Fitting data to models in Pmetrics

Jason Roberts
The University of Queensland, Australia

BIG acknowledgement for Michael Neely
Importance of a plan

- Reproducible and consistent work plan
  - i.e. the order you perform your modelling
  - What should you do at each stage?
Define purpose

• Why are you developing your model?
• What function do you expect/desire for it to have?
• Who do you need to present it to?
• Will you test how good it is?
Gather/prepare data

Likelihood of being correct after your first completing data entry?
Define structural model

• Know your data
• Look at your data
• 1c, 2c, 3c, linear, non-linear? etc.
• Be prepared to test all plausible models and select that which performs statistically and is biologically plausible
• Try different error models (gamma and lambda) and compare performance
1 compartment model

#Pri
Cl,0.1,60
V,10,80

#Sec
Ke=Cl/V

#Out
Y(1) = X(1)/V

#Err
G=1
0.5,1,0,0!
2 compartment model

#Pri
Cl, 0.1, 60
V, 10, 80
Kcp, 0.001, 30
Kpc, 0.001, 60

#Sec
Ke = Cl / V

#Dif
XP(1) = RateIV(1) - Ke * X(1) - Kcp * X(1) + Kpc * X(2)
XP(2) = Kcp * X(1) - Kpc * X(2)

#Out
Y(1) = X(1) / V

#Err
G = 1
0.5, 1, 0, 0!
Other model syntax

• Bolus inputs - #Bol
• Initial conditions – #Ini
  \[ X(2) = IC \times V \] (where IC is a measured value with #Cov)
• Bioavailability - #F
• Lag time - #Lag
Error models

#ERR

Gamma → error = SD * gamma  (e.g. values 1-10)

Lambda → error = (SD² + lambda²) * 0.5
(nb SD is std deviation of each observation)

ERRrun() – helps estimate assay coefficients
General workflow

1. In your computer’s file browser (i.e. Windows Explorer or Mac Finder), copy the model and data file to the project /Runs folder.

2. In R/Rstudio, use `setwd()` to ensure that your current working directory is the project /Runs folder.

3. Execute `NPrun()` or `ITrun()` with the appropriate arguments.
NPAG Start
Iteratively Choose Best Points

Fit the model to the data from subjects

NPrun(model="model.txt", data="data.csv")
End
A non-parametric population model

plot(final.1)
NPAG runs

NPrun(model = "model.txt", data = "data.csv", run, include, exclude, ode = -4, tol = 0.01, salt, cycles = 100, indpts, icen = "median", aucint, idelta = 12, prior, overwrite = F, nocheck = F, parallel = NA)
NPrun: model

- The quoted name of a model file in your working directory, e.g. `model="model2.txt"`

- If your model filename is the default, “model.txt”, you can omit

- You can also specify model as the number of a previous run. Pmetrics will copy that model for you to the current working directory, e.g. `model=3`
Documentation

• Record what you do and why you do it and the order you do it

• Ensure you can justify selecting one model over another – various criteria for this...
NPrun: data

- The quoted name of a data file in your working directory, e.g. \texttt{data=“data2.csv”}

- If your data filename is the default, “data.csv”, you can omit

- You can also specify data as the number of a previous run. Pmetrics will copy that data file for you to the current working directory, e.g. \texttt{data=3}
NPrun: run

- Optional argument not normally needed

- Specify as a number to force a run to be numbered as such, e.g. \texttt{run=4} will result in a folder labeled “4” with the results of the run

- If the folder specified by \texttt{run} exists already, for safety, you will see the following error if \texttt{overwrite} is not set to TRUE

\textbf{Error:}
4 exists already. Set \texttt{overwrite=\texttt{\texttt{T}}} to overwrite.
NPRun: include

- Optional vector of subject IDs to include in the run

- If all IDs are numeric, does not need to be quoted, but should be quoted for alpha IDs

  - e.g. `include=c(1:4, 8, 9, 12)` or `include=c("A1001", "B297")`
NPrun: exclude

- Optional vector of subject IDs to exclude from the run
- If all IDs are numeric, does not need to be quoted, but should be quoted for alpha IDs
- e.g. `exclude=c(1:4, 8, 9, 12)` or `exclude=c("A1001", "B297")`
- You cannot use `include` and `exclude` together
NPrun: ode

- Sets the “stiffness” of the ordinary differential equation solver, with more stiffness taking longer
- Ignored for algebraic models
- Specified as a log value, e.g. -4 (default)

- Not usually required. Higher numbers will result in faster runs but possibly less accurate log-likelihoods and slightly different parameter value distributions
NPrun: tol

- Controls the convergence criterion for NPAG.
- Smaller numbers make it harder to converge.
- Default is 0.01
NPrun: salt

- This is different than bioavailability. It is a vector of numbers for the salt values or active fraction of each input, e.g. \texttt{salt=c(0.8, 1)}

- Default is 1 for all inputs

- Sometimes is other than 1, e.g. aminophylline salt value is 0.8
NPrun: cycles

- Control the maximum number of cycles for NPAG to run, e.g. cycles=5000.
- Default is 100
- Set to 0 in combination with a specified prior to calculate Bayesian posteriors for external data
- The most common argument to change, other than model or data.
**NPrun: indpts**

- The index of the number of starting points
- Automatically set by Pmetrics based on number of random parameters in #PRI block of model file

<table>
<thead>
<tr>
<th>Npar</th>
<th>Index</th>
<th>Gridpoints</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>2129</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>5003</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>10007</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>20011</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>40009</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>80021</td>
</tr>
<tr>
<td>6+</td>
<td>101, 102, …108</td>
<td>(100-index)*80021</td>
</tr>
</tbody>
</table>
NPrun: icen

• The summary function used to generate predictions in the HTML report at the end of a run

• Default is “median”, but could be “mean”

• Both sets of predictions are available in results loaded by PMload()
NPrun: aucint

- Old argument maintained for backwards compatibility
- Determined AUC interval to be automatically included
- `makeAUC()` has replaced this argument
NPrun: idelta

- Controls the frequency for Bayesian prior/posterior predictions for each subject (i.e., PMpost and PMpop objects)
- Specify in 1/60 time units, typically minutes (be careful if your time units are days or something else)
- Default is 12, e.g. 5 predictions per hour
NPrun: prior

- Specify a previous run to use as a prior for the current run, e.g. prior=3
- Could be specified as a filename of a prior density file placed into the /Runs folder at the time of a run
  - Called “DEN0001” and found in the /outputs folder of the run that serves as the prior
NP: overwrite

- Overwrite a previous run
- Used in combination with `run` argument
NP\texttt{run}: nocheck

- Suppresses \texttt{PMcheck()} which automatically runs silently with each \texttt{NP\texttt{run}()}.  
- Default is FALSE.
NPrun: parallel

- Parallel mode will be selected automatically for models that have a #DIF block, i.e. use differential equations
- Serial mode will be selected for algebraic models
- Use this argument to override the default behavior
NPrun examples

- Run 100 cycles with model.txt and data.csv
  - NPrun()

- Run 1000 cycles with model2.txt and data from run 1
  - NPrun(model="model2.txt", data=1, cycles=1000)

- Run 5000 more cycles from run 3, which did not converge
  - NPrun(model=3, data=3, prior=3, cycles=5000)

- Use model in run 5 to calculate Bayesian posterior parameters in new dataset, data2.csv
  - NPrun(model=5, prior=5, data="data2.csv", cycles=0)
IT2B runs

- Iterative 2-stage Bayesian
- Parametric algorithm
- Arguments are the same as for NPrun() except no indpts, aucint, idelta, or prior.
ERRrun

- Variation on IT2B
- Designed to estimate C0, C1, C2, C3 directly from data
- Least preferred method as estimates are model dependent
- Arguments are the same as for IT2B with addition of `search="cursory|medium|extensive"` for depth of coefficient search
Testing comparative model performance

- `PMcompare(x, y, ..., icen = "median", outeq = 1, plot = F)`

- `x = run number 1, y = run number 2, ... = additional run numbers or arguments to plot.PMop() if plot=T`

- do not have to load with PMload() first

- does not load run results to avoid conflicts with runs which may already be loaded

- `icen = basis for predictions, either “median” (default) or “mean”`

- `outeq = output equation to compare`

- `plot = plot obs vs. pred or not`
Comparing runs

<table>
<thead>
<tr>
<th>run</th>
<th>type</th>
<th>nsub</th>
<th>nvar</th>
<th>par</th>
<th>converge</th>
<th>-2*LL</th>
<th>aic</th>
<th>bic</th>
<th>popBias</th>
<th>popImp</th>
<th>popPer_RMSE</th>
<th>postBias</th>
<th>postImp</th>
<th>postPer_RMSE</th>
<th>pval</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NPAG</td>
<td>20</td>
<td>4</td>
<td>Ka Ke V Tlag1</td>
<td>FALSE</td>
<td>440.0</td>
<td>450.4</td>
<td>464.6</td>
<td>1.217</td>
<td>18.34</td>
<td>46.17</td>
<td>-0.08785</td>
<td>0.9888</td>
<td>12.18</td>
<td>NA</td>
</tr>
<tr>
<td>2</td>
<td>NPAG</td>
<td>20</td>
<td>4</td>
<td>Ka Ke V0 Tlag1</td>
<td>FALSE</td>
<td>439.7</td>
<td>450.2</td>
<td>464.4</td>
<td>1.424</td>
<td>17.44</td>
<td>43.07</td>
<td>-0.08622</td>
<td>1.0000</td>
<td>12.39</td>
<td>0.607</td>
</tr>
</tbody>
</table>

Model 1 Population

Model 1 Posterior

Model 2 Population

Model 2 Posterior

PMcompare(1,2,plot=T,cex.stat=0.9)
PMsave

- Save objects attached to a run back to disk to be loaded with PMload() in the future

- PMsave(run, ...)

  - run = suffix of objects to save and run folder to save to, e.g. 2 will save all objects with 2 to run folder 2, e.g. objNew.2 will automatically be saved with PMsave(2) and will be loaded next time with PMload(2)

  - ... = additional objects which will also be attached to run 2 with run suffix, e.g. PMsave(2, objA, objB) will result in objA.2 and objB.2 saved to run 2 folder and loaded next time with PMload(2).
Adding covariates

• Best base model selected
• Error model established
• Then look to explain differences between patients in parameter estimates → covariates
• Be targeted
  – biological plausibility
  – Statistical improvement in model
Adding covariates

#Prim
Cl,2,80
V,5,100
Kcp,1,30
Kpc,1,30

#Cov
CRCL
APACHE

#Sec
Clcr=Cl*CRCL/96
Vapac=V*(APACHE/21)
Ke=Cl/V

#Out
Y(1) = X(1)/V

#Err
G=6
0.5,0.1,0,0!
Adding covariates (2)

#Prim
Cl,2,80
V,5,100
Kcp,1,30
Kpc,1,30

#Cov
CRCL
APACHE

#Sec
Clcr=Cl*CRCL/96
Vapac=V*(APACHE/21)

Ke=Clcr/Vapac

#Out
Y(1) = X(1)/V

#Err
G=1
0.5,0.1,0,0!
Model Evaluation

- Goodness of fit plots
- Pop predictions vs observed
- Individual predictions vs observed
- NPDE
- VPC
Pop & Individual (posterior)
NPDE

- Normalised prediction distribution errors
- Each subject used for 1000 sims using pop model
- makeNPDE()
VPC

SIMrun(poppar=final.1,
model="aaa.txt",data="bbb.csv",nsim=1000,makecsv="ccc.csv",limits = NA)

plot(npde.1,plot.type="vpc")
Then...

• If you are satisfied:
  – Error model is appropriate for your data
  – Structural model is best
  – Covariates are relevant and helpful
  – Goodness of fit plots are acceptable...

Then, you’ve fit your data! If not?
Troubleshooting

• Many potential issues: e.g.
  – Is your csv file correct?
  – Are your initial estimates acceptable? (within boundaries)
  – Is your structural model missing something because of an incorrect assumption?

Check for solutions then re-run
Demo runs

• Two compartments
• Add a covariate
• IT2B run