic Algorithm (GA)

https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_parent_selection.htm
Sale M, Sherer EA.. Br J Clin Pharmacol. 2015 Jan;79(1):28-39
Ismail M, Sale M, Yu Y, et al. J Pharmacokinet Pharmacodyn. 2022 Apr;49(2):243-256.

Tournament Selection:



Crossover: Swap model information with probability $P_{crossover}$

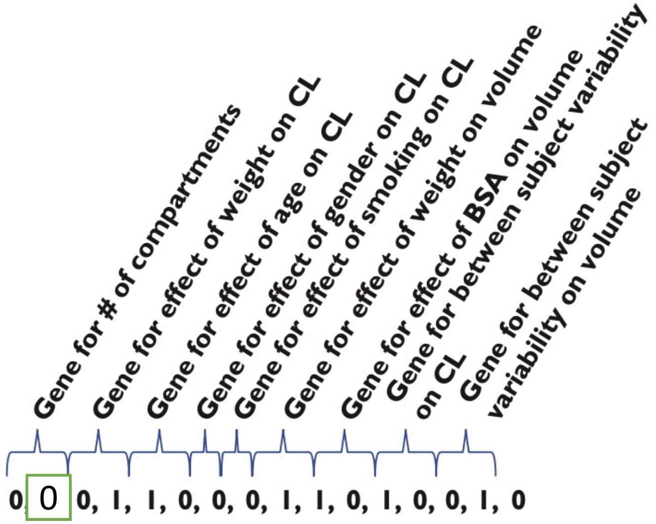
Parent Chromosomes

Model	Fitness	N_{CMT}	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

Children Chromosomes

Model	Fitness	N_{CMT}	Weight on CL	Weight on V	Age on CL	Age on V	Sex on CL	Sex on V	Error Model
800	94	2	Exponential	None	None	Linear	None	None	Combined
343	98	1	None	Linear	Exponential	None	Exponential	Linear	Proportional

Mutation: change model information with probability $P_{mutation}$



Slide courtesy of Xinnong Li, Ph.D. candidate UB

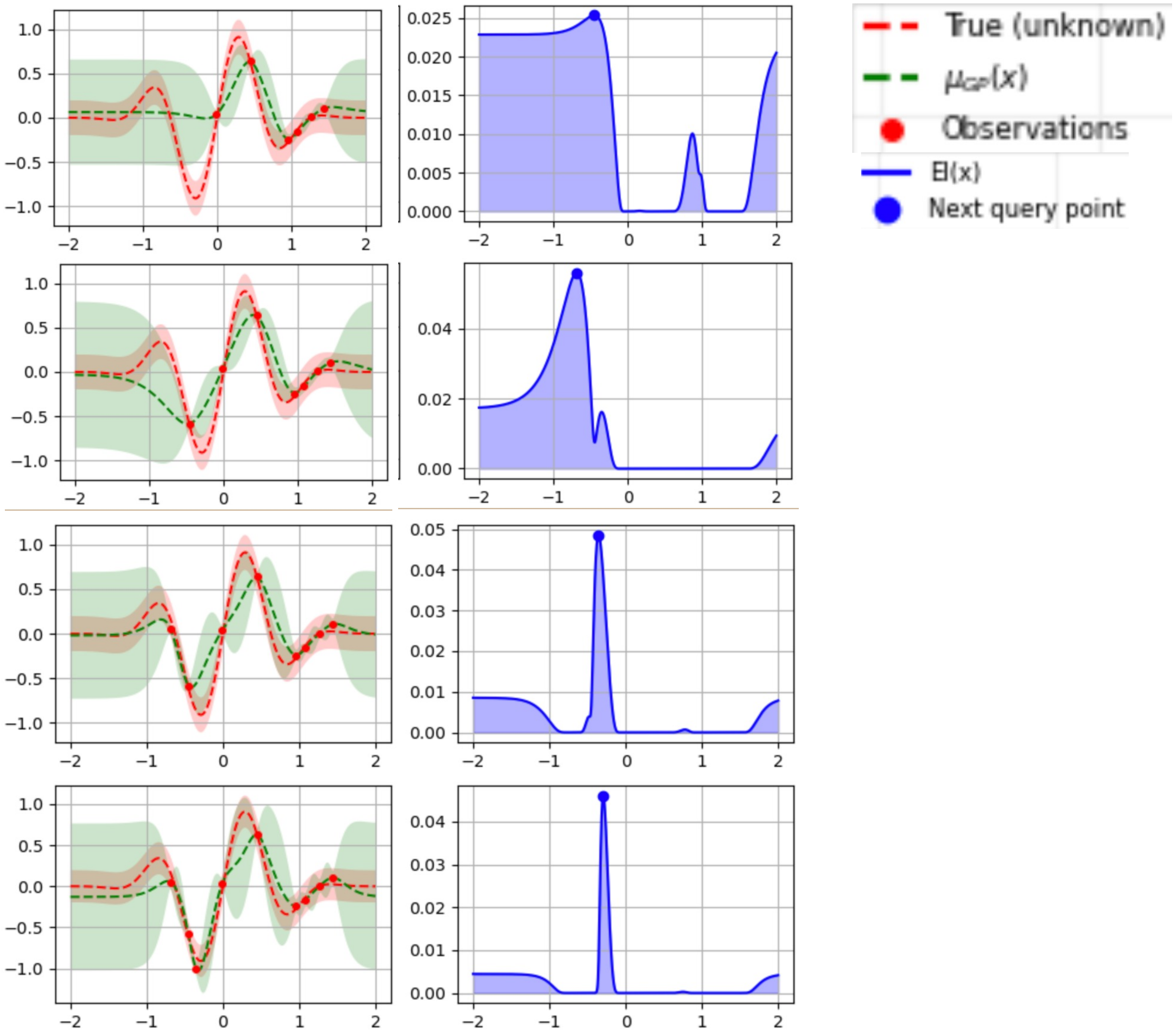
pyDarwin - Gaussian Process (GP)

Workflow:

- For the unknown objective function, we first treat it as a random function and place a **prior** over it. After getting some observations, the prior will be updated to the **posterior distribution**.
- Apply the **acquisition function** to choose the next query point. In this study, the acquisition function is $x_{t+1} = arg \max_{x} u(x)$, where $u(x)$ is equal to $EI(x) = \mathbb{E}[f(x) - f(x_t^+)]$
- Sample the next observation y_{t+1} at x_{t+1}
- Repeat the process until the final recommendation was made

a. True function and GP approximation function

b. Acquisition function



Slide courtesy of Xinnong Li, Ph.D. candidate UB

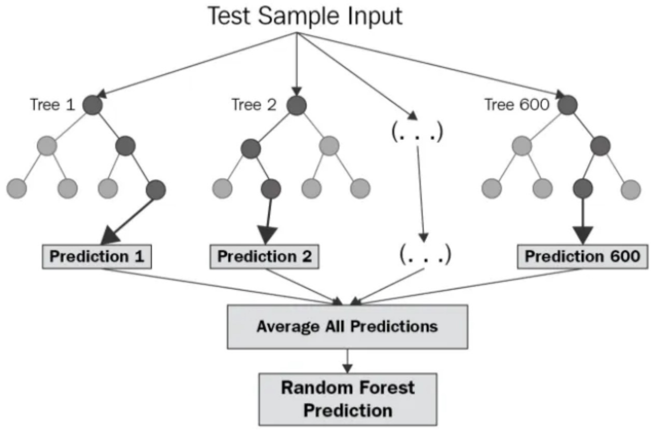
https://scikit-optimize.github.io/stable/auto_examples/bayesian-optimization.html

pyDarwin - Random Forest (RF)

Workflow (regression):

- **Bootstrapping** – randomly select models to generate trees
- **Aggregating** – split the trees by randomly picked features
- Run the record down each tree and do the **averaging** - get the averaged fitness value
- Find the next query point based on the acquisition function and update the ensemble of decision trees
- After N queries, the algorithm makes the final recommendation which represents the best estimate of optimizer

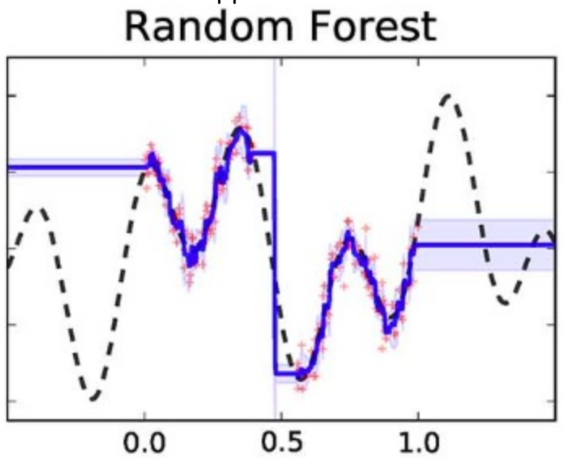
a. RF regression steps



b. Model Sample

	COM#	BOV	WT~CL	…
Model 1	1	Yes	No	…
Model 2	2	No	power	…
Model 3	1	No	No	…
⋮	⋮	⋮	⋮	…
Model n	1	Yes	exp	…

c. True function and its approximation with uncertainty



Slide courtesy of Xinnong Li, Ph.D. candidate UB

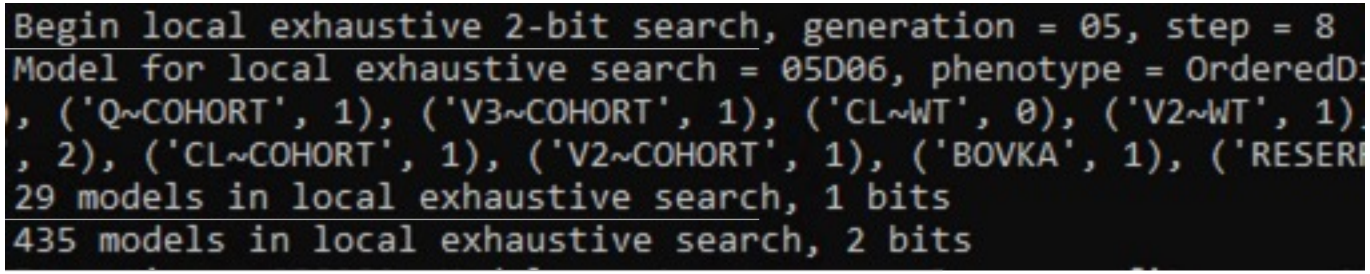
B. Shahriari, K. Swersky, Z. Wang, et al, in *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148-175, Jan. 2016.

https://levelup.gitconnected.com/random-forest-regression-209c0f354c84

pyDarwin - Local Downhill Search

- Local downhill search is implemented to ensure an efficient selection of the best possible models
- Do the **local-1-bit search** and **local-2-bit search** every 5 generations and at the end
- For N bits, local-1-bit search needs to run N models, local-2-bit search needs to run N*(N-1)/2 models
- Local-2-bit search is necessary to have confidence that true best model is discovered (still no guarantee)

a. Local downhill search appeared in the searching process



b. A chromosome example

Sale M, Ismail M, Wang F, et al. 2022. PAGE 30 (2022) Abstrt 10053

Slide courtesy of Xinnong Li, Ph.D. candidate UB

pyDarwin - Example

- Simulation Model:
- Linear 2 compartment, first order absorption (ADVAN4)
  - Typical Value (TV) for Clearance (CL) = 200 L/hr
  - TV for Central Volume (Vc) = 1000 L
  - TV for Ka of 2/hr,
  - TV k23 and k32 of 0.2/hr
  - TVALAG of 0.2/hr
  - Log normal between subject variance of 0.2 (all parameters)
- True covariates included:
  - CL~ (Weight, bilirubin, race and ALT)
  - Vc ~Weight
  - Ka~age
- Three additional covariates were included that did not influence the model

Sale et al, PAGE 2022

pyDarwin - Example

- Algorithms utilized:
  - Gaussian process/Bayesian Optimization (GP)
  - Random Forest (RF)
  - Gradient Boosted Random Tree (GBRT)
  - Genetic algorithm (GA)
  - Exhaustive search

pyDarwin - Example

- The search space for the model selection consisted of 10 dimensions:
- Number of compartment (1,2,3)
- Volume as a function of Weight (yes|no)
- Volume as a function of Sex (yes|no)
- Clearance as a function Weight (yes|no)
- Clearance as a function Age (yes|no)
- Between subject variability (BSV) on Ka (yes|no)
- K23/K32 (if present) as a function of Weight (yes|no)
- Absorption model (first order|zero order|combined zero, then first order) vs Absorption lag time (yes|no)
- BSV on zero order absorption or lag time, if present (yes|no)
- Residual error model (additive|proportional + additive)
- ~13000 candidate models

Sale et al, PAGE 2022

pyDarwin - Example

- The search criteria included:
    - Objective function value (OFV)
    - Parsimony penalty (10 points for each estimated parameter, THETA, OMEGA and SIGMA)
    - 100 point penalties for:
        - failing to converge
        - failing the covariance step
        - failing the correlation test
        - condition number > 1000

Sale et al, PAGE 2022

pyDarwin - Example

- Determination of “true optimal” model

- True optimal model (from exhaustive search):
  - two compartment
  - zero-order absorption
  - combined proportional + additive residual error
  - no covariates
  - Fitness: 4818

- Simulation model had:
  - higher (worse) reward than the “true” optimal model
  - failed the covariance step
  - incurred a 300 point penalty (fitness 5118)
    - 100 each for:
      - covariance
      - correlation
      - condition number

Sale et al, PAGE 2022

pyDarwin - Example

- All ML methods failed to identify the "true" optimal model
- All identified a 2-compartment model, without the zero-order infusion
- The 1-bit local search still did not result in the true optimal model (4922 fitness)
- A 2-bit search was implemented, where all combinations of 2-bit changes were examined
- This resulted in 120 new models (Table 1) (16 bits, [N^2-N)/2])
- Best fitness with 2-bit downhill search for all ML methods 4818

Sale et al, PAGE 2022

pyDarwin - Example

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	crash	crash	crash	crash	crash	crash	crash	crash	crash	crash	crash	crash	crash	crash	crash	crash
2		crash	5165.69	4827.70	crash	5032.32	5417.99	crash	4924.46	4923.49	5148.49	5167.12	4928.49	24018.29	4818.16	5776.74
3			4922.70	4928.74	crash	4922.60	4922.60	4924.50	4924.66	4924.39	5158.77	5128.77	5128.77	6716.45	4918.22	5666.87
4				4932.68	crash	4932.58	4932.58	crash	4934.63	crash	5178.74	5138.74	5138.74	5144.30	4927.68	5647.75
5					4918.77	crash	crash	crash	crash	crash	crash	crash	crash	crash	crash	crash
6						99999.00	5032.60	4925.97	4926.18	4928.11	5152.60	5132.60	5132.60	crash	4922.13	5670.00
7							5032.64	4925.97	4926.18	4928.11	5152.60	5132.60	5132.60	crash	4922.13	5670.00
8								4926.09	crash	4930.31	5154.50	crash	5134.50	crash	4923.91	5669.31
9									4926.30	4930.41	crash	5134.66	5134.66	5310.48	4924.08	5669.31
10										4928.20	5154.40	5134.39	5134.40	5548.57	4923.87	5674.41
11											5152.70	5138.77	crash	5441.17	5156.14	6006.24
12												5132.70	crash	32929.81	5128.22	5976.87
13													5132.70	5137.88	4926.14	5776.24
14														5315.49	crash	6233.71
15															4922.22	5662.69
16																5670.06

Sale et al, PAGE 2022

pyDarwin - practicalities

```
$INPUT ID TIME ADDL OCC II DV EVID MDV AMT ORAL RACE COHORT SEX LASTNEG FIRSTPOS
$DATA {data_dir}/finalm6_3.CSV IGNORE=#
$SUBROUTINE {ADVAN[1]}

$PK
   CWTKG = WEIGHT/76.6
   CBMI = BMI/25.4
   CWTKGZERO = WEIGHT - 76.6
   BMIZERO = BMI-25.4

   {BOVKA[1]}

   IF (OCC.EQ.1) THEN
   KA=24*THETA(5)*EXP(ETA(5))
   F1=THETA(4)*EXP(ETA(4))
   ENDIF


   IF (OCC.GE.2) THEN
   KA=24*EXP(ETA(3))*THETA(3){KA~BMI[1]} {KA~COHORT[1]} {KA~WT[1]} *EXP(BOVKA)
   F1=1
   ENDIF


   TVCL=24*THETA(1){CL~WT[1]}{CL~BMI[1]}{CL~COHORT[1]}
   CL=TVCL*EXP(ETA(1))
   TVV2=THETA(2) {V2~WT[1]}{V2~BMI[1]}{V2~COHORT[1]}
   V2=TVV2*EXP(ETA(2))
   K=CL/V2

   {ADVAN[2]}

   S2=V2
          ]
```

Option file

- ML algorithm
- Penalty terms
- Population size
- Number of generations
- etc...

Token file

- Features to test
- Different options in one feature

Template file

- Similar to the control stream in NONMEM

https://certara.github.io/pyDarwin/html/Overview.html

Slide courtesy of Xinnong Li, Ph.D. candidate UB

pyDarwin - practicalities

- Options file

- specifies
  - Author
  - Algorithm
  - Population size
  - Parallelprocesses
  - # per generation
  - # of generations
  - Penalties
  - nmfe??.bat path
  - Timeout
  - Post processing (python/R code for additional penalties etc.)

```json
{
    "author": "Certara",
    "algorithm": "EX",
    "exhaustive_batch_size": 100,
    "population_size": 8,
    "num_parallel": 4,
    "crash_value": 99999999,

    "penalty": {
        "theta": 10,
        "omega": 10,
        "sigma": 10,
        "convergence": 100,
        "covariance": 100,
        "correlation": 100,
        "condition_number": 100,
        "non_influential_tokens": 0.00001
    },

    "remove_run_dir": false,

    "nmfe_path": "c:/nm744/util/nmfe74.bat",
    "model_run_timeout": 1200
}
```

pyDarwin - practicalities

- Template file
- Specifies
  - Control stream
  - Model structure
  - Location of swappable tokens

```
$PROBLEM    2 compartment fitting
$INPUT       ID TIME AMT DV WTKG GENDER AGE DROP
$DATA      {data_dir}/dataExample1.csv IGNORE=@

$SUBROUTINE ADVAN2
$ABBR DERIV2=NO
$PK
  CWTKG = WTKG/70  ;; CENTERED ON ONE
  CAGE = AGE/40
;; thetas out of sequence
  TVV2=THETA(2){V2~WT[1]} {V2~GENDER[1]}
  V2=TVV2*EXP(ETA(2))
  TVCL= THETA(1) {CL~WT[1]}
  CL=TVCL*EXP(ETA(1))
  K=CL/V2
  TVKA=THETA(3)
  KA=TVKA  {KAETA[1]}
  S2    = V2/1000
  {ALAG[1]}
$ERROR
 REP = IREP
 IPRED =F
 IOBS = F {RESERR[1]}
 Y=IOBS
$THETA  ;; must be one THETA per line.
 (0.001,100)    ; THETA(1) CL UNITS =  L/HR
 (0.001,500)    ; THETA(2) V  UNITS = L
 (0.001,2)      ; THETA(3) KA UNITS = 1/HR

 {V2~WT[2]}   ;;; comment must consist of more than one word
 {V2~GENDER[2]}   ;;; otherwise it's a definition, and it will push you back
 {CL~WT[2]}
 {ALAG[2]}
```

pyDarwin - practicalities

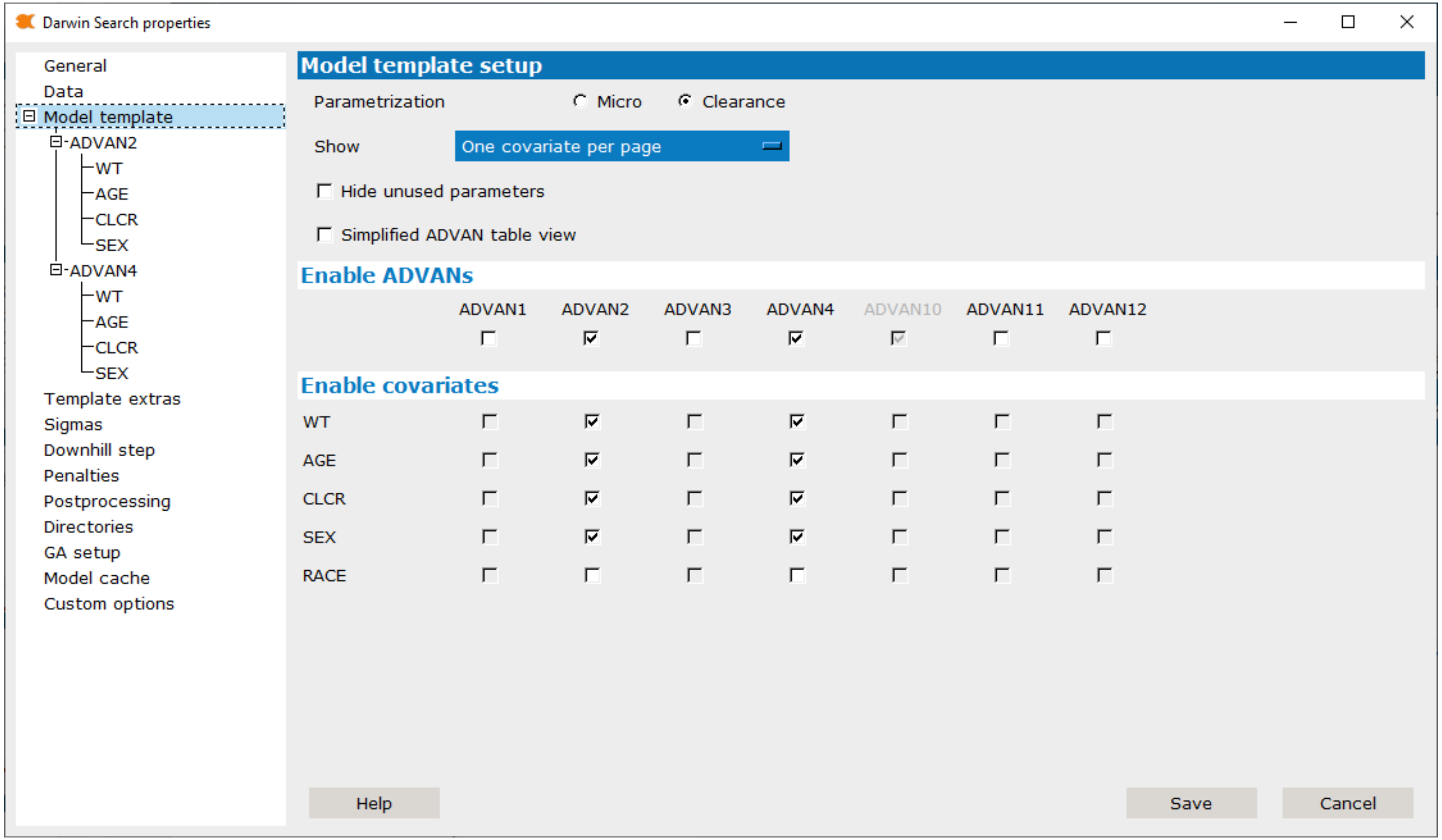
- Tokens file
- Specifies
  - elements to be substituted into tokens for each of the options selected
  - Each token is named and sequenced using the json format

```
{
    "V2~WT": [
            ["*CWTKG**THETA(V2~WT)",
                    "  (-4,0.8,4) \t; THETA(V2~WT) POWER volume ~WT "
            ],
            ["",
             ""
            ]

    ],

    "V2~GENDER": [
            ["",
                    ""
            ],["*EXP(GENDER*THETA(V2~GENDER))",
                    "  (-4,0.1,4) \t; THETA(V2~GENDER) exponential volume~GENDER "
            ]

    ],
    "CL~WT": [
            ["*CWTKG**THETA(CL~WT)",
                    "  (-4,.7,4) \t; THETA(CL~WT) POWER clearance~WT "
            ],
            ["",
                    ""
            ]
    ],
  "KAETA": [
            ["*EXP(ETA(KAETA)) ",
                    "$OMEGA ;; 2nd??OMEGA block \n  0.1\t\t; ETA(KAETA) ETA ON KA"
            ],
            ["",
             ""
            ]

    ],
```

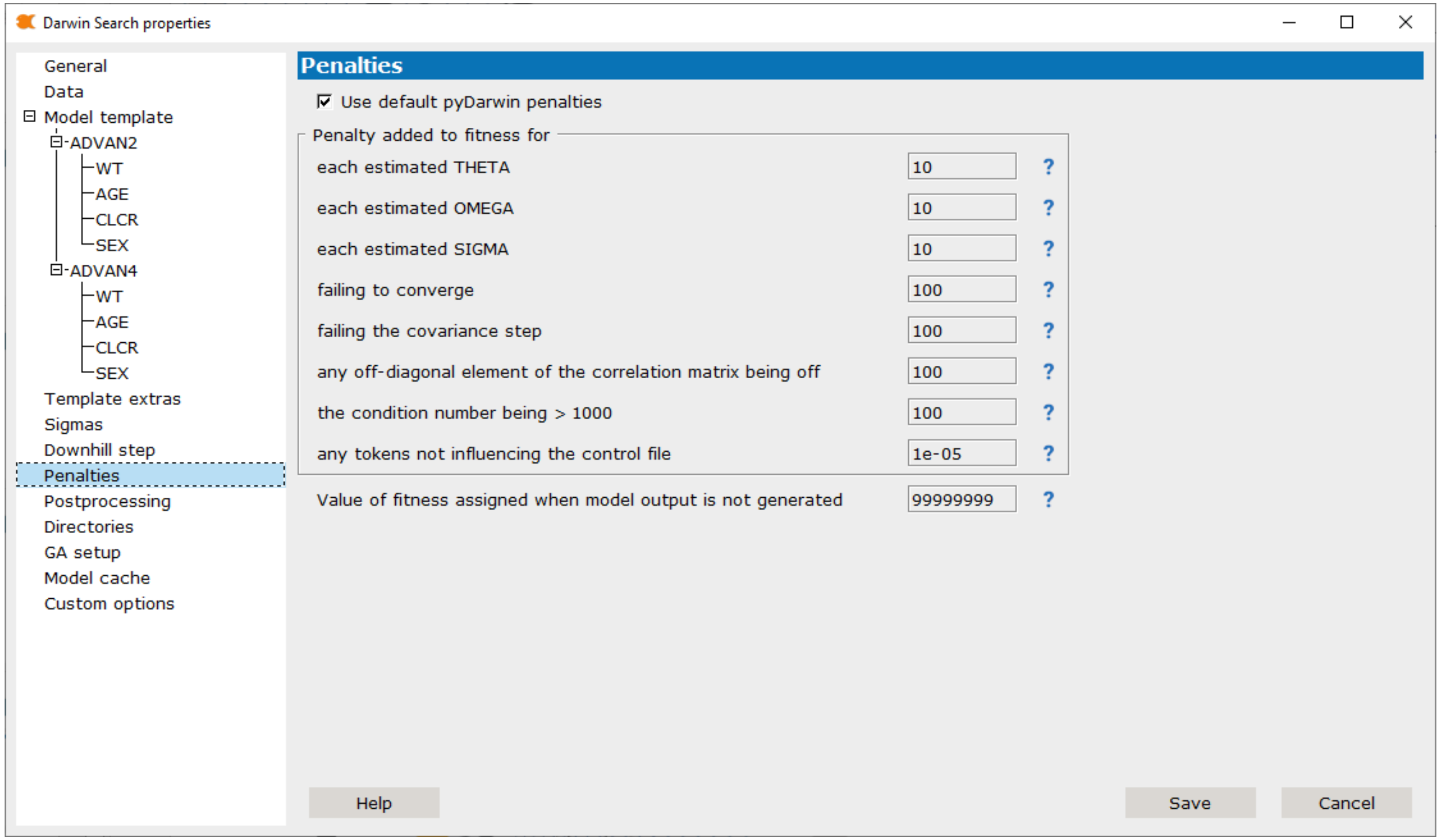
```
$PROBLEM    2 compartment fitting
$INPUT       ID TIME AMT DV WTKG GENDER AGE DROP
$DATA      {data_dir}/dataExample1.csv IGNORE=@

$SUBROUTINE ADVAN2
$ABBR DERIV2=NO
$PK
  CWTKG = WTKG/70  ;; CENTERED ON ONE
  CAGE = AGE/40
;; thetas out of sequence
  TVV2=THETA(2){V2~WT[1]} {V2~GENDER[1]}
  V2=TVV2*EXP(ETA(2))
  TVCL= THETA(1) {CL~WT[1]}
  CL=TVCL*EXP(ETA(1))
  K=CL/V2
  TVKA=THETA(3)
  KA=TVKA  {KAETA[1]}
  S2    = V2/1000
  {ALAG[1]}
$ERROR
 REP = IREP
 IPRED =F
 IOBS = F {RESERR[1]}
 Y=IOBS
$THETA  ;; must be one THETA per line.
 (0.001,100)    ; THETA(1) CL UNITS =  L/HR
 (0.001,500)    ; THETA(2) V  UNITS = L
 (0.001,2)      ; THETA(3) KA UNITS = 1/HR

 {V2~WT[2]}   ;;; comment must consist of more than one word
 {V2~GENDER[2]}   ;;; otherwise it's a definition, and it will push you back
 {CL~WT[2]}
 {ALAG[2]}
```

```
{
   "V2~WT": [
      ["*CWTKG**THETA(V2~WT)",
         "  (-4,0.8,4) \t; THETA(V2~WT) POWER volume ~WT "
      ],
      ["",
       ""
      ]
   ],
   "V2~GENDER": [
      ["",
         ""
      ], ["*EXP(GENDER*THETA(V2~GENDER))",
         "  (-4,0.1,4) \t; THETA(V2~GENDER) exponential volume~GENDER "
      ]
   ],
   "CL~WT": [
      ["*CWTKG**THETA(CL~WT)",
         "  (-4,.7,4) \t; THETA(CL~WT) POWER clearance~WT "
      ],
      ["",
         ""
      ]
   ],
 "KAETA": [
      ["*EXP(ETA(KAETA)) ",
         "$OMEGA ;; 2nd??OMEGA block \n  0.1\t\t; ETA(KAETA) ETA ON KA"
      ],
      ["",
       ""
      ]
   ],
```

Select structural models and which covariates to include

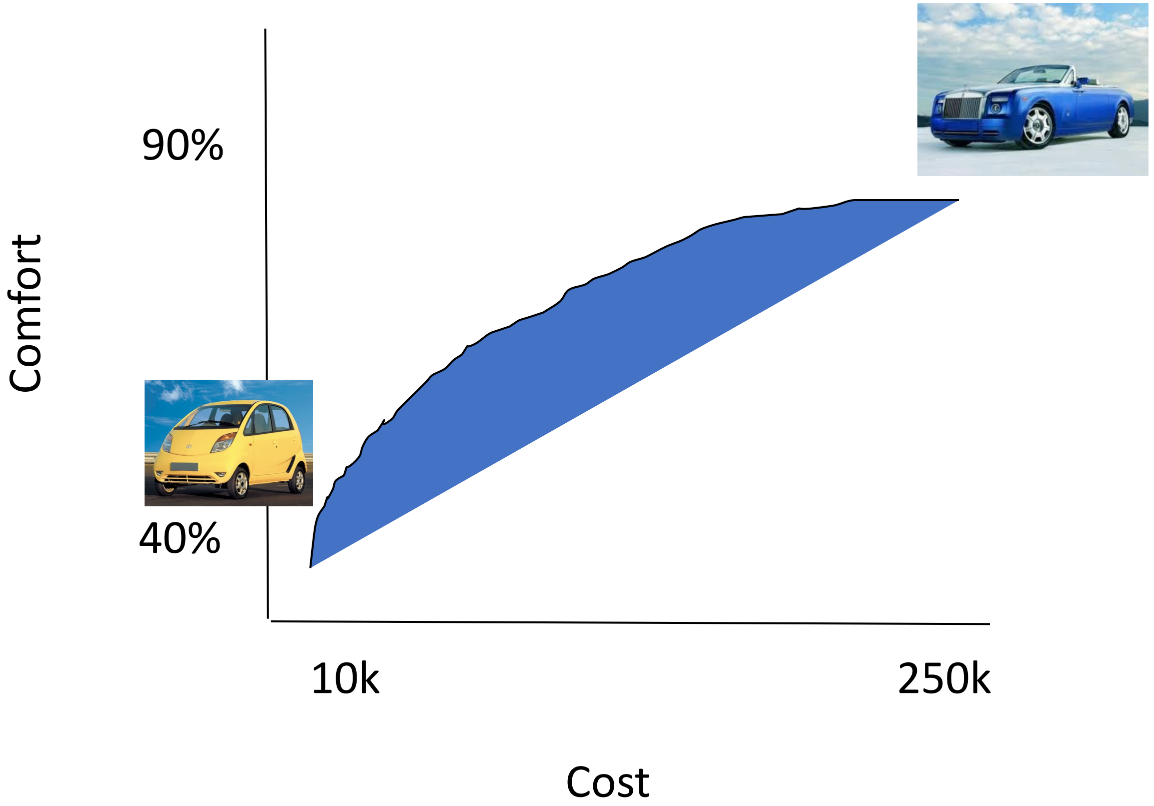


Penalty/Fitness/Cost function



Multi-objective GA optimization

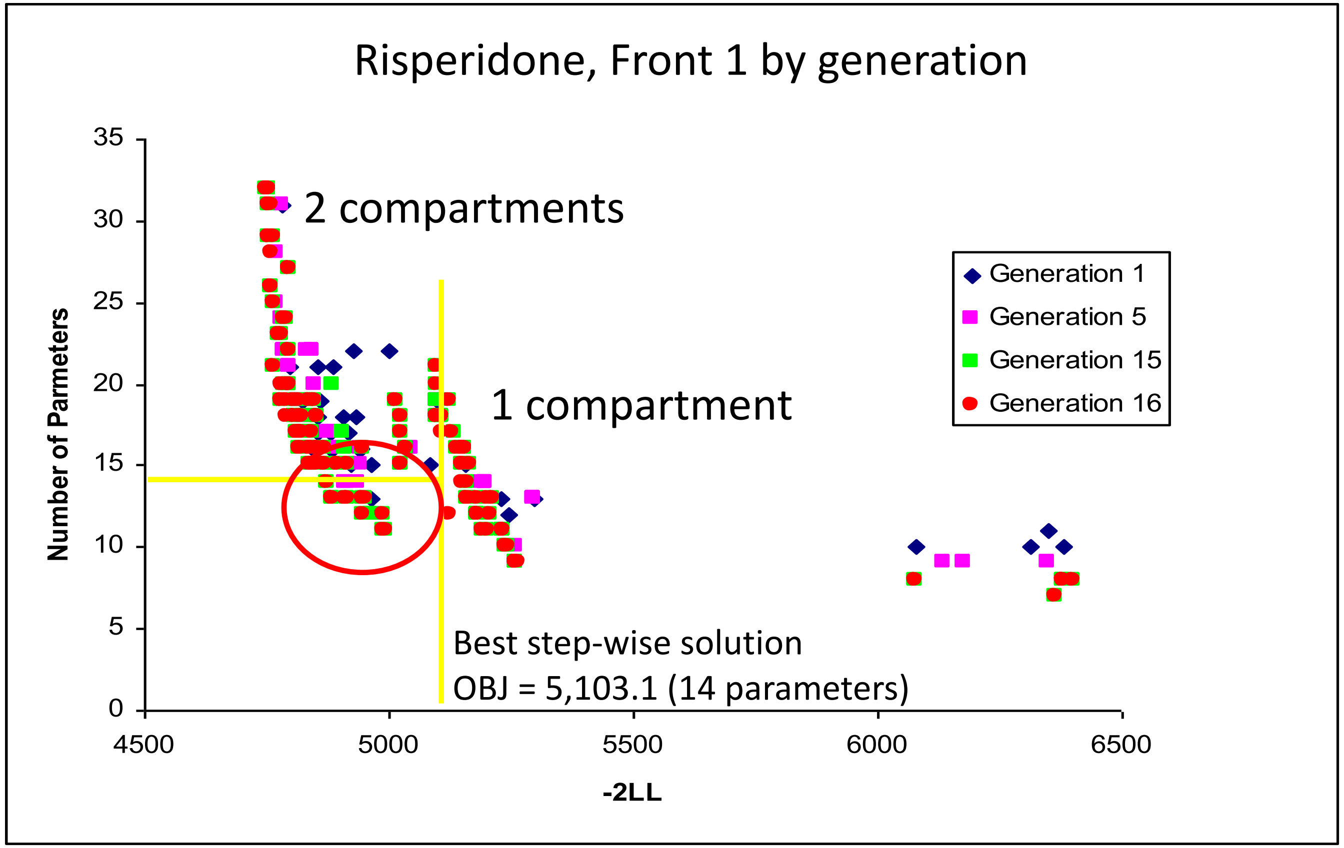
- Optimize over many criteria



- Decisionmakers
  - preferred not to be presented with a single “best” option

Multi-objective genetic algorithm

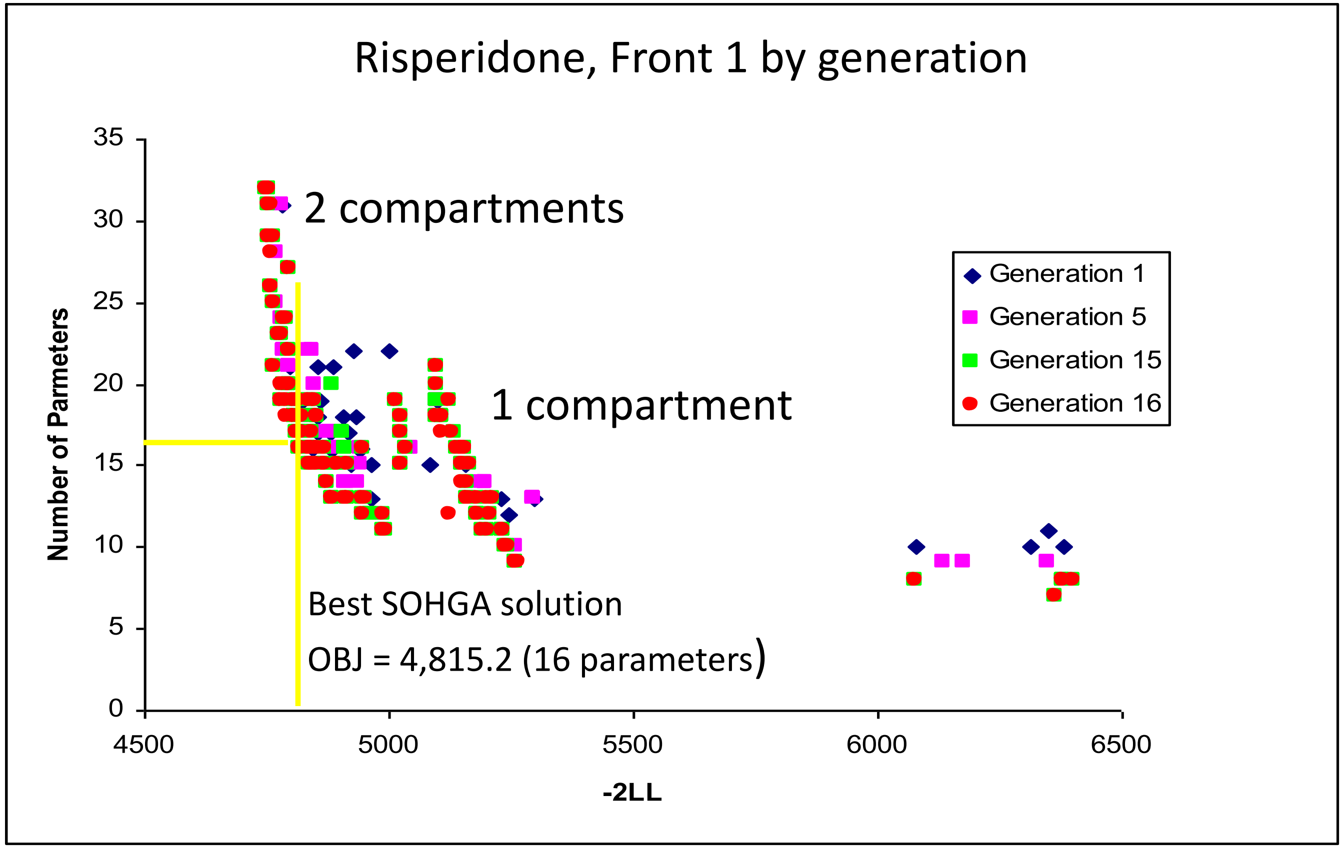
- Front in -2xLL vs. # parameters space



Sherer et al PAGE 2012

Multi-objective genetic algorithm

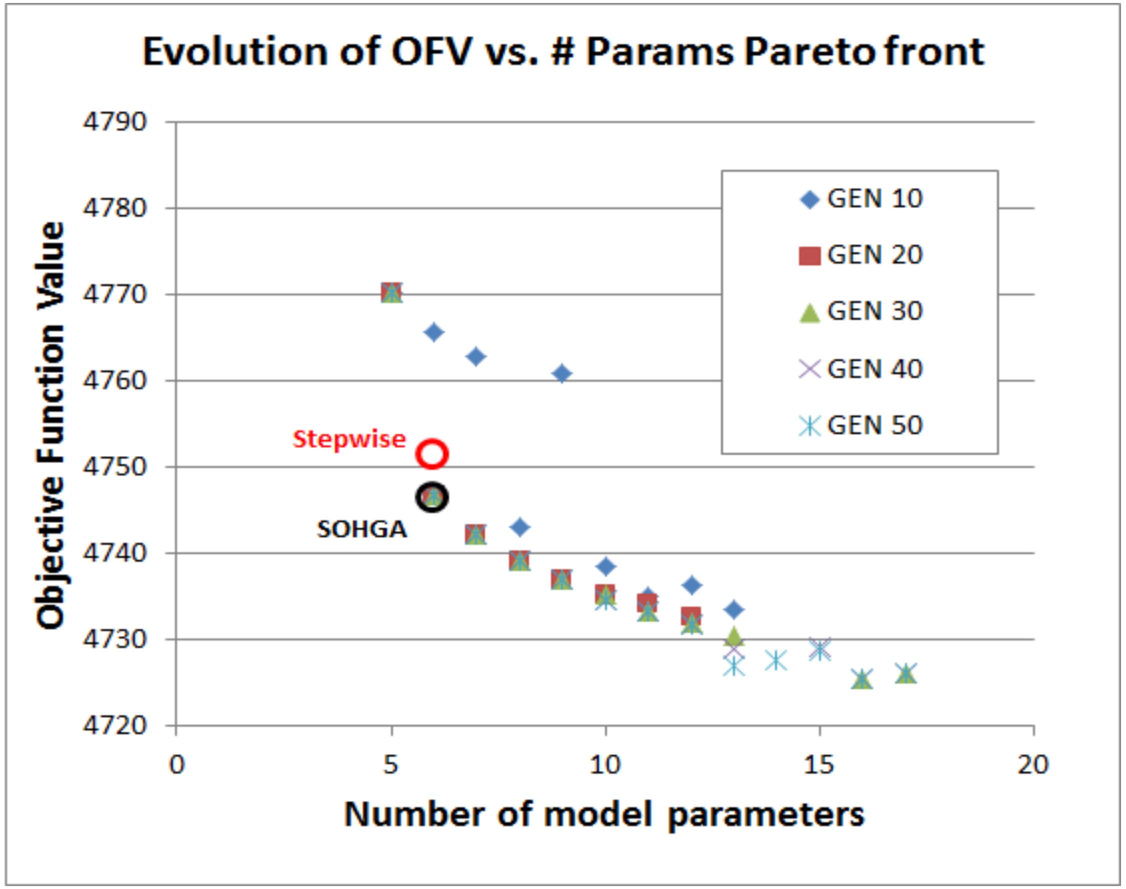
- Front in -2xLL vs. # parameters space



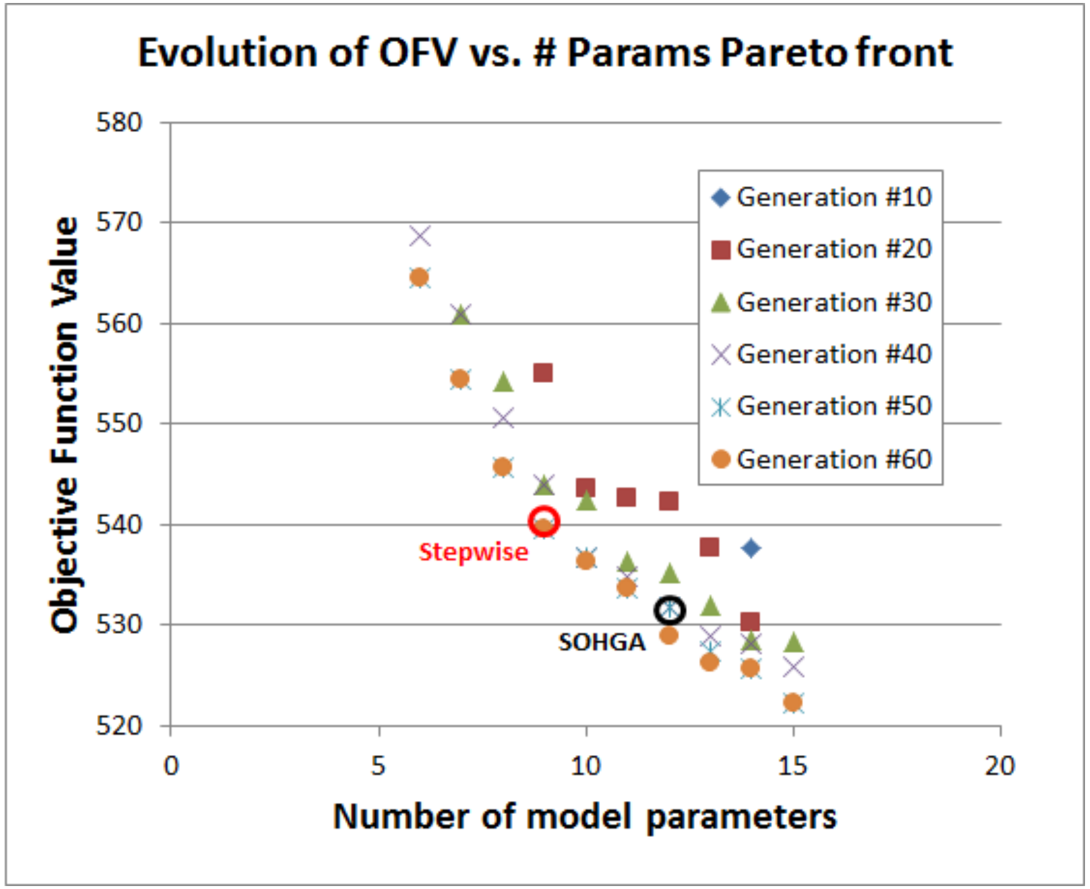
Sherer et al PAGE 2012

Multi-objective genetic algorithm

Ziprasidone

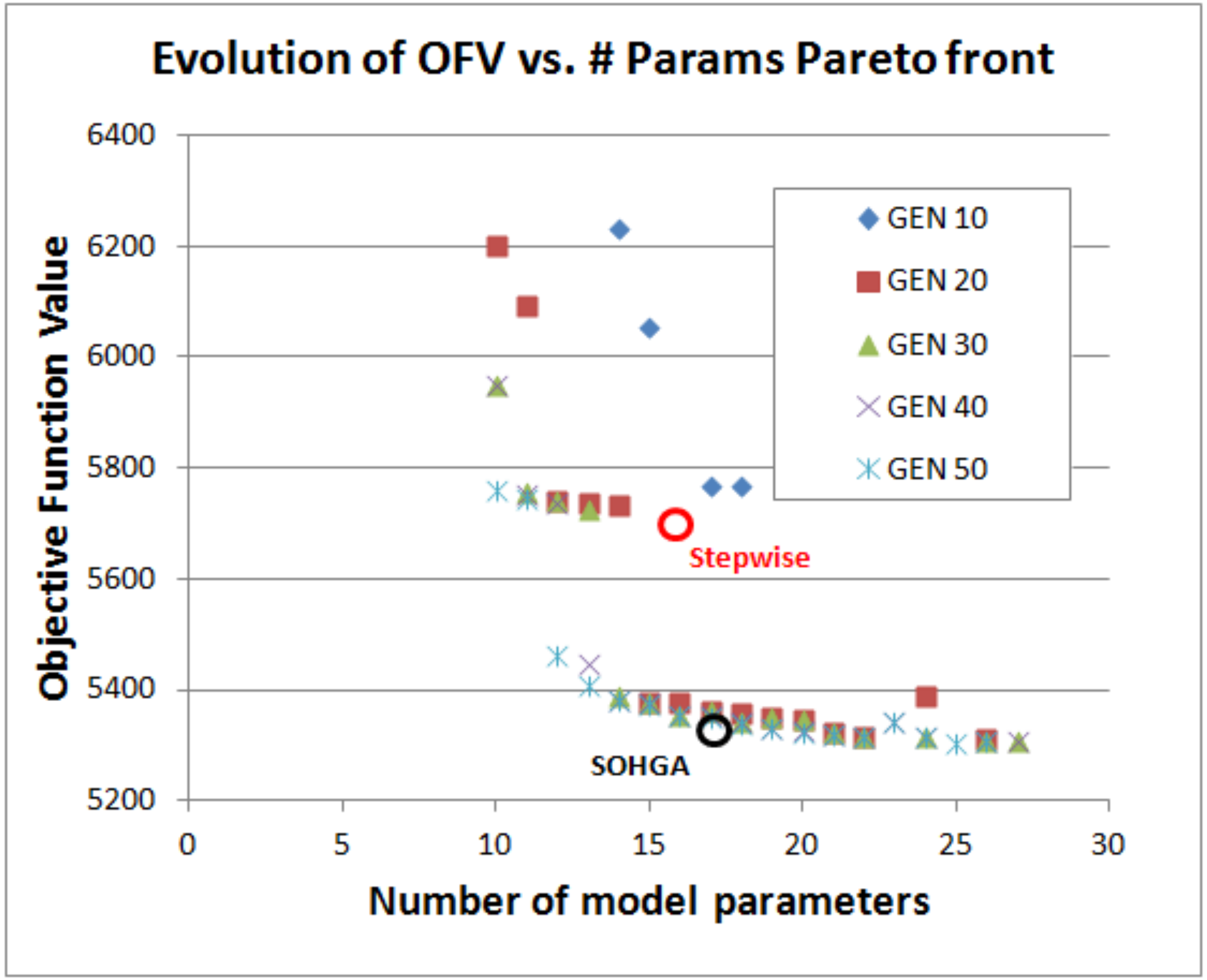


Perphenazine



Sherer et al PAGE 2012

Multi-objective genetic algorithm – IV
Citalopram



Sherer et al PAGE 2012

Conclusions

- ML methods typically selected the better models vs stepwise selection (human driven)
- One bit downhill search is not always sufficient to discover the numerically optimal model
- Multi-objective optimization (GA) provides insight into non-dominated solutions for specific metrics

Acknowledgements

- Certara
  - Mark Sale M.D.
  - James Craig M.S.
  - Keith Nieforth Pharm.D.
  - Pavel Polozov
- Indiana University
  - Eric Sherer Ph.D.
- University at Buffalo
  - Mohamed Ismail Ph.D. (ePD)
  - Xinnong Li M.S.
  - Sihang Liu Ph.D. (BMS)
  - Nikhil Pillai Ph.D. (Sanofi)
- FDA
  - Liang Zhou Ph.D.
  - Fenggong Wang Ph.D
  - Kairui Feng Ph.D.
  - Meng Hu Ph.D.
- FDA grant (U01FD007355)