NONMEM/PsN/qpsn/R testing

Introduction

The purpose of these tests is to test basic function of NONMEM, PsN, R and supporting scripts in AZ's High Performance Computing infrastructure.

Requirements

There is currently no formal requirements document however the following constitutes a draft list from which tests are defined:

NONMEM

NONMEM must be able to perform:

- Population PK analyses (closed form and ODE based)
- Population PK/PD analysis for continuous and categorical endpoints
- All supported estimation methods should work as expected
- MPI parallelisation

PsN

PsN must be able to perform:

- Model execution
- Bootstrap
- Stepwise covariate method (SCM)
- VPC
- SSE

R

R must be able to:

- Read in NONMEM/PsN outputs.
- Produce basic goodness of fit diagnostics

Testing plan

The testing plan is designed to provide a *reasonable* degree of assurance that the system accomplishes its intended requirements.

Tests

List of tests and assessment criteria

- Bootstrap
- Stepwise covariate method (SCM)
- VPC
- SSE

Model execution tests

For each run, NONMEM will be executed via qpsn. For each NONMEM run:

- OFV will be extracted from the psn.ext file using R
- basic goodness of fit diagnostics will be produced using R
- convergence diagnostics will be produced using R

Here is a list of tests, with assessment criteria list as sub-bullets:

1. (*run1.mod*) Theophylline PK (NONMEM installation test):
   - concordance of estimated OFV with vendor provided value
2. (*run2.mod*) Theophylline PK IMP:
   - approximate concordance of estimated OFV with run 1.
   - approximate concordance of basic goodness of fit diagnostics with run 1.
3. (*run3.mod*) Theophylline PK SAEM:
   - approximate concordance of estimated OFV with run 1.
   - approximate concordance of basic goodness of fit diagnostics with run 1.
4. (*run4.mod*) Theophylline PK IMP ODE:
   - close concordance of estimated OFV with run 2.
   - close concordance of convergence diagnostics with run 2.
   - close concordance of basic goodness of fit diagnostics with run 2.
5. (*run5.mod*) Theophylline PK IMP (larger dataset = THEOPPlarge.csv)
   - concordance of parameter estimates with run 2
   - concordance of basic goodness of fit diagnostics with run 2 but with larger file size.
6. (*run6.mod*) Theophylline PK IMP (larger dataset = THEOPPlarge.csv) parallelised with "-c auto"
   - Number of cores selected
   - concordance of OFV with run 5
   - concordance of basic goodness of fit diagnostics as run 5
   - consistency of results under repetition (repeat test 5 times concurrently)
7. (*run7.mod*) Theophylline PK-continuous PD IMP (larger dataset = THEOPPlarge.csv using simulated continuous PD) parallelised
   - concordance of parameter estimates with simulated values
   - basic goodness of fit diagnostics split by endpoint
8. (*run8.mod*) Theophylline PK-continuous PD IMP (larger dataset = THEOPPlarge.csv using simulated bernoulli PD)
   - concordance of parameter estimates with simulated values
   - basic goodness of fit diagnostics split by endpoint
9. (*run9.mod*) Theophylline PK-continuous PD IMP (larger dataset = THEOPPlarge.csv using simulated time to event PD)
   - concordance of parameter estimates with simulated values
   - basic goodness of fit diagnostics split by endpoint

Bootstrap tests

1. (*run2.mod*) Bootstrap -samples=10
   - raw results produced with 10 rows
   - boostrap_results produced

Results

Manual instructions:

1. See tables below - run command in "cmd"
2. After running open Models/XX/raw_results_runXX.csv where XX is the number of the run

3. See “ofv” & “model_run_time” fields. If there are errors, these need debugging.

Model execution results

1. (*run1.mod*) Theophylline PK (NONMEM installation test):
    - concordance of estimated OFV with vendor provided value

test	result_char	result_logical
cmd	qpsn -t 3000 -r 1000 – execute run1.mod -dir=1	NA
status	running:0 finished:1 errors:0	NA
OFV	96.8004496145226	TRUE
gof_size	20660	TRUE
conv_diag_size	3611	TRUE
model_run_time	0:00:21	NA

2. (*run2.mod*) Theophylline PK IMP:
    - approximate concordance of estimated OFV with run 1.
    - approximate concordance of basic goodness of fit diagnostics with run 1.

test	result_char	result_logical
cmd	qpsn -t 3000 -r 1000 – execute run2.mod -dir=2	NA
status	running:0 finished:1 errors:0	NA
OFV	97.3820732016404	TRUE
gof_size	20639	TRUE
conv_diag_size	3611	TRUE
model_run_time	0:00:44	NA

3. (*run3.mod*) Theophylline PK SAEM:
    - approximate concordance of estimated OFV with run 1.
    - approximate concordance of basic goodness of fit diagnostics with run 1.

test	result_char	result_logical
cmd	qpsn -t 3000 -r 1000 – execute run3.mod -dir=3	NA
status	running:0 finished:1 errors:0	NA
OFV	97.3894160207807	TRUE
gof_size	20635	TRUE
conv_diag_size	3611	TRUE
model_run_time	0:00:33	NA

4. (*run4.mod*) Theophylline PK IMP ODE:
    - close concordance of estimated OFV with run 2.
    - close concordance of convergence diagnostics with run 2.
    - close concordance of basic goodness of fit diagnostics with run 2.

test	result_char	result_logical
cmd	qpsn -t 3000 -r 1000 – execute run4.mod -dir=4	NA
status	running:0 finished:1 errors:0	NA
OFV	97.3820732014096	TRUE
gof_size	20639	TRUE
conv_diag_size	3611	TRUE

test	result_char	result_logical
model_run_time	0:02:54	NA

5. (*run5.mod*) Theophylline PK IMP (larger dataset = THEOPPlarge.csv)
   - concordance of parameter estimates with run 2
   - concordance of basic goodness of fit diagnostics with run 2 but with larger file size.

test	result_char	result_logical
cmd	qpsn -t 3000 -r 1000 – execute run5.mod -dir=5	NA
status	running:0 finished:1 errors:0	NA
OFV	2375.52288342877	TRUE
gof_size	253965	TRUE
conv_diag_size	3611	TRUE
model_run_time	0:10:06	NA

6. (*run6.mod*) Theophylline PK IMP (larger dataset = THEOPPlarge.csv) parallelised with "-c auto"
   - Number of cores selected
   - concordance of OFV with run 5
   - concordance of basic goodness of fit diagnostics as run 5
   - consistency of results under repetition (repeat test 5 times concurrently)

test	result_char	result_logical
cmd	qpsn -c auto -t 3000 – execute run6_1.mod -dir=6_1	NA
status	running:0 finished:1 errors:0	NA
OFV	2375.47453002607	TRUE
gof_size	253961	TRUE
conv_diag_size	3611	TRUE
model_run_time	0:01:33	NA

test	result_char	result_logical
cmd	qpsn -c auto -t 3000 – execute run6_2.mod -dir=6_2	NA
status	running:0 finished:1 errors:0	NA
OFV	2375.5183551084	TRUE
gof_size	254073	TRUE
conv_diag_size	3611	TRUE
model_run_time	0:01:47	NA

test	result_char	result_logical
cmd	qpsn -c auto -t 3000 – execute run6_3.mod -dir=6_3	NA
status	running:0 finished:1 errors:0	NA
OFV	2375.51928315953	TRUE
gof_size	253863	TRUE
conv_diag_size	3611	TRUE
model_run_time	0:01:28	NA

test	result_char	result_logical

test	result_char	result_logical
cmd	qpsn -c auto -t 3000 – execute run6_4.mod -dir=6_4	NA
status	running:0 finished:1 errors:0	NA
OFV	2375.51460166246	TRUE
gof_size	253771	TRUE
conv_diag_size	3611	TRUE
model_run_time	0:01:33	NA

test	result_char	result_logical
cmd	qpsn -c auto -t 3000 – execute run6_5.mod -dir=6_5	NA
status	running:0 finished:0 errors:1	NA

7. (*run7.mod*) Theophylline PK-continuous PD IMP (larger dataset = THEOPPlarge.csv using simulated continuous PD) parallelised
    - concordance of parameter estimates with simulated values
    - basic goodness of fit diagnostics split by endpoint

test	result_char	result_logical
cmd	qpsn -c auto -t 3000 -r 1000 – execute run7.mod -dir=7	NA
status	running:0 finished:1 errors:0	NA
OFV	-420.024591920084	TRUE
gof_size	326990	TRUE
conv_diag_size	3611	TRUE
model_run_time	0:04:52	NA

8. (*run8.mod*) Theophylline PK-continuous PD IMP (larger dataset = THEOPPlarge.csv using simulated bernoulli PD)
    - concordance of parameter estimates with simulated values
    - basic goodness of fit diagnostics split by endpoint

test	result_char	result_logical
cmd	qpsn -c auto -t 3000 -r 1000 – execute run8.mod -dir=8	NA
status	running:0 finished:0 errors:1	NA

9. (*run9.mod*) Theophylline PK-continuous PD IMP (larger dataset = THEOPPlarge.csv using simulated time to event PD)
    - concordance of parameter estimates with simulated values
    - basic goodness of fit diagnostics split by endpoint

Bootstrap tests

1. (*run2.mod*) Bootstrap -samples=10
    - raw results produced with 10 rows
    - boostrap_results produced