Pharmacoepidemiological Research on Outcomes of Therapeutics by a European ConsorTium

IMI Work Package 5: Supplement to Rimonabant Wave 1 Case Study

Report 1.b.i: Rimonabant

11/10/2011

Shahrul Mt-Isa, Ed Chan, Georgy Genov, Juhaeri Juhaeri

On behalf of PROTECT Work Package 5 participants

Supplement dates from 11 Oct 2011 and is the first version released externally Date of any subsequent amendments below	Person making amendments	Brief description of amendments

Disclaimer: The processes described and conclusions drawn from the work presented herein relate solely to the testing of methodologies and representations for the evaluation of benefit and risk of medicines. This report neither replaces nor is intended to replace or comment on any regulatory decisions made by national regulatory agencies, nor the European Medicines Agency

Acknowledgements: The research leading to these results was conducted as part of the PROTECT consortium (Pharmacoepidemiological Research on Outcomes of Therapeutics by a European ConsorTium, www.imi-protect.eu) which is a public-private partnership coordinated by the European Medicines Agency.
The PROTECT project has received support from the Innovative Medicines Initiative Joint Undertaking (www.imi.europa.eu) under Grant Agreement n° 115004, resources of which are composed of financial contribution from the European Union's Seventh Framework Programme (FP7/2007-2013) and EFPIA companies' in kind contribution

Contents

Glossary

Abbreviation	Description
AR	Attributable risk
PAR	Population attributable risk
AFE	Attributable fraction among exposed
DIN	Disease impact number
PIN	Population impact number
CIN	Case impact number
EIN	Exposure impact number (also known as the number needed to treat or NNT)
ECIN	Exposed cases impact number
PIN-ER-*t*	Population impact number of eliminating a risk factor over time *t*
NEPP	Number of events prevented in a population

1 Rimonabant Case Study Final Report Supplement

1.1 Simulation engine

Purpose of development

It is not the intention for the users to directly use –_*simulin()*–; it is a “back-end” programme that interacts with the main “front-end” programme –*pims*– as discussed in Scenario 1 described in point 2 below. Therefore, the instructions for use of –_*simulin()*– are left undocumented.

Description of arguments

The simulation engine is named –_*simulin()*– and takes 17 arguments. The first five arguments are compulsory which are the number of iterations (also see 1.2) and the items in the first four columns in Table 1. The rows of Table 1 indicate the epidemiological measures and impact numbers that will be calculated when the parameters in the columns are specified.

Table 1 Input parameters and metrics calculated

	I_u	RR	RR$_{LCI}$	RR$_{UCI}$	P_e	P_d	n	n_{sample}	v_k
AR	X	X	X	X				X	X
EIN	X	X	X	X				X	X
AFE	X	X	X	X				X	X
ECIN	X	X	X	X				X	X
PAR	X	X	X	X	X			X	X
CIN	X	X	X	X	X			X	X
DIN	X	X	X	X	X			X	X
PIN	X	X	X	X	X	X		X	X
PIN-ER-*t*	X	X	X	X	X		X	X	X
NEPP	X	X	X	X	X	X	X	X	X

Distributions for simulation

Let $X_1, X_2, \ldots, X_k$ be the simulated number of people with certain events or characteristics, and defined further below.

Let $\nu_1, \nu_2, \ldots, \nu_k$ be the respective number people from where the data are based which are fixed.

We specify I_u as follows:

$$X_1 \sim Bin(\nu_1, I_u)$$

$$r_u = \frac{X_1}{\nu_1} = \text{simulated rates in baseline population}$$

We specify RR as follows:

$$lrr = \log(\mathrm{RR})$$

$$\Lambda_{\mathrm{lrr}} \sim Normal\left(lrr, \sigma_{lrr}^2\right)$$

$$\rho = \mathrm{e}^{\Gamma_{\mathrm{lrr}}} = \text{simulated relative risk}$$

$$\sigma_{lrr} = \frac{\log(\mathrm{RR}_{uci}) - \log(\mathrm{RR}_{lci})}{2 \times \phi^{-1}\left(\pi\left(\mathrm{z}_{\alpha/2}\right)\right)} = \text{observed standard error}$$

where $\pi\left(z_{\alpha/2}\right)$ is the cumulative standard normal distribution corresponding to the confidence intervals. Default will assume 95% CI.

We specify P_e and P_d as follows:

$$X_2 \sim Bin(\nu_2, P_e)$$

$$X_3 \sim Bin(\nu_3, P_d)$$

and

$$r_e = \frac{X_2}{\nu_2} = \text{simulated rates of exposure}$$

$$r_d = \frac{X_3}{\nu_3} = \text{simulated rates of condition}$$

The size of the population of interest by default is assumed to be fixed. But we can also allow for n to be estimated with uncertainty, where the user will have to supply the standard deviation Σ:

$$n_S \sim Normal(n, \Sigma^2) = \text{simulated total population}$$

or otherwise

$$n_S = n$$

From the simulation of input parameters, the program will calculate the impact numbers as follows:

Calculated metrics	**Formula**
Attributable risk (AR)	$r_u \times (\rho - 1)$
Population attributable risk (PAR)	$\frac{r_e \times (\rho - 1)}{1 + r_e \times (\rho - 1)}$
Attributable fraction among exposed (AFE)	$\frac{\rho - 1}{\rho}$
Disease impact number (DIN)	$\frac{1}{\text{AR} \times r_e}$
Population impact number (PIN)	$\text{DIN} \times \frac{1}{r_d}$
Case impact number (CIN)	$\frac{1}{\text{PAR}}$
Exposure impact number (EIN) or NNT	$\frac{1}{\text{AR}}$
Exposed cases impact number (ECIN)	$\frac{1}{\text{AFE}}$
Population impact number of eliminating a risk factor over time *t* (PIN-ER-*t*)	$n_S \times r_u \times \text{PAR}$
Number of events prevented in a population (NEPP)	$n_S \times r_e \times r_d \times r_u \times (\rho - 1)$

Display of results

The main programme displays the results of the simulated impact numbers in a table on Stata result screen.

Saved outputs

The simulated distributions and the calculated impact numbers are saved onto Stata dataset. The summary of impact numbers (means, medians, [95%] confidence intervals) are saved in a text file.

Stata codes

```
*! Simulate Impact Numbers V 0.1
*! Developed as part of the IMI-PROTECT Consortium

mata:
function _simulin(real scalar reps, real scalar iu, real scalar rr, /*
*/ real scalar rrll, real scalar rrul, |real scalar pe, real scalar pd, /*
*/ real scalar num, real scalar nu1, real scalar nu2, real scalar nu3, /*
*/ real scalar sdn, real scalar _revalp, real scalar alph1, real scalar alph2, /*
*/ string myfile, real scalar multiparse, string multifile)
{
real matrix ru1

/* Draw baseline rates */
ru1 = rbinomial(reps,1,nu1,iu) :/ nu1


/* Draw RR */
lrr = log(rr)
slrr = (log(rrul)-log(rrll))/(2*invnormal(_revalp))
rho = exp(rnormal(reps,1,lrr,slrr))

newvars = st_addvar("double", ("iu","rr"))
initsum = ru1, rho

if (pe!=-1) {
      /* Draw pe rates */
      re = rbinomial(reps,1,nu2,pe) :/ nu2
      newvars01 = st_addvar("double", ("pe"))
      newvars = newvars, newvars01
      initsum = initsum, re
}

if (pd!=-1) {
      /* Draw pd rates */
      if (pd==1){
            pd = 0.9999999
      }
      rd = rbinomial(reps,1,nu3,pd) :/ nu3
      newvars02 = st_addvar("double", ("pd"))
      newvars = newvars, newvars02
      initsum = initsum, rd
}

if (num!=-1) {
      /* Use fixed n or draw n from Normal dist */
      if (sdn==-1){
            ns = num
      }
      else {
            ns = rnormal(reps,1,num,sdn)
      }
}

/*----------------------------------------------------
 Calculate impact numbers
----------------------------------------------------*/
real matrix results, ar, ein, afe, ecin

/* Calculate the risk reduction */
rid = rho:-1
```

```
/*** Attributable risk (AR) ***/
ar = ru1:*rid

/*** Exposure impact number (EIN) or the NNT ***/
ein = editmissing(1:/ar, max(1:/ar))

/*** Attributable fraction among exposed (AFE) ***/
afe = rid:/rho

/*** Exposed cases impact number (ECIN) ***/
ecin = editmissing(1:/afe, max(1:/afe))

results = ar, ein, afe, ecin

if (pe!=-1) {
      real matrix par, cin, din
      /*** Population attributable risk (PAR) ***/
      par = (re:*rid):/(1:+(re:*rid))

      /*** Case impact number (CIN) ***/
      cin = editmissing(1:/par, max(1:/par))

      /*** Disease impact number (DIN) ***/
      din = editmissing(1:/(ar:*re), max(1:/(ar:*re)))

      results = results, par, cin, din

      if (pd!=-1) {
            real matrix pin
            /*** Population impact number (PIN) ***/
            pin = editmissing(din:/rd, max(din:/rd))

            results = results, pin

            if (num!=-1) {
                  real matrix pinert, nepp
                  /*** Population impact number of eliminating a risk factor over time
t (PIN-ER-t) ***/
                  pinert = ns:*ru1:*par

                  /*** Number of events prevented in a population (NEPP) ***/
                  nepp = ns:*re:*rd:*ru1:*rid

                  results = results, pinert, nepp
            }
      }
}


pim1 = mean(results)
pim2 = mm_quantile(results,1,(0.50 \ alph1 \ alph2))
pim = pim1 \ pim2

st_addobs(reps)
newvars1 = st_addvar("double", ("ar","ein","afe","ecin"))
newvars = newvars,newvars1
vname = "ar","ein","afe","ecin"
if (pe!=-1) {
      newvars2 = st_addvar("double", ("par", "cin", "din"))
      newvars = newvars, newvars2
      vname = vname, "par", "cin", "din"
      if (pd!=-1) {
            newvars3 = st_addvar("double", ("pin"))
            newvars = newvars, newvars3
            vname = vname, "pin"
            if (num!=-1) {
                  newvars4 = st_addvar("double", ("pinert", "nepp"))
                  newvars = newvars, newvars4
                  vname = vname, "pinert", "nepp"
            }
      }
}

st_store(.,newvars, (initsum,results))

vname = vname', strofreal(pim')

/* Display results */
printf("\n Population Impact Measures Simulation Results \n")
printf("\n{txt}{space 13}{c |}         Mean        Median        %5.0g%%        %5.0g%%
\n",alph1*100, alph2*100)
printf("{hline 13}{c +}{hline 55}\n")
for (i=1;i<=rows(vname);i++){
      printf("{txt}%12s {c |} {res}%12s  %12s  %12s  %12s \n", vname[i,1], vname[i,2],
vname[i,3], vname[i,4], vname[i,5])
}

if (myfile!="_EMPTY_"){
      if (multiparse==0) {
            fh = fopen(myfile,"w")
            varn = sprintf("%12s\t%12s\t%12s\t%12s\t%12s",
"Measure","Mean","Median","LL","UL")
            fput(fh,varn)
            for (i=1; i<=rows(vname); i++) {
            outres = sprintf("%12s\t%12s\t%12s\t%12s\t%12s", vname[i,1], vname[i,2],
vname[i,3], vname[i,4], vname[i,5])
            fput(fh,outres)
            }
      }
      else if (multiparse>=1) {
            if (multifile=="_EMPTY_"){
                  multifile = "myresults.txt"
            }
            namestr = substr(myfile,1,strpos(myfile,".")-1)
            fh = fopen(multifile,"a")
            if (multiparse==1) {
                  varn = sprintf("%15s\t%12s\t%12s\t%12s\t%12s\t%12s","Name",
"Measure","Mean","Median","LL","UL")
                  fput(fh,varn)
            }
            for (i=1; i<=rows(vname); i++) {
                  outres = sprintf("%15s\t%12s\t%12s\t%12s\t%12s\t%12s", namestr,
vname[i,1], vname[i,2], vname[i,3], vname[i,4], vname[i,5])
                  fput(fh,outres)
            }
      }
      fclose(fh)
}
}
mata mosave _simulin(), replace
end
```

1.2 Simulation scenario 1 - "immediate" version i.e. manual numeric input required

Purpose of development

The main interface for the impact number simulation is the programme –*pims*–. The parameters in Table 1 are to be specified through this programme.

Description of arguments

The order of arguments in the main part of –*pims*–, at the minimum, is $I_u, RR, RR_{lci}, RR_{uci}$ which will calculate AR, EIN, AFE, and ECIN.

When the arguments contain five parameters, $I_u, RR, RR_{lci}, RR_{uci}, p_e$, it will additionally calculate PAR, CIN, and DIN.

When the probability of disease is also specified, i.e. $I_u, RR, RR_{lci}, RR_{uci}, p_e, p_d$, it will additionally calculate PIN.

When the final argument n, the size of population of interest, is included i.e. $I_u, RR, RR_{lci}, RR_{uci}, p_e, p_d, n$ then it will also calculate the PIN-ER-*t* and NEPP. At the moment, PIN-ER-*t* is displayed as "pinert" without any distinction to the value of t. Users are advised to take note of the time duration of the evidence data.

Evidence from single study can be entered directly into the programme. For multiple studies, evidence data are to be combined appropriately when using this programme e.g. through meta-analyses.

Programme defaults – assumptions on parameters

–*pims*– makes some assumptions in the sampling when the optional parameters are not specified. The following are the defaults used by –*pims*– to be passed to –*_simulin()*–:

i. Number of simulations

By default, 10000 simulations are run.

ii. Evidence data assumptions

We work on the assumptions that the specified parameters come from populations of size 10000 when they are not specified, i.e. $n_{sample} = v_1 = v_2 = v_3 = 10000$. These can be and are advised to be specified to obtain better results.

iii. Confidence intervals

Wherever applicable, we assume that the confidence intervals are at the 95% level. This can be changed through the options in the programme.

Stata codes

```
*! Version 1.0 11Nov2011
*! Developed as part of the IMI-PROTECT Consortium

*-----------------------------------------------------
* Main simulation: second order Monte Carlo
*-----------------------------------------------------
cap prog drop pims
prog define pims, rclass
version 12
syntax anything ///
      [, reps(integer 10000) SEED(real 7903) ///
      ALPHA(real 0.05) SAMPLEAlpha(real 0.05) ///
      BASen(integer 10000) PEN(integer 10000) ///
      PDN(integer 10000) SDN(real -1) FILEDROP ///
      SAVing(name) SUMmary(string) REPLACE CLEAR ///
      MULTI(real 0) SUMName(string) ///
      ]

if "`filedrop'"!=""{
      cap rm "`summary'"
}

set seed `seed'

* Order of anything: iu, rr, rrll, rrul, pe, pd, n

** Check main input parms
local parmc: word count `anything'
if `parmc'<4{
      di in red "Too few parameters"
      exit
}
if `parmc'>7{
      di in red "Too many parameters"
      exit
}
tokenize `anything'
if !(`2'>=`3' & `2'<=`4' & `3'<=`4') {
      di in red "rr, ll, ul must be in that order"
      exit
}

forvalues x=5/7{
      if "``x''"==""{
            local `x' = -1
      }
}
confirm integer number `7'

** Significance level
local a = `alpha'/2
local b = 1-`a'
local _revalp = 1-`samplealpha'/2

** Simulate and display results
`clear'
qui ds
if "`saving'"=="" & "`r(varlist)'"!=""{
      di in red "Must specify option clear when dataset is not empty. This will replace
data in memory."
      exit
}
else if "`saving'"!="" & "`r(varlist)'"!="" {
      preserve
      clear
}
if "`summary'"==""{
      local summary "_EMPTY_"
}
if "`sumname'"==""{
      local sumname "_EMPTY_"
}

cap drop ar ein afe ecin
cap drop par cin din
cap drop pin
cap drop pinert nepp

mata: _simulin(`reps', `1', `2', `3', `4', `5', `6', `7', /*
*/ `basen', `pen', `pdn', `sdn', `_revalp', `a', `b', /*
*/ "`summary'", `multi', "`sumname'")

if "`saving'"!="" {
      save `saving', `replace'
}

*return matrix pims=pims
end
```

1.3 Simulation scenario 2 – raw numbers (frequencies) are used as arguments

Purpose of development

We build an additional programme, called –*pimsraw*–, to calculate the required parameters as in Scenario 1 which will be fed into the existing programme for simulation and calculation. This still assumes that evidence come from a single trial for the impact numbers calculations.

The engine for the simulation is as defined in above in Scenario 1.

Description of arguments

This add-on which allows users to input raw data in the following two formats:

i. *var_case var_exposed*

With the two variables input in the stated order, the programme calls – *cs* – in Stata to calculate the relative risk when exposure time is equal. The programme also accepts the optional "if", "in", as well as frequency weights arguments in the command line. Other options for – *cs* – as documented in Stata help file for the command may also be specified, with the exception of "by" option which will be ignored if specified.

ii. *var_case var_exposed var_time*

With the three variables input in the stated order, the programme calls – *ir* – in Stata to calculate the relative risk when exposure time is unequal (person-time). The programme also accepts the optional "if", "in", as well as frequency weights arguments in the command line. Other options for – *ir* – as documented in Stata help file for the command may also be specified, with the exception of "by" option which will be ignored if specified.

By default the baseline risk is estimated from the data, unless specified.

Stata codes

```
*! Version 1.0 11Nov2011
*! Developed as part of the IMI-PROTECT Consortium

*----------------------------------------------------
* Add-on 1: Simulation using raw data
*----------------------------------------------------
prog define pimsraw
version 12
syntax varlist(min=2 max=3) [if] [in] [fw] ///
      [, RAWOPTions(string) ///
      BASERate(numlist max=1) QUIetly ///
      PE(real -1) PD(real -1) num(real -1) * ///
      ]

* the same as iri and csi command
tokenize `varlist'
if "`3'"!=""{
      local cmd "ir"
}
else {
      local cmd "cs"
}

`quietly' `cmd' `varlist' `if' `in' [`weight'`exp'], `rawoptions'
if "`baserate'"!=""{
      local 1 = `baserate'
}
else {
      if "`cmd'"=="ir" {
            local 1 = r(ird)/(r(irr)-1)
      }
      else if "`cmd'"=="cs" {
            local 1 = r(rd)/(r(rr)-1)
      }
}
if "`cmd'"=="ir" {
      local 2 = r(irr)
      local 3 = r(lb_irr)
      local 4 = r(ub_irr)
}
else if "`cmd'"=="cs" {
      local 2 = r(rr)
      local 3 = r(lb_rr)
      local 4 = r(ub_rr)
}

pims `1' `2' `3' `4' `pe' `pd' `num', `options'

end
```

1.4 Simulation scenario 3 – extension to evidence from multiple trials

Purpose of development

We build another additional programme, called –*pimsmulti*– to simulate impact numbers from evidence from different trials and for a number of criteria (≥ 1). The idea is to make simulations for many criteria seem seamless. The parameters required as in Scenario 1 will be calculated separately for each criterion, and then fed into the existing –*pims*– programme.

Description of arguments

This programme is also an add-on to –*pims*– which takes multilevel data in a structured dataset. The dataset that is used to run –*pimsmulti*– should contain the following at a minimum:

i. Four variables with the number of events in the treatment and control group as *trt_event trt_noevent ctrl_event ctrl_noevent*

ii. A string variable containing the names of the criteria

–*pimsmulti*– calls Stata's –*metan*– command to run meta-analysis based on the four variables to calculate the relative risk for each criteria, assuming that studies (data source) are unique within each criterion. Two variables relating to p_e and p_d may also be specified to calculate the latter impact numbers. Two further variables relating to the size of sample where p_e and p_d came from may also be specified. These variables should be unique (replicated) for each criterion; and are specified in the programme's "options". The size of population of interest, n, is constant for all criteria, therefore is entered as a scalar in the "options".

Saved outputs

The results at each iteration are saved in Stata data files in current working directory separately for each criterion. The summary of the impact numbers are collated and saved in a text file (.txt).

Stata codes

```
*! Version 1.0 11Nov2011
*! Developed as part of the IMI-PROTECT Consortium

*-------------------------------------------------------
* Add-on 2: Simulation using multiple endpoints/studies
*-------------------------------------------------------
prog define pimsmulti
version 12
syntax varlist(min=4 max=4),  BY(varlist max=1) ///
      [SUMmary(string) SAVing(name) REMove SUMNAme(string) ///
      QUIetly BASERate(varlist max=1) FILEDROP ///
      RANDOM FIXED METANOPTions(string) SCALEbase(real 1) ///
      PE(numlist max=1) PD(numlist max=1) num(real -1) * ///
      ]
confirm string variable `by'

if "`sumname'"!="" {
      local myfilen "`sumname'"
}
else{
      local myfilen "myresults.txt"
}


cap confirm file "`myfilen'"
if !_rc {
      if "`remove'"!="" {
            rm "`myfilen'"
      }
      else{
            di in red "Previous simulation results in file -`myfilen'- already exist"
            exit
      }
}

qui{

tempvar bvar
encode `by', gen(`bvar')
noi label list `bvar'

sum `bvar'
local counter = r(max)

tokenize `varlist'
            tempvar one two
            *gen `one' = 1
            *metan `3' `4' `one' `one', nograph `fixed' `random' `metanoptions'
            *local iu = r(ES)
            *replace `one' = `3' + `4'
            gen `one' = `3'+`4'
            gen `two' = `3'/`one'

forvalues x=1/`counter' {
      *preserve
      *drop if `bvar'!=`x'

      metan `varlist' if `bvar'==`x', nograph `fixed' `random' `metanoptions'
      local rr = r(ES)
      local ll = r(ci_low)
      local ul = r(ci_upp)
```

```
    if "`baserate'"!=""{
        sum `baserate' if `bvar'==`x'
        local iu = r(min)
    }
    else {
        sum `one' if `bvar'==`x'
        local basen = r(sum)
        local basen "basen(`basen')"
        sum `two' if `bvar'==`x'
        local iu = `scalebase’*max(r(min), 0.0005)
    }

    local postfix = regexr(proper(itrim(substr("`:label `bvar' `x''",1,12)))," ","")
    local _sav "`saving'`postfix'"
    local _sumn "`summary'`postfix'"
    if "`filedrop'"!=""{
        cap rm "`_sav'.dta"
        *cap rm "`_sumn'.txt"
    }

    noi `quietly' pims `iu' `rr' `ll' `ul' `pe' `pd' `num', ///
    `options' summary("`_sumn'.txt") `basen' ///
    saving("`_sav'") multi(`x') sumname(`myfilen')
    *restore
}
}
end
```