Notes on RNMImport

Francisco Gochez, Mango Solutions

February 24, 2011

1 Introduction

This set of notes is a brief overview of the RNMImport package. At the moment (version 0.1.17), these are quite terse, but will be expanded upon in later releases. This is meant to give only a basic idea of how the package works.

2 Importing runs

```
Full path to configuration file:  C:/Users/croosen/AppData/Local/Temp/RtmpU0yMl6/Rinst14a4
------------------------------------------------------
|
|
|To execute unit tests, type runRNMImportTests().
|Report will be written to RNMImport_internalunit.html
|
|
------------------------------------------------------
```

The main command for importing a NONMEM run is `importNm`, which works with a control file and a path. List files names are automatically deduced from allowable file extensions (see meta data section below), but can be passed explicitly.

```
> # Import an example run
> runPath <- system.file(package = "RNMImport", "unittests/testdata/TestRun")
> # List file deduced automatically
> run <- importNm(conFile = "TestData1.ctl", path = runPath)
> print(run)
```

```
Control file:
            size mode               mtime               ctime
controlFile  941  666 2011-02-24 14:08:13 2011-02-24 14:08:13
                          atime exe
controlFile 2011-02-24 14:08:13  no

controlFile C:/Users/croosen/AppData/Local/Temp/RtmpU0yMl6/Rinst14a439cf/RNMImport/unittes
Output report file:
           size mode               mtime               ctime
reportFile 7820  666 2011-02-24 14:08:13 2011-02-24 14:08:13
                         atime exe
reportFile 2011-02-24 14:08:13  no

reportFile C:/Users/croosen/AppData/Local/Temp/RtmpU0yMl6/Rinst14a439cf/RNMImport/unittest
Number of problems: 1
Problems:
*******************
Problem  1

Standard NONMEM problem:
###############
```

```
Problem statement:  System Test 1
Data file: Data1
Input table dimensions:
1061 19
Input table columns:
SID SEX AGE RACE HT SMOK HCTZ PROP CON AMT WT TIME SECR DV EVID SS II ID OCC
PK:
[1] "TVCL=THETA(1) "        "TVV=THETA(2) "          "TVKA=THETA(3) "
[4] "CL=TVCL*EXP(ETA(1) ) " "V =TVV *EXP(ETA(2) ) " "KA=TVKA*EXP(ETA(3) ) "
[7] "S2=V"

Error:
[1] "IPRED = F"                 "IRES  = DV - F"
[3] "W     = F"                 "IF(W.EQ.0) W = 1"
[5] "IWRES = IRES/W"            "Y     = IPRED + W*EPS(1) "
Parameter estimates:
###############
THETAs:
THETA1 THETA2 THETA3
 19.60  84.60   1.66
OMEGAs:
       OMEGA1 OMEGA2 OMEGA3
OMEGA1  0.164  0.000    0.0
OMEGA2  0.000  0.165    0.0
OMEGA3  0.000  0.000    1.3
SIGMAs:
[1] 0.0202
Output table files: sdtab1,patab1,cotab1,catab1,mutab1,mytab1
Output table dimensions:
1061 24
Output table columns:
ID TIME IPRED IWRES DV PRED RES WRES CL V KA AGE HT WT SECR SEX RACE SMOK HCTZ PROP CON OC
```

```
> print(class(run))
```

```
[1] "NMRun"
attr(,"package")
[1] "RNMImport"
```

When calling `importNm`, a control file, “list” file and output table files are all required, else an error is generated. Input data tables are optional, but a warning is omitted if it is missing. The returned object is of class `NMRun`, whose declaration is given below:

```
setClass(
        "NMRun",
        representation(
                        controlText = "character",
                        lstText = "character",
                        controlComments = "character",
                        controlFileInfo = "data.frame",
                        listFileInfo = "data.frame",
                        numProblems = "numeric",
                        problems = "list"
                        ),
                validity = validity.NMRun
                )
```

The information of primary interest is in `problems`. This list has one element corresponding to each `$PROB` statement in the control file, although at the moment, ONLY ONE problem statement can be handled. An individual problem can be extracted with the `getProblem` function.

```
> prob <- getProblem(run)
> print(prob)

Standard NONMEM problem:
###############
Problem statement:  System Test 1
Data file: Data1
Input table dimensions:
1061 19
Input table columns:
SID SEX AGE RACE HT SMOK HCTZ PROP CON AMT WT TIME SECR DV EVID SS II ID OCC
PK:
[1] "TVCL=THETA(1) "        "TVV=THETA(2) "         "TVKA=THETA(3) "
[4] "CL=TVCL*EXP(ETA(1) ) " "V =TVV *EXP(ETA(2) ) " "KA=TVKA*EXP(ETA(3) ) "
[7] "S2=V"

Error:
[1] "IPRED = F"                 "IRES  = DV - F"
[3] "W     = F"                 "IF(W.EQ.0) W = 1"
[5] "IWRES = IRES/W"            "Y     = IPRED + W*EPS(1) "
Parameter estimates:
###############
THETAs:
THETA1 THETA2 THETA3
 19.60  84.60   1.66
OMEGAs:
       OMEGA1 OMEGA2 OMEGA3
OMEGA1  0.164  0.000    0.0
OMEGA2  0.000  0.165    0.0
```

```
OMEGA3  0.000  0.000    1.3
SIGMAs:
[1] 0.0202
Output table files: sdtab1,patab1,cotab1,catab1,mutab1,mytab1
Output table dimensions:
1061 24
Output table columns:
ID TIME IPRED IWRES DV PRED RES WRES CL V KA AGE HT WT SECR SEX RACE SMOK HCTZ PROP CON OC
```

Individual problems can be of class `NMBasicModel`, `NMSimDataGen` or `NMSimModel`.

- `NMBasicModel` Is a standard NONMEM model fit, without simulation step
- `NMSimDataGen` Is a NONMEM problem with simulation step, but no model fitting
- `NMSimModel` Is a NONMEM propblem with simulation and model fitting on each simulation

Below is an example of loading a simulation + model fitting run:

```
> runPath <- system.file(package = "RNMImport", "unittests/testdata/TestSimRun")
> # List file deduced automatically
> simRun <- importNm(conFile = "TestData1SIM.con", path = runPath)
> simProblem <- getProblem(simRun)
> print(simProblem)

NONMEM data simulation problem:
###############
Problem statement:  System Test 2
PK:
[1] "TVCL=THETA(1) "        "TVV=THETA(2) "         "TVKA=THETA(3) "
[4] "CL=TVCL*EXP(ETA(1) ) " "V =TVV *EXP(ETA(2) ) " "KA=TVKA*EXP(ETA(3) ) "
[7] "S2=V"

Data file: Data1
Input table dimensions:
1061 19
Input table columns:
SID SEX AGE RACE HT SMOK HCTZ PROP CON AMT WT TIME SECR DV EVID SS II ID OCC
Output table files: sdtab1,patab1,cotab1,catab1,mutab1,mytab1
Output table dimensions:
5305 24
Output table columns:
ID TIME IPRED IWRES DV PRED RES WRES CL V KA AGE HT WT SECR SEX RACE SMOK HCTZ PROP CON OC
Number of simulations performed:  5
Seeds:  20050213
Final estimates for each subproblem:

THETAs:
     THETA1 THETA2 THETA3
```

```
sim1   17.2  117.0   1.24
sim2   18.3  108.0   1.40
sim3   18.0  108.0   1.24
sim4   19.0   98.9   1.55
sim5   17.1  109.0   1.38
OMEGAs:
, , sim1

       OMEGA1 OMEGA2 OMEGA3
OMEGA1  0.174   0.00   0.00
OMEGA2  0.000   0.19   0.00
OMEGA3  0.000   0.00   1.45

, , sim2

       OMEGA1 OMEGA2 OMEGA3
OMEGA1  0.167  0.000   0.00
OMEGA2  0.000  0.143   0.00
OMEGA3  0.000  0.000   1.24

, , sim3

       OMEGA1 OMEGA2 OMEGA3
OMEGA1  0.181  0.000   0.00
OMEGA2  0.000  0.142   0.00
OMEGA3  0.000  0.000   1.57

, , sim4

       OMEGA1 OMEGA2 OMEGA3
OMEGA1  0.245  0.000  0.000
OMEGA2  0.000  0.189  0.000
OMEGA3  0.000  0.000  0.945

, , sim5

       OMEGA1 OMEGA2 OMEGA3
OMEGA1  0.174  0.000   0.00
OMEGA2  0.000  0.197   0.00
OMEGA3  0.000  0.000   1.47

SIGMAs:
, , sim1

       SIGMA1
SIGMA1 0.0241

, , sim2

       SIGMA1
```

```
SIGMA1  0.026

, , sim3

       SIGMA1
SIGMA1 0.0262

, , sim4

       SIGMA1
SIGMA1 0.0295

, , sim5

       SIGMA1
SIGMA1 0.0244

Objective function(s):
    sim1     sim2     sim3     sim4     sim5
3696.247 3575.252 3660.355 3606.526 3701.472
```

3 Extracting data from a problem

3.1 Parameter estimates

For retrieving parameter estimates, one uses the functions `getThetas`, `getOmegas` and `getSigmas`. These take an additional parameter `stdError`, which controls whether or not standard errors should be returned if they are available.

```
> print(getThetas(prob))

THETA1 THETA2 THETA3
 19.60  84.60   1.66

> print(getOmegas(prob))

       OMEGA1 OMEGA2 OMEGA3
OMEGA1  0.164  0.000    0.0
OMEGA2  0.000  0.165    0.0
OMEGA3  0.000  0.000    1.3

> print(getThetas(simProblem))

THETA1 THETA2 THETA3
 17.20 117.00   1.24

> print(getOmegas(simProblem))

, , sim1

       OMEGA1 OMEGA2 OMEGA3
```

```
OMEGA1  0.174   0.00   0.00
OMEGA2  0.000   0.19   0.00
OMEGA3  0.000   0.00   1.45
```

Additional extraction functions include `getObjective`, `getEstimateCov` (extract estimator covariance and correlation matrices), `getControlStatements` (extract an object holding the parsed statements of an object's control file), and others. See the online help for full details. Note that these can be used with `NMRun` objects as long as the problem numnber is specified (it is 1 by default), for instance:

```
> print(getThetas(run, problemNum = 1))

THETA1 THETA2 THETA3
 19.60  84.60   1.66

> print(getOmegas(prob, problemNum = 1))

       OMEGA1 OMEGA2 OMEGA3
OMEGA1  0.164  0.000    0.0
OMEGA2  0.000  0.165    0.0
OMEGA3  0.000  0.000    1.3

>
```

3.2 Input and output data

`nmData` is a generic function for extracting a NONMEM run's input and output data tables, as described by the control file $DATA and $TABLE statements. The data is allowed to be missing when a run is loaded, in which case obviously it will not be retrievable. For a basic model, `nmData` has the following arguments:

- `obj` - `NMBasicProblem` class object
- `dataTypes` - `character` vector with strings `input` and/or `output`, determines which type of data is to be retrieved.
- `returnMode` - Whether to return data as a list of input and outputs, or a single data frame

```
> probOutData <- nmData(prob, dataTypes = "output")
> print(head(probOutData))

  ID TIME    IPRED      IWRES     DV    PRED       RES      WRES     CL     V
1  1    0   8.2032 19.7210000 169.98  9.0449   0.00000  0.000000 18.004 72.62
2  1    1 113.5800  0.0086953 114.57 89.9790  24.59100  0.330790 18.004 72.62
3  1    2  96.9540  0.1000000 106.65 87.0960  19.55400  0.888990 18.004 72.62
4  1    3  76.3100  0.0609290  80.96 72.0560   8.90360  0.599010 18.004 72.62
5  1    4  59.6050 -0.3017300  41.62 57.7050 -16.08500 -2.036100 18.004 72.62
6  1    5  46.5210 -0.0073314  46.18 45.8640   0.31617  0.079517 18.004 72.62
      KA AGE  HT    WT SECR SEX RACE SMOK HCTZ PROP CON OCC SID  absWRES
1 2.5569  55 154 80.97    1   2    2    0    1    1   1   0   1 0.000000
2 2.5569  55 154 80.97    1   2    2    0    1    1   1   0   1 0.330790
```

```
3 2.5569  55 154 80.97    1   2    2    0    1    1   1   0   1 0.888990
4 2.5569  55 154 80.97    1   2    2    0    1    1   1   0   1 0.599010
5 2.5569  55 154 80.97    1   2    2    0    1    1   1   0   1 2.036100
6 2.5569  55 154 80.97    1   2    2    0    1    1   1   0   1 0.079517
```

```
> probData <- nmData(prob)
> print(head(probData))
```

```
  ID TIME    IPRED      IWRES     DV    PRED       RES      WRES     CL     V
1  1    0   8.2032 19.7210000 169.98  9.0449   0.00000  0.000000 18.004 72.62
2  1    1 113.5800  0.0086953 114.57 89.9790  24.59100  0.330790 18.004 72.62
3  1    2  96.9540  0.1000000 106.65 87.0960  19.55400  0.888990 18.004 72.62
4  1    3  76.3100  0.0609290  80.96 72.0560   8.90360  0.599010 18.004 72.62
5  1    4  59.6050 -0.3017300  41.62 57.7050 -16.08500 -2.036100 18.004 72.62
6  1    5  46.5210 -0.0073314  46.18 45.8640   0.31617  0.079517 18.004 72.62
      KA AGE  HT    WT SECR SEX RACE SMOK HCTZ PROP CON OCC SID  absWRES   AMT
1 2.5569  55 154 80.97    1   2    2    0    1    1   1   0   1 0.000000 10000
2 2.5569  55 154 80.97    1   2    2    0    1    1   1   0   1 0.330790     0
3 2.5569  55 154 80.97    1   2    2    0    1    1   1   0   1 0.888990     0
4 2.5569  55 154 80.97    1   2    2    0    1    1   1   0   1 0.599010     0
5 2.5569  55 154 80.97    1   2    2    0    1    1   1   0   1 2.036100     0
6 2.5569  55 154 80.97    1   2    2    0    1    1   1   0   1 0.079517     0
  EVID SS II SID.INPUT SEX.INPUT AGE.INPUT RACE.INPUT HT.INPUT SMOK.INPUT
1    1  1 12         1         2        55          2      154          0
2    0  0  0         1         2        55          2      154          0
3    0  0  0         1         2        55          2      154          0
4    0  0  0         1         2        55          2      154          0
5    0  0  0         1         2        55          2      154          0
6    0  0  0         1         2        55          2      154          0
  HCTZ.INPUT PROP.INPUT CON.INPUT WT.INPUT TIME.INPUT SECR.INPUT DV.INPUT
1          1          1         1    80.97          0          1   169.98
2          1          1         1    80.97          1          1   114.57
3          1          1         1    80.97          2          1   106.65
4          1          1         1    80.97          3          1    80.96
5          1          1         1    80.97          4          1    41.62
6          1          1         1    80.97          5          1    46.18
  ID.INPUT OCC.INPUT
1        1         0
2        1         0
3        1         0
4        1         0
5        1         0
6        1         0
```

Note that the `|INPUT` postfix is used to handle data that is repeated in the output and input tables. Precedence is given to output data, which has no postfix. For simulation problems, one can select a vector of subproblems from which to extract the data. Note that at the moment if more than one subproblem is selected yet the user requests output and input data as a single `data.frame`, and error will be generated due to the mismatch in data size. In future releases, we plan to allow this by repeating the input data as necessary.

```
> simOutData <- nmData(simRun, dataTypes = "output", subProblemNum = 2:3, problemNum = 1)
> print(dim(simOutData))
```

```
[1] 2122   25
```

Data may also be extracted by type via the `nmDatabyType` function. This extracts columns according to the type of data they hold, and type mappings are defined in the metadata. See the next section for details.

```
> x <- nmDatabyVarType(run, varTypes = "Parameter,Covariate", problemNum = 1 )
> print(head(x))
```

```
      CL     KA     V AGE RACE SEX SMOK
1 18.004 2.5569 72.62  55    2   2    0
2 18.004 2.5569 72.62  55    2   2    0
3 18.004 2.5569 72.62  55    2   2    0
4 18.004 2.5569 72.62  55    2   2    0
5 18.004 2.5569 72.62  55    2   2    0
6 18.004 2.5569 72.62  55    2   2    0
```

Additional variables may be created by certain functions, including `addDerivedCategorical`, which derives a categorical variable from an existing data column. These added columns may then be extracted with `addedData`.

```
> prob <- getProblem(run)
> prob <- addDerivedCategorical(prob, "RES", "RES.CUT", breaks = 3, labels = c("low", "med
> print(head(addedData(prob)))
```

```
  RES.CUT
1  medium
2    high
3    high
4  medium
5  medium
6  medium
```

4 Configuration / metadata

RNMImport has tools for modifying the package configuration. For instance, paths can be stored under "names". These names can be referenced by using round brackets in numerous functions.

```
> print(runPath)
```

```
[1] "C:/Users/croosen/AppData/Local/Temp/RtmpUOyMl6/Rinst14a439cf/RNMImport/unittests/test
```

```
> setNmPath("runPath", runPath)
> # note the use of round brackets
> controlContents <- importNmMod("TestData1SIM.con", path = "(runPath)" )
> print(head(controlContents))
```

```
$Raw
 [1] "$PROB System Test 2"
 [2] "$INPUT SID SEX AGE RACE HT SMOK HCTZ PROP CON AMT WT TIME SECR"
 [3] "DV DROP=RATE EVID SS II ID OCC"
 [4] "$DATA Data1 IGNORE=@"
 [5] "$SUBROUTINE ADVAN2 TRANS2"
 [6] "$PK "
 [7] "TVCL=THETA(1)"
 [8] "TVV=THETA(2)"
 [9] "TVKA=THETA(3)"
[10] "CL=TVCL*EXP(ETA(1))"
[11] "V =TVV *EXP(ETA(2))"
[12] "KA=TVKA*EXP(ETA(3))"
[13] "S2=V"
[14] "$THETA 19.6 84.6 1.66"
[15] "$OMEGA .164 .165"
[16] "$OMEGA 1.30"
[17] "$ERROR "
[18] "IPRED = F"
[19] "IRES  = DV - F"
[20] "W     = F"
[21] "IF(W.EQ.0) W = 1"
[22] "IWRES = IRES/W"
[23] "Y     = IPRED + W*EPS(1)"
[24] "$SIGMA  0.0202 "
[25] "$SIM (20050213) SUBPROBLEMS=5"
[26] "$EST NOABORT POSTHOC "
[27] "$TABLE ID  TIME IPRED IWRES "
[28] "NOPRINT ONEHEADER FILE=sdtab1"
[29] "$TABLE ID  CL V KA "
[30] "NOPRINT  ONEHEADER FILE=patab1"
[31] "$TABLE ID AGE HT WT SECR "
[32] "NOPRINT  ONEHEADER FILE=cotab1"
[33] "$TABLE ID SEX RACE SMOK HCTZ PROP CON"
[34] "NOPRINT  ONEHEADER FILE=catab1"
[35] "$TABLE ID  OCC TIME IPRED IWRES "
[36] "NOPRINT ONEHEADER FILE=mutab1"
[37] "$TABLE SID NOPRINT ONEHEADER FILE=mytab1"

$Comments
NULL

$controlFile
[1] "C:/Users/croosen/AppData/Local/Temp/RtmpUOyMl6/Rinst14a439cf/RNMImport/unittests/test

$problemContents
$problemContents[[1]]
$problemContents[[1]]$Theta
       Lower   Est Upper
THETA1  -Inf 19.60   Inf
```

```
THETA2  -Inf 84.60   Inf
THETA3  -Inf  1.66   Inf

$problemContents[[1]]$Omega
       OMEGA1 OMEGA2 OMEGA3
OMEGA1  0.164  0.000    0.0
OMEGA2  0.000  0.165    0.0
OMEGA3  0.000  0.000    1.3

$problemContents[[1]]$Sigma
       SIGMA1
SIGMA1 0.0202

$problemContents[[1]]$Problem
[1] "System Test 2"

$problemContents[[1]]$Tables
    File                              Columns NoHeader firstOnly append
1 sdtab1               ID, TIME, IPRED, IWRES    FALSE     FALSE   TRUE
2 patab1                        ID, CL, V, KA    FALSE     FALSE   TRUE
3 cotab1                ID, AGE, HT, WT, SECR    FALSE     FALSE   TRUE
4 catab1 ID, SEX, RACE, SMOK, HCTZ, PROP, CON    FALSE     FALSE   TRUE
5 mutab1          ID, OCC, TIME, IPRED, IWRES    FALSE     FALSE   TRUE
6 mytab1                                  SID    FALSE     FALSE   TRUE

$problemContents[[1]]$Subroutine
[1] "ADVAN2" "TRANS2"

$problemContents[[1]]$Input
   nmName Label
1  "SID"  "SID"
2  "SEX"  "SEX"
3  "AGE"  "AGE"
4  "RACE" "RACE"
5  "HT"   "HT"
6  "SMOK" "SMOK"
7  "HCTZ" "HCTZ"
8  "PROP" "PROP"
9  "CON"  "CON"
10 "AMT"  "AMT"
11 "WT"   "WT"
12 "TIME" "TIME"
13 "SECR" "SECR"
14 "DV"   "DV"
15 "DROP" "RATE"
16 "EVID" "EVID"
17 "SS"   "SS"
18 "II"   "II"
19 "ID"   "ID"
20 "OCC"  "OCC"
```

```
$problemContents[[1]]$Data
     File    IG  ACCEPT REWIND  RECORDS TRANSLATE NULL
[1,] "Data1" "@" ""     "FALSE" ""      ""        ""

$problemContents[[1]]$Sim
      nSub      Seed1      Seed2       TRUE    simOnly
       "5" "20050213"       "-1"  "INITIAL"    "FALSE"
attr(,"rawStatement")
[1] "(20050213) SUBPROBLEMS=5"

$problemContents[[1]]$PK
[1] "TVCL=THETA(1) "        "TVV=THETA(2) "         "TVKA=THETA(3) "
[4] "CL=TVCL*EXP(ETA(1) ) " "V =TVV *EXP(ETA(2) ) " "KA=TVKA*EXP(ETA(3) ) "
[7] "S2=V"

$problemContents[[1]]$Error
[1] "IPRED = F"                 "IRES  = DV - F"
[3] "W     = F"                 "IF(W.EQ.0) W = 1"
[5] "IWRES = IRES/W"            "Y     = IPRED + W*EPS(1) "

$problemContents[[1]]$Estimates
[1] "NOABORT POSTHOC"
```

```
> removeNmPath("runPath")
```

One can also configure categorical variable “formats”, which define how levels of the category should be interpreted, as well as what the variables mean. These format descriptions are comma seperated lists. Below we show the existing formats (defaults are defined in a file included with the package) for `SEX` and `SMOK`, and then change `SMOK`. The function `imposeCategoryFormat` then forces variables to take a particular format.

```
> print(getVarDescription(c("SEX", "SMOK")))
```

```
   Variable   Label           Format   VarType
63      SEX  Gender 0=male, 1=female Covariate
64     SMOK Smoking      0=no, 1=yes Covariate
```

```
> setVarDescription("SMOK", "Smokes", varFormat = "0=NO, 1 = YES", varType = "Covariate")
> prob <- imposeCategoryFormat(prob, varSubset = "SMOK")
> nmData(prob)[,"SMOK"]
```

```
   [1] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
  [19] NO  NO  NO  NO  YES YES YES YES YES YES YES YES YES YES YES NO  NO  NO
  [37] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
  [55] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
  [73] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
  [91] NO  NO  NO  NO  NO  YES YES YES YES YES YES YES YES YES YES YES YES YES
 [109] YES YES YES YES YES YES YES YES NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [127] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [145] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [163] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [181] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [199] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  YES YES YES YES YES YES
 [217] YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES
 [235] YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES
 [253] YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES
 [271] YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES
 [289] YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES
 [307] YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES
 [325] YES YES NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [343] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [361] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [379] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [397] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [415] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [433] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [451] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [469] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [487] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [505] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [523] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [541] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [559] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  YES
 [577] YES YES YES YES YES YES NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [595] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [613] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [631] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [649] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [667] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  YES YES YES YES
 [685] YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES
 [703] YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES
 [721] YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES
 [739] YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES
 [757] YES YES YES YES YES YES YES YES NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [775] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [793] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [811] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [829] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  YES
 [847] YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES YES
 [865] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [883] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [901] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [919] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [937] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [955] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [973] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
 [991] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
[1009] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
[1027] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
```

```
[1045] NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO  NO
Levels: NO YES
```