

RUNRECORD user guide

PsN 4.6.0

Revised 2016-05-13

RunRecord is a PsN based script that facilitates the creation of run records for NONMEM runs. It takes a range of run numbers as input, extracts information and generates a txt file that can be imported into Excel for further manipulation. The information extracted includes parameter estimates, standard errors, condition number, objective function value (OFV), parameter names and information about model components. Some of this comes from the NONMEM output files while some parts are extracted from user-supplied information in the NONMEM model file.

1 Information from the NONMEM output

For each run in the range of run numbers the following information is extracted from the NONMEM output files: OFV, condition number, minimization status, covariance step status, number of individuals, number of observations, estimation step time, covariance step time, parameter estimates and standard errors.

Diagonal omegas and sigmas are reported as standard deviations, off-diagonal omegas and sigmas are reported as correlations. Standard errors are reported as relative standard errors- RSE (SE/estimate). For omegas and sigmas, RSEs are computed for the variances and covariances. By supplying the appropriate options to the runrecord script, this can be changed to variances/covariances and standard errors, respectively (see below).

2 Information from the NONMEM model file

It is assumed that the user specifies certain information, in a certain way, in the model file. Below is an example model file (for run number 2):

```
$PROB Simpraz - full data set
;; 1. Based on: 1
;; 2. Description:
;;   Added an OMEGA BLOCK(2) for CL and V
;; 3. Label:
;;   Basic model
;; 4. Structural model:
;;   One compartment linear model
;; 5. Covariate model:
;;   No covariates
;; 6. Interindividual variability:
;;   CL, V and KA. BLOCK(2) for CL and V
;; 7. Interoccasion variability:
;; 8. Residual variability:
;;   Proportional
;; 9. Estimation:
;;   FO
$INPUT ID SEX AGE RACE HT SMOK HCTZ PROP CON AMT WT TIME SECR
      DV RATE EVID OCC
$DATA newdata2.dta IGNORE=#
$SUBS ADVAN2 TRANS2

$PK

      TVCL = THETA(1)
      TVV  = THETA(2)
      TVKA = THETA(3)

      CL  = TVCL * EXP(ETA(1))
      V   = TVV  * EXP(ETA(2))
      KA  = TVKA * EXP(ETA(3))

      S2 = V
```

\$ERROR

```
IPRED = F
IRES  = DV - F
W     = F
IF(W.EQ.0) W = 1
IWRES = IRES/W
Y     = IPRED + W*EPS(1)
```

\$THETA

```
(0,18.7) ; CL ; L/h
(0,87.3) ; V  ; L
(0,2.13) ; KA ; 1/h
```

\$OMEGA BLOCK(2)

```
0.128    ; IIV_CL
0.01     ; CORR_CL_V
0.142    ; IIV_V
```

\$OMEGA

```
1.82     ; IIV_KA
```

\$SIGMA 0.0231

\$EST NOABORT POSTHOC

\$COV PRINT=E

3 Information tags

Just below the \$PROBLEM statement are lines that begin with two semi-colon, which contains structured information about the run.

Information tag	Description/Purpose
Based on	Specifies the run number of the parent run (used for computing delta OFV). Should be left empty if there is no parent run. If [nodOFV] is specified after the parent run number, no delta OFV will be computed, i.e. the current run is based on the parent but it is not relevant to compute a delta OFV.
Description	Should include a brief description of the run. For example what has been changed from the parent run.
Label	This is a name for the model. It is meant to make it easier to identify model versions or stages in the run record. For example, runs 1-8 may be labeled Basic model while 8-20 may be labeled Full model (the latter indicating that covariates are being included into the model). The final model could, obviously, be labeled Final model.
Structural model	A description of the structural model.
Covariate model	A description of the covariate model.
Interindividual variability	A description of how between subject variability is modeled.
Interoccasion variability	A description of how within subject, between occasion variability is modeled.
Residual variability	A description of how the residual variability is modeled.
Estimation	A description of the estimation options used.

- Note that it is only Based on: that expects information on the same line. The other information tags are placeholders for the line just below it until the line of the next information tag.
- An information tag may be empty, but it is a good idea to still keep the tag in the model file. If a tag should be empty (like Inter-occasion

variability: if no IOV is in the model), only include the line with the tag, not the empty information line below it.

- The order of the tags is irrelevant.
- The number of the tags (as in the example above) is irrelevant. As a matter of fact, the numbering is not necessary at all but may be a good way to impose some order.
- The colon (:) after the tag is required, as is the exact spelling.

4 Parameter labels

Each initial parameter estimate can optionally have an associated label. This is specified by adding a semi-colon and the label after the initial parameter estimate. When constructing the txt file, the runrecord script indexes the parameter estimates after the labels. This means that, for example, THETA(1) may be CL in one run and V and the second as long as the label is correctly specified. It also means that parameters should be given consistent labels across runs.

5 Running runrecord

In the simplest form, runrecord is run in the following way:

```
runrecord -to=10
```

This will make runrecord to extract information from runs 1-10 and put it in a file called AArunInfo.txt.

The behavior of run record can be influenced by options:

-from = *N*

Run number to start processing from. Default is 1.

-to = *N*

The highest run number to process. Mandatory.

-root = *string*

"root" name of the model files. Defaults to "run", e.g. run1.mod

- res_file** = *file name*
The name of the results file. The default is “AAruninfo.txt”.
- mod_ext** = *string*
Model file extension. Defaults to “mod”, e.g. run1.mod.
- out_ext** = *string*
Output file extension. Defaults to “lst”, e.g. run1.lst.
- sep** = *sign*
Separator between fields in the AAruninfo.txt file. Defaults to “.”.
- rsep** = *sign*
Separator between comment rows read from the model file. This is pasted between rows -within fields- in the output file.
- max_lvl** = *N*
Max number of model children levels.
- var** = *yes or no*
Whether omegas and sigmas should be reported as variances or sd. Default is 'no'.
- rse** = *yes or no*
Whether standard errors should be reported as relative SEs or not. The default is 'yes'.

6 Importing the txt file into Excel

The txt file can just be opened in Excel using the standard File->Open procedure. This will open an unformatted version of the txt file. However, to improve readability and consistency, it may be more convenient to use the Data->Get external data command instead. This will create a connection between the spread sheet and the AArunInfo.txt file that can easily be updated without losing carefully applied formatting. The following procedure is proposed:

1. Create a first version of the AArunInfo.txt file based on one or two runs.

2. Open an empty spreadsheet in Excel.
3. Select Data->Get External Data->Import Text File
4. Select the AArunInfo.txt file, press Next, select Semicolon as delimiter and press Finish.
5. In the Import Data dialog, select Properties:
6. Under
If the number of rows in the data range changes upon refresh
make sure that
“Overwrite existing cells with new data, clear unused cells”
is selected and the other alternatives are deselected.
7. Press OK twice, and the data is in the spreadsheet.
8. Format the information as desired.
9. There is an External Data toolbar that is handy when new runs are to be added to the run record: By pressing the red exclamation mark, the contents of the spreadsheet is updated to include the current information of the AArunInfo.txt file. So, if the runrecord script has been run to include more NONMEM runs in the AArunInfo.txt file, it is just to open the Excel spreadsheet and press the red exclamation mark to include the new information.
10. There seems to be one bug in the Refresh Data procedure in the previous bullet - line heights are reset to the default: Fortunately, this is easily fixed. Select the whole spreadsheet (ctrl-A in Windows or cmd-A on the Mac) and Select Format->Cells, select the Alignment tab: and click three times in the Wrap text check box! (It is silly, but it works!)