Technical Issues in KinGUI II

Zhenglei Gao

May 5, 2011

Problems to solve

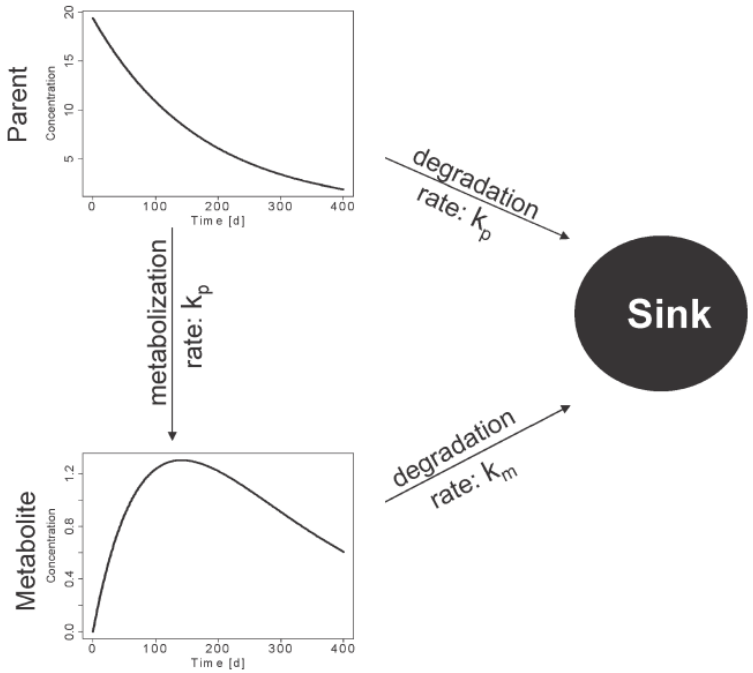

Figure: Illustration of a very simple a 2-compartment model with first order reactions

The Model:

$$y_{1j} = M_0 e^{-k_p t_i} + \epsilon_{1j}$$

$$y_{2j} = cM_0 k_p \frac{e^{-k_m t_i} - e^{-k_p t_i}}{k_p - k_m} + \epsilon_{2j}$$

where, M_0 is the initial concentration of Parent substance; k_p is the degradation rate of Parent; k_m is the degradation rate of Metabolite; c is the formation fraction from Parent to Metabolite; $j = 1, \cdots, n$.

- Interested in the parameter($\theta = (k_p, k_m, c, M_0)$ in the simple case) estimations and the confidence intervals.

Backgrouds for the development of KinGUI II

- Technical Reasons
  - MATLAB implementation is difficult to disseminate because it is fixed to an old version of the runtime environment.
  - KINGUI suffers from various, serious bugs which need to be resolved
    - Negative formation rates when 3 or more metabolites are formed from one precursor.
    - Mixing up of formation rate values in some cases.
- Scientific Reasons
  - Confidence interval estimation methods implemented in KinGUI inherently assumes constant error variance across all species, which is neither realistic nor met by out data.
  - MCMC methods are not implemented.

Available Tools

- 'optimx' and related other optimization packages.
- 'FME' and related packages - for flexible model fitting.
- 'mkin'- does kinetic evaluations similar as KinGUI. The author Johannes Ranke is a collabrator of KinGUI II.
- 'deSolve'- ODE solver.

New Implementation

- Two new optional methods
  - Iteratively Reweighted Least Squares(IRLS)
  - MCMC
- Codes developed specially for the GUI users.
  - Reduandant for normal R users
  - Gui version functions using different parameterizations with 0-1 constraints.

Available R optimization algorithms

- **"Marq"** of nlm.ls from 'nlm.ls'
- **"Port"** of nlminb
- "Newton" of nlm
- "Nelder-Mead", "BFGS", "CG","**L-BFGS-B**", "SANN" of optim
- "Pseudo" of 'pseudoOptim' from
- **"spg"**,"Rcvmin","R.." of 'optimx' from 'optimx'
- "bobyqa", "newuoa", "uobyqa" from 'minqa'
- "Better Nelder-Mead" of 'nmk' from 'dfoptim'

* The bold ones are the ones I selected by instinction to try out.

Major Issues in KinGUI II

- Major Technical Issues
  - Missing global minimum
  - Lower Efficiency
  - ODE solver
- 'Unrealistic' user requirement, otherwise multiple starting values could be tried.
  - Random starting values
  - One (stable) solution for all problems

ODE solver

- 'lsoda' vs. 'ode45'
  - With the same tolerance lever, they can reach different solutions
  - 'lsoda' is in general faster but sometimes stops doing integration while 'ode45' gives a solution in the same settings. Error message: *Returning early. Results are accurate, as far as they go < − repeated convergence test failures on a step, but integration was successful-inaccurate Jacobian matrix?*
- Current Solution
  - 'lsoda'as the default
  - If integration not successful, try 'ode45'
  - Otherwise, set model cost Inf to discard the current propose of parameter values.

Efficiency

- Possible Reasons
  - Single function evaluation time
    - Transformation of the parameters due to the box constraints.
    - More than one calculation in the modCost function.
  - Number of function evaluations
- Current Solution: accept 'spg' as the default optimizer.
- Next Steps
  - **Provide gradient for the objective function**
  - Instead of calculating all the needed values using 'modCost', calculate the needed ones for corresponding optimization algorithms used.
  - Better algorithms?
    - Try ", ", and ".
    - Translating the matlab 'lsqnonlin'

Local vs. Global Minimum

- KinGui default optimization algorithm: trust-region-reflective
  - With random starting values(far from the best fit ones) performs well
  - relatively fast enough.
- R optimization can easily miss the minimum. 'L-BFGS-B','spg' seems better in the complex cases but much worse in the simple cases. (very strange)
- Current Solutions
  - none! Manually it could be done by multiple starting values, trying different algorithm, but the GUI version requirement is simple and stable.

Other Issues

- Different algorithms reach different solutions.
  - Data is too scattered or we did not choose a sensible model to describe the data-generating process.

A Simple Case(Folder 'BCS1', bcs1.r)

- 2-compartment model with one parent and one metabolite
- Problems:
  - 'bobyqa','spg' methods need much more function evaluations.
  - 'bobyqa','L-BFGS-B' and 'spg'does not find the minimum with the same starting values as in 'Marq'.
  - 'Marq' can get stuck in a local minimum in such a simple case!! (in BCS1.r, choose starting value k(parent)=0.0058)
  - **without solving ODE, using 'gnls' in 'nlme' elapsed time is almost 0.**
- Performance on my laptop(using IRLS, quiet=FAISE, with intermediate output):

	Marq	L-BFGS-B	spg	Port	bobyqa
elapsed	41.15	824.53	6793.51	173.83	160.48

A 6-compartment Model(Folder 'BCS2', bcs2.r)

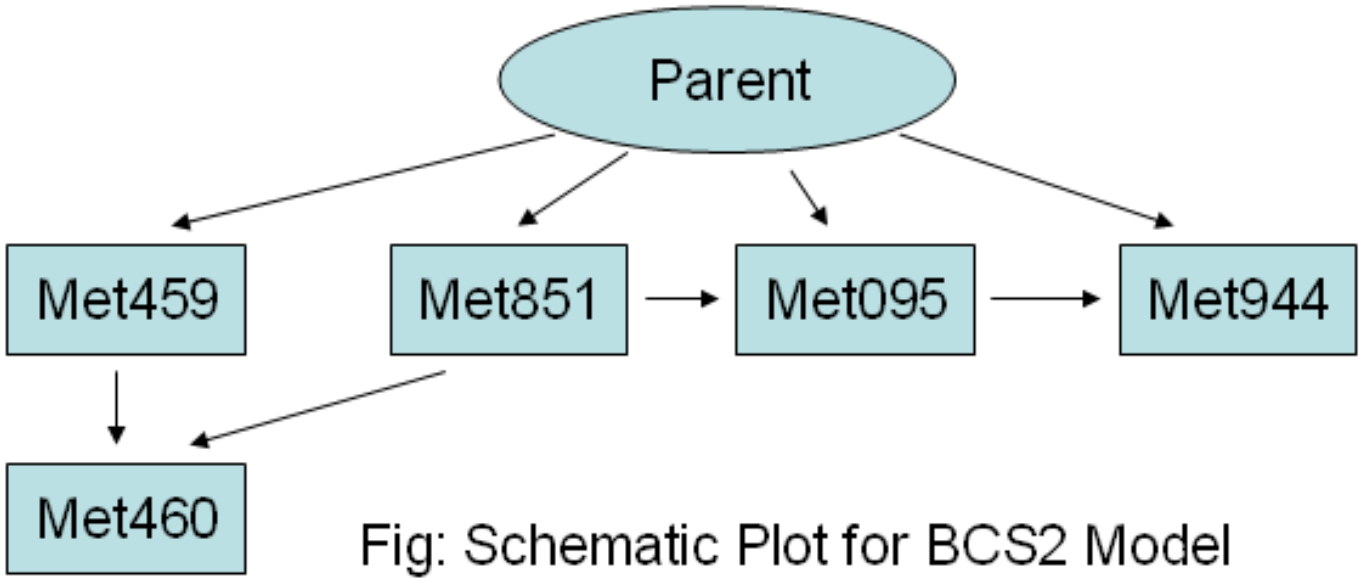

Fig: Schematic Plot for BCS2 Model

- Settings: with default start values same as in the Matlab code settings.

A 6-compartment Model(Folder 'BCS2', bcs2.r)

- With default start values same as in the Matlab code settings.
  - Levenberg-Marquardt: stuck in local minima. (Non-finite(or null) value for a parmeter specified in nls.lm)
  - L-BFGS-B: satisfactory result but very slow.
- R Files: Folder 'BCS2', ex2.r.

BCS3(Folder 'BCS3', ex3.r)

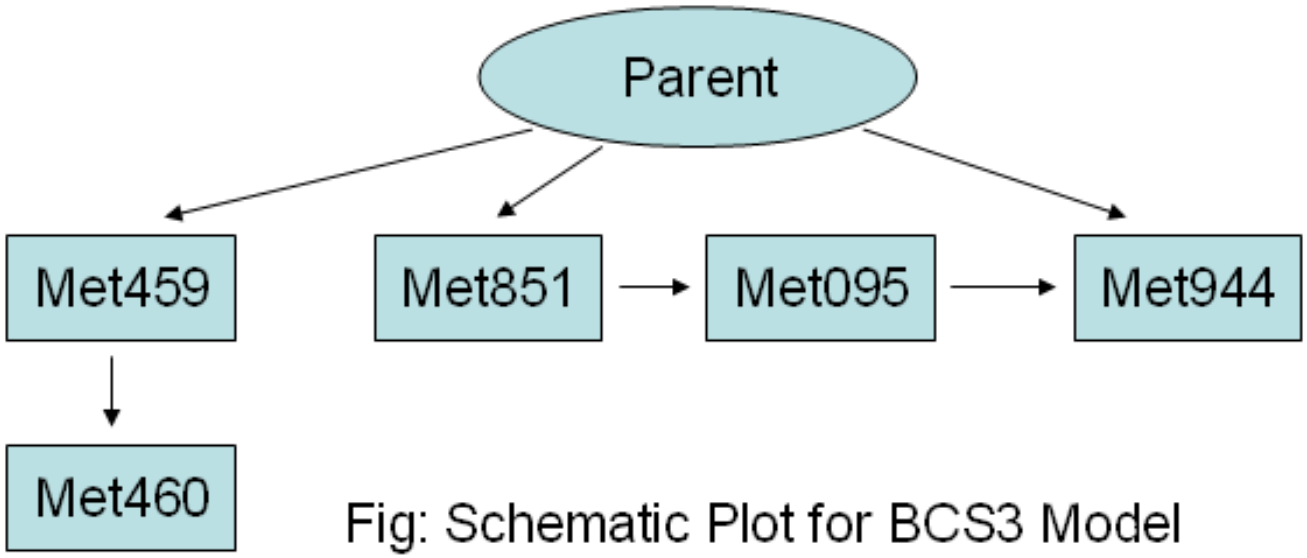

Fig: Schematic Plot for BCS3 Model

- Settings: good starting values obtained from KinGUI

BCS3(Folder 'BCS3', ex3.r)

- Single function evaluation time:
  - user:0.4; system:0.0; elapsed:0.4
  - Average(I set trace=3 in optimx): 0.4137,0.0004,0.4143.
  - Reported by spg: xtimes=769.03/fns=5.8917
- Number of function evaluations(in one iteration, to be comparable to KinGUI):
  - spg: Reported by fns:129; Actual function calls: 2744; niter=101.
  - L-BFGS-B: **NA because of line search in bad direction**(needs finite value of fn which I set at 1e20)
  - Levenberg-Marquardt(faster than spg): niter=1; actuall function calls:49
- Problems
  - spg: hessian not invertable, cannot be used to obtain confidence intervals(some of the parameter estimations are on the boundaries.)

BCS4(Folder 'BCS4', ex4.r)

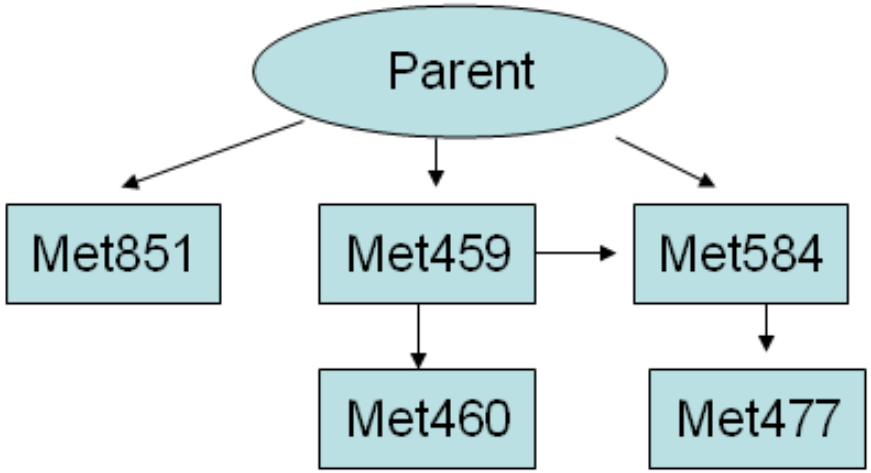

Fig: Schematic Plot for BCS4 Model

- Settings: good starting values obtained from KinGUI

BCS4(Folder 'BCS4', ex4.r)

- **Cannot estimate covariance from hessian even changing to L-M methods after the first iteration.**
- Different algorithm gives different results.

BCS5

- ...

Package

- Including two sets of functions with file names '*.r' and '*.gui.r'. R functions in '*.gui.r' are written for the GUI.
- The codes are far from clean.
- Help files(.Rd) will be provided later.
- ...

Line Search and Trust Region

- two global strategies that modify the normal locally convergent algorithm
- Line search choose the direction first, then the step length.
- Trust region choose the maximum step length first, then the direction.
  - Levenberg-Marquardt is the first trust-region method developed. Why it does not find the 'global minimum'as the other line search algorithms do?

Other Issues

- How to deal with 'ghost' compartment like in SLV