**The natural history of *Chlamydia trachomatis* infection in women: a multi-parameter evidence synthesis**

**MJ Price1, AE Ades2\*, K Soldan3, NJ Welton2, J Macleod2, I Simms3,**

**D De Angelis3,4, KME Turner2, PJ Horner2,5**

\**Correspondence to Professor AE Ades, School of Social and Community Medicine, University of Bristol, Canynge Hall, 39 Whatley Rd, Bristol, BS8 2PS, U.K.*

**WinBUGS code**

1 School of Health and Population Sciences, University of Birmingham, Birmingham, UK

2 School of Social and Community Medicine, University of Bristol, Bristol, UK

3 Public Health England (formerly Health Protection Agency), Colindale, London, UK

4 Medical Research Council Biostatistics Unit, Cambridge, UK

5 Bristol Sexual Health Centre, University Hospital Bristol NHS Foundation Trust, Bristol, UK

# Contents

[Contents 2](#_Toc445925899)

[Appendix 4.2 2](#_Toc445925900)

[WinBUGS code and Data for the 2-rate model 2](#_Toc445925901)

[Appendix 5.1 5](#_Toc445925902)

[WinBUGS code and data for the Chapter 5 analysis 5](#_Toc445925903)

[Appendix 6.3 10](#_Toc445925904)

[WinBUGS code for the 2-rate model 10](#_Toc445925905)

[Appendix 7.2 19](#_Toc445925906)

[WinBUGS code for PID incidence synthesis 19](#_Toc445925907)

[Appendix 8.1 21](#_Toc445925908)

[WinBUGS code for cumulative PID exposure 21](#_Toc445925909)

[Appendix 9.1 38](#_Toc445925910)

[WinBUGS code for prediction of EP rates in Table 35 38](#_Toc445925911)

[Appendix 10.2 51](#_Toc445925912)

[WinBUGS code and data files for prediction of TFI prevalence 51](#_Toc445925913)

[Appendix 11.2 62](#_Toc445925914)

[WinBUGS code applied to serology data 62](#_Toc445925915)

# Appendix 2

#### WinBUGS code and Data for the 2-rate model

**model {**

**# likelihood**

for (i in 1:studnum) {

 for (j in 1:studobs[i]) {

 r[i,j] ~ dbin(theta[i,j],n[i,j])

 }

 }

**# model**

for (i in 1:studnum-2) {

 for (j in 1:studobs[i]) {

 theta[i,j] <- ((z[i,j] / n[i,j]) \* (1 - exp(-lambda.C[1] \* t[i,j])) + (1 - (z[i,j] / n[i,j])) \* (1 - exp(-lambda.C[2] \* t[i,j])))

 **/** (psi / (1 - ((1-psi) \* equals(seind[i,j],0))))

 }

 }

**# Left-truncated studies with repeat observations**

for (i in 8:9) {

 for (j in 1:studobs[i]) {

 temp[i,j,1] <- ((z[i,1] / n[i,1]) \* exp(-lambda.C[1] \* T[i,j]) /

 ((z[j,1] / n[i,1]) \* exp(-lambda.C[1] \* T[i,j]) +

 (1 - (z[i,1] / n[i,1])) \* exp(-lambda.C[2] \* T[i,j]))) \*

 (1 - exp(-lambda.C[1] \* t[i,j]))

 temp[i,j,2] <- ((1 - (z[i,j] / n[i,1])) \* exp(-lambda.C[2] \* T[i,j]) /

 (**0.00001+**(z[i,1] / n[i,1]) \* exp(-lambda.C[1] \* T[i,j]) +

 (1 - (z[i,1] / n[i,1])) \* exp(-lambda.C[2] \* T[i,j]))) \*

 (1 - exp(-lambda.C[2] \* t[i,j]))

 theta[i,j] <- (temp[i,j,1] + temp[i,j,2]) **/**

 (psi / (1 - ((1-psi) \* equals(seind[i,j],0))))

 }

 }

**# priors**

p1 ~ dbeta(1,1)

lambda.C[1] <- 120

lambda.C[2] ~ dexp(0.001)

psi ~ dbeta(78,8) #sensitivity of culture given initial positive culture

**# Class proportions**

# t=0 studies

for (i in 1:4) {

 for (j in 1:studobs[i]) {

 z[i,j] ~ dbin(p1,n[i,j]) # start at t=0

 }

 }

# Left-truncated studys

for (i in 5:studnum) {

 for (j in 1:studobs[i]) {

 z[i,j] ~ dbin(w1,n[i,j])

 }

 }

**# deviance**

for (i in 1:studnum) {

 for (j in 1:studobs[i]) {

 dev[i,j] <- 2 \* (r[i,j] \* log(r[i,j] / (theta[i,j] \* n[i,j])) +

 (n[i,j] - r[i,j]) \* log((n[i,j] - r[i,j]) /

 (n[i,j] - (n[i,j] \* theta[i,j]))))

 }

 dev.stud[i] <- sum(dev[i,1:studobs[i]])

 }

sumdev <- sum(dev.stud[])

# left truncation

w1 <- (p1 / lambda.C[1]) / (p1 / lambda.C[1] + (1 - p1) / lambda.C[2])

**# summary statistics**

dur <- 1 / lambda.C[2]

**# Predicted values for Forest plot**

for (i in 1:studnum) {

 for (j in 1:studobs[i]) {

 stud.lambda.Cexpect[i,j] <- -log(1 - theta[i,j]) / t[i,j]

 stud.dur.expect[i,j] <- 1 / stud.lambda.Cexpect[i,j]

 }

 }

**}**

**# Data**

list(

**# duration**

**# study order**

# 1 Johhanisson

# 2 Joyner

# 3 Geisler

# 4 Paavonen

# 5 Rahm

# 6 Sorensen

# 7 McCormack

# 8 Morre

# 9 Mollano

r = structure(.Data=c(

10,7,6,6,NA,

2,7,1,0,3,

23,NA,NA,NA,NA,

3,NA,NA,NA,NA,

17,0,0,NA,NA,

8,NA,NA,NA,NA,

3,NA,NA,NA,NA,

2,2,4,0,2,

44,23,7,2,NA

),.Dim=c(9,5)),

n = structure(.Data=c(

23,14,14,8,NA,

12,28,4,8,6,

129,NA,NA,NA,NA,

15,NA,NA,NA,NA,

93,1,1,NA,NA,

13,NA,NA,NA,NA,

7,NA,NA,NA,NA,

20,5,15,1,13,

82,37,14,6,NA

),.Dim=c(9,5)),

t=structure(.Data=c(

0.038,0.058,0.077,0.125,NA,

0.012,0.03,0.049,0.088,0.274,

0.045,NA,NA,NA,NA,

0.083,NA,NA,NA,NA,

0.25,0.5,0.75,NA,NA,

1,NA,NA,NA,NA,

1.375,NA,NA,NA,NA,

0.083,0.5,0.417,0.917,0.5,

1,1,1,1,NA

),.Dim=c(9,5)),

seind = structure(.Data=c(

1,1,1,1,NA,

0,0,0,0,0,

0,NA,NA,NA,NA,

1,NA,NA,NA,NA,

1,1,1,NA,NA,

0,NA,NA,NA,NA,

1,NA,NA,NA,NA,

0,0,0,0,0,

0,0,0,0,NA

),.Dim=c(9,5)),

T=structure(.Data=c(

NA,NA,NA,NA,NA,

NA,NA,NA,NA,NA,

NA,NA,NA,NA,NA,

NA,NA,NA,NA,NA,

NA,NA,NA,NA,NA,

NA,NA,NA,NA,NA,

NA,NA,NA,NA,NA,

0,0,0.083,0.083,0.5,

0,1,2,3,NA

),.Dim=c(9,5)),

studnum = 9,

studobs = c(4,5,1,1,3,1,1,5,4),

)

**# Initial values - 1**

list(

psi = 0.9,

lambda.C = c(NA,0.7),

p1 = 0.2,

)

**# Initial values - 2**

list(

psi = 0.6,

lambda.C = c(NA,0.1),

p1 = 0.5

)

# Appendix 5

#### WinBUGS code and data for the Chapter 5 analysis

**model {**

**# Duration**

**# Duration of asymptomatic infection**

dura ~ dnorm(0,0.0001) I(0,)

D ~ dnorm(dura,pr.D)

dev.D <- pr.D\*pow((dura-D),2) # Duration Deviance

lambdaC <- 1/dura # clearance rate

**# Duration of symptomatic infection**

durs ~ dunif(0.0767,0.1533) **# symptomatic, 4-8 weeks**

**#**durs ~ dunif(0.0577,0.2308) **#symptomatic, 3-12 weeks sensitivity analysis**

cut.durs <- cut(durs) # pevents updating of durs

lambdaS <- 1 / cut.durs # clearance rate

**# Probability ct is symptomatic**

r.phi ~ dbin(phi,n.phi) phi ~ dbeta(1,1)

dev.phi <- 2 \* (r.phi \* log(r.phi / (phi \* n.phi)) + (n.phi - r.phi) \*

 log((n.phi - r.phi) / (n.phi - (n.phi \* phi)))) **# Deviance**

**# Mean duration of infection**

dur <- phi\*cut.durs + (1-phi)\*dura

**# Incidence model**

**# Likelihood and model**

for (i in 1:2) { # loop over rate, 1=infection, 2=reinfection

 for (s in 1:3) { # loop over setting - GP=1, fP=2, GUM=3

 for (a in 1:3) { # loop over age group:1=16,17;2=18-20, 3=21-24

 lambda[i,s,a] <- equals(i,1)\*(gamma[a] \* rho[s] \* lambda111) +

 equals(i,2)\*(gamma[a] \* rho[s] \* lambda111 \* eta[s])

 r.inc[i,s,a] ~ dbin(theta.inc[i,s,a],n.inc[i,s,a])

 theta.inc[i,s,a] <- phi \* (

 (1 - (lambdaS + lambda[i,s,a] \*

 exp(-(lambdaS + lambda[i,s,a]) \* 0.5)) /

 (lambdaS + lambda[i,s,a]))

 ) + (1 - phi) \* (

 (1 - (lambdaC + lambda[i,s,a] \*

 exp(-(lambdaC + lambda[i,s,a]) \* 0.5)) /

 (lambdaC + lambda[i,s,a]))

 )

 dev.inc.rat[i,s,a] <- 2 \* (r.inc[i,s,a] \* log(r.inc[i,s,a] /

 (theta.inc[i,s,a] \* n.inc[i,s,a])) + (n.inc[i,s,a]-

 r.inc[i,s,a]) \* log((n.inc[i,s,a] - r.inc[i,s,a]) /

 (n.inc[i,s,a] - (n.inc[i,s,a] \* theta.inc[i,s,a]))))

 }

 }

 }

sumdev.inc.rat <- sum(dev.inc.rat[ , , ])  **# Deviance,** # LaMontagne

**#priors for log incidence parameters**

loglambda111 ~ dnorm(0,.0001) #Age 16-17, GP, infection

log(lambda111) <- loglambda111

for (s in 1:3) {

 logeta[s] ~ dnorm(0,.0001) # GP=1, fP=2, GUM=3

 log(eta[s]) <- logeta[s]

 }

logrho[1] <- 0 # GPtheta.inc

for (s in 2:4) {

 logrho[s] ~ dnorm(0,.0001) # 2=fp, 3=GUM, 4=pop, rel to GP

 }

for (s in 1:4) {

 log(rho[s]) <- logrho[s]

 }

loggamma[1] <- 0 #1 = 16-17

for (a in 2:5) {

 loggamma[a] ~ dnorm(0,.01) #2=18-19, 3=20-24,4=25-29,5=30-44 rel to 16-17

}

for (a in 1:5) {

 log(gamma[a]) <- loggamma[a]

 }

**# setting specific Odds ratios from Adams**

for (s in 1:3) { #s+1= 2=FP, 3=GUM, 4=pop, rel to GP

 r.OR[s] ~ dnorm(logrho[s+1],pr.OR[s])

 dev.OR[s] <- pr.OR[s] \* pow((r.OR[s]-logrho[s+1]),2) # OR deviance

}

sumdev.OR <- sum(dev.OR[])

**# overall age-specific incidence in population, 1=16-17, 2=18-19, 3=20-24,**

**# 4=25-29, 5=30-44**

for (a in 1:3) {

 p[a] ~ dbeta(aprior[a],bprior[a])

 cut.p[a] <- cut(p[a]) **# prevents updating p[a]**

 lamda.F[a] <- rho[4]\*((1-cut.p[a])\*lambda[1,1,a] +

 cut.p[a]\*lambda[2,1,a])

 lamda.pop[a] <- lamda.F[a] / (1 - lamda.F[a] \* dur)

 }

for (a in 4:5) {

 lamda.pop[a] <- lamda.pop[1] \* gamma[a]

 }

**# Age-specific Prevalence from Adams, 1=18-19, 2=20-24, 3=25-29, 4=30-44**

for (a in 1:4) {

 Lprev[a]~dnorm(lprev[a+1],pr.Lprev[a])

 dev.prev[a] <- pr.Lprev[a] \* pow((Lprev[a]-lprev[a+1]),2)

 }

for (a in 1:5) {

 lprev[a] <- logit(prev[a])

 }

for (a in 1:3) {

 prev[a] <- min(.999,(dur \* lamda.pop2[a]))

}

prev[4] <- prev[3] \* gamma[4]/gamma[3]

prev[5] <- prev[3] \* gamma[5]/gamma[3]

sumdev.prev <- sum(dev.prev[])

sumdev <- dev.D + sumdev.inc.rat + sumdev.OR + sumdev.prev + dev.phi

**# incidence in difference age bands, using population weights N[]**

for (a in 16:17) {inc[a] <- lamda.pop[1]}

for (a in 18:20) {inc[a] <- lamda.pop[2]}

for (a in 21:24) {inc[a] <- lamda.pop[3]}

for (a in 25:29) {inc[a] <- lamda.pop[4]}

for (a in 30:44) {inc[a] <- lamda.pop[5]}

lamda.pop2[1] <- lamda.pop[1] # age 16,17

lamda.pop2[2] <- lamda.pop[2] # age 18,19.

lamda.pop2[3] <- inprod(inc[20:24],N[20:24])/sum(N[20:24]) # age 20-24

inc1624 <- inprod(inc[16:24],N[16:24])

inc2544 <- inprod(inc[25:44],N[25:44])

inc1644 <- inprod(inc[16:44],N[16:44])

lamda.pop2[4] <- inc1624 / sum(N[16:24]) # age 16-24

lamda.pop2[5] <- inc2544 / sum(N[25:44]) # age 25-44

lamda.pop2[6] <- inc1644 / sum(N[16:44]) # age 16-44

**# other population summaries**

prop.treat.1624.2002 <- 31510 / inc1624

prop.treat.1624.2003 <- 34660 / inc1624

prop.asymp.clin.1624.2003 <- 1 - (phi / prop.treat.1624.2003)

**# prevalence in reconstructed age groups**

for (a in 1:6) {

 prev.pop2[a] <- dur \* lamda.pop2[a]

 }

**}** **# end of program**

**Data**

list(

**# duration**

D=1.36, pr.D=59.17, # from Duration paper 2-Class estimate

**# incidence**

r.inc=structure(.Data=c(

4,3,4, 9,5,7, 5,16,9,

5,7,10, 13,12,5, 6,15,5

),.Dim=c(2,3,3)),

n.inc=structure(.Data=c(

73,195,188, 194,273,201, 102,235,245,

14,65,79, 95,127,63, 40,139,81

),.Dim=c(2,3,3)),

**# priors for infection/reinfection weights, 16-17, 18-20, 21-24**

aprior=c(46.5,73.1,106.4), bprior=c(400,634.6,938.4),

**# Logit Prevalence**

Lprev=c(-2.987,-3.41,-4.185,-4.82),

pr.Lprev=c(19.3,20.8,18.4,17.2),

**# Setting-specific Log Odds Ratios. Fp,GUM, pop**

r.OR= c(0.239,0.871, -.511),

pr.OR= c(67.03,35.21, 17.28),

**# symptomatic ct - Geisler**

r.phi = 26, n.phi = 115,

**# Population sizes from census, age =1...44 - 2002**

N=c(NA,NA,NA,NA,NA, NA,NA,NA,NA,NA, NA,NA,NA,NA,NA,

305500,306300,296400,291400,294800, 310100,313900,305600,294700,295000,

304100,317000,329600,349600,370300, 380900,376900,387800,390900,399400,

401200,402600,398700,391900,381900, 370900,356200,349000,343800)

)

**# Initial values - 1**

**# Duration**

list(dura=1, durs=.115,

**# incidence**

loglambda111=-2.5,

logeta=c(1,.5,.5),

logrho=c(NA, 1, 1.5,-.5),

loggamma=c(NA,-.2,-.5,-.7,-.8),

**# probability symptomatic**

phi = 0.5,

**# proportion re-infection**

p=c(.1, .1, .1)

)

**# Initial values - 2**

**# Duration**

list(dura=.2, durs=.112,

**# incidence**

loglambda111=-4,

logeta=c(.5,1,1),

logrho=c(NA, 0, 0,0),

loggamma=c(NA,0,0,0,0),

**# probability symptomatic**

phi = .1,

**# proportion re-infection**

p=c(.3, .05, .2)

)

# Appendix 8

#### WinBUGS code for the 2-rate model

**model {**

**# chlamydia informative priors**

r.phi ~ dbin(phi,129)

lambda.c ~ dnorm(0.743,193)

dur.symp ~ dunif(0.077,0.15) **# 4 - 8 weeks**

lambda.t <- 1 / cut(dur.symp)

**# Prospective PID analysis constant progression rate**

**# Likelihood**

for (s in 1:4) {

 for (i in 1:2) {

 r[s,i] ~ dbin(p[s,i],n[s,i])

 }

 }

**# transistion probability calulations**

for (s in 1:4) {

 for (i in 1:2) {

 for(j in 1:3) {

 theta[s,1,index[i,j]] <- lambda[s,1,i,j]

 theta[s,2,index[i,j]] <- lambda[s,2,i,j]

 }

 }

 lambda[s,1,1,1] <- - lambda[s,1,1,2] - lambda[s,1,1,3]

 lambda[s,1,1,3] <- theta.CTpos1[s]

 lambda[s,1,2,2] <- - lambda[s,1,2,1] - lambda[s,1,2,3]

 lambda[s,1,2,3] <- theta.CTneg[s]

 lambda[s,2,1,1] <- - lambda[s,2,1,2] - lambda[s,2,1,3]

 lambda[s,2,1,3] <- theta.CTpos2[s]

 lambda[s,2,2,2] <- - lambda[s,2,2,1] - lambda[s,2,2,3]

 lambda[s,2,2,3] <- theta.CTneg[s]

 }

# Infection rate

for (s in 1:2) {

 lambda[s,1,2,1] <- **0.0**

 lambda[s,2,2,1] <- **0.0**

 }

for (s in 3:4) {

 lambda[s,1,2,1] <- **0.0**

 lambda[s,2,2,1] <- **0.0**

 }

# Clearance + treatment rate

lambda[1,1,1,2] <- lambda.c **+ 0.64**

lambda[1,2,1,2] <- lambda.c **+ 0.64**

lambda[2,1,1,2] <- lambda.c **+** 0.43

lambda[2,2,1,2] <- lambda.c **+** 0.43

lambda[3,1,1,2] <- lambda.c + 0.32

lambda[3,2,1,2] <- lambda.c + 0.32

lambda[4,1,1,2] <- lambda.c + **0.32**

lambda[4,2,1,2] <- lambda.c + **0.32**

**# Models for rates**

for (s in 1:4) {

 theta.CTpos1[s] <- alpha[s] + delta[1]

 theta.CTpos2[s] <- alpha[s] + delta[2]

 theta.CTneg[s] <- alpha[s]

 }

**# Other data**

r.sch ~ dbin(pi.sch,n.sch) **# Scholes - ct prevalence**

r.ost ~ dbin(pi.ost,n.ost) **# Ostergaard - ct prevalence**

**# Priors**

delta[1] ~ dexp(0.00001)

delta[2] ~ dexp(0.00001)

for (s in 1:4) {

 alpha[s] ~ dexp(0.00001)

 }

pi.sch ~ dbeta(1,1)

pi.ost ~ dbeta(1,1)

phi ~ dbeta(1,1)

**# Calculation of parameters in the likekihood**

for (s in 1:4) {

 solution1d[s,1,1:dim] <- three.state(init[1:dim],time,theta[s,1,1:n.par], origin, tol)

 solution1d[s,2,1:dim] <- three.state(init[1:dim],time,theta[s,2,1:n.par], origin, tol)

 for (i in 1:2) {

 for (z in 1:6) {

 vectorforwbdev[s,i,z] <- solution1d[s,1,z]

 vectorforwbdev[s,i,z+6] <- solution1d[s,2,z]

 }

 vectorforwbdev[s,i,13] <- round(t[s] \* 365)

 vectorforwbdev[s,i,14] <- B

 vectorforwbdev[s,i,15] <- lambda.c

 vectorforwbdev[s,i,16] <- lambda.t

 vectorforwbdev[s,i,17] <- phi

 vectorforwbdev[s,i,18] <- clinorscreen[s]

 vectorforwbdev[s,i,19] <- caseprop[s,i]

 p[s,i] <- generatep(vectorforwbdev[s,i,1:19])

 }

 caseprop[s,2] <- 0

 }

caseprop[1,1] <- 1

caseprop[2,1] <- 1

caseprop[3,1] <- pi.sch

caseprop[4,1] <- pi.ost

clinorscreen[1] <- 0 # clinic based study

for (s in 2:4) {

 clinorscreen[s] <- 1 # screening studies

 }

**# Residual Deviance**

for (s in 1:4) {

 for (i in 1:2) {

 dev[s,i] <- 2 \* (r[s,i] \* log(r[s,i] / (p[s,i] \* n[s,i])) +

 (n[s,i] - r[s,i]) \* log((n[s,i] - r[s,i]) /

 (n[s,i] - (n[s,i] \* p[s,i]))))

 }

 }

dev.sch <- 2 \* (r.sch \* log(r.sch / (pi.sch \* n.sch)) +

 (n.sch - r.sch) \* log((n.sch - r.sch) /

 (n.sch - (n.sch \* pi.sch))))

dev.ost <- 2 \* (r.ost \* log(r.ost / (pi.ost \* n.ost)) +

 (n.ost - r.ost) \* log((n.ost - r.ost) /

 (n.ost - (n.ost \* pi.ost))))

sumdev <- sum(dev[ , ]) + dev.sch + dev.ost

**# Results**

kappa <- (1 - phi) \* (

 (1 - (exp( - (lambda.c + delta[1]) \* (B / 365)))) \*

 delta[1] / (lambda.c + delta[1]) +

 exp( - (lambda.c + delta[1]) \* (B / 365)) \*

 delta[2] / (lambda.c + delta[2])

 ) +

 phi \* (

 (1 - (exp( - (lambda.t + delta[1]) \* (B / 365)))) \*

 delta[1] / (lambda.t + delta[1]) +

 exp( - (lambda.t + delta[1]) \* (B / 365)) \*

 delta[2] / (lambda.t + delta[2])

 )

**# proportion of PIDs prevented by annual screening**

**# assumes two-week delay between test and treatment**

for (i in 1:B) {

 temp1[i] <- phi \* (

 ((1 - exp(-(lambda.t + delta[1]) \* (i/365))) -

 (1 - exp(-(lambda.t + delta[1]) \* ((i-1)/365)))) \*

 (delta[1] / (lambda.t + delta[1])) \*

 (1 - (max(0,(i-14)) / 365))) +

 (1 - phi) \* (

 ((1 - exp(-(lambda.c + delta[1]) \* (i/365))) -

 (1 - exp(-(lambda.c + delta[1]) \* ((i-1)/365)))) \*

 (delta[1] / (lambda.c + delta[1])) \*

 (1 - (max(0,(i-14)) / 365)))

 }

for (i in B+1:379) {

 temp1[i] <- phi \* (

 ((1 - exp(-(lambda.t + delta[2]) \* (i/365))) -

 (1 - exp(-(lambda.t + delta[2]) \* ((i-1)/365)))) \*

 (delta[2] / (lambda.t + delta[2])) \*

 (1 - (max(0,(i-14)) / 365))) +

 (1 - phi) \* (

 ((1 - exp(-(lambda.c + delta[2]) \* (i/365))) -

 (1 - exp(-(lambda.c + delta[2]) \* ((i-1)/365)))) \*

 (delta[2] / (lambda.c + delta[2])) \*

 (1 - (max(0,(i-14)) / 365)))

 }

prop.prevent <- 1 - (sum(temp1[ ]) / kappa)

**# Bayesian p-value**

test <- delta[1] - delta[2]

B.p <- step(test)

**}**

**# Data**

list(

**# PID (1 month) prospective**

# 1. Rees

# 2. POPI

# 3. Scholes

# 4. Ostergaard

r=structure(.Data=c(

8,3,

7,1,

33,7,

20,9

),.Dim=c(4,2)),

n=structure(.Data=c(

67,62,

74,63,

1598,645,

487,443

),.Dim=c(4,2)),

t = c(0.125,1,1,1),

B = 60, # If this were set above 365 WBDEV code would need changing

r.sch = 44,

n.sch = 645,

r.ost = 43,

n.ost = 867,

r.phi=30,

**time = 0.00274,**

**# forward equations**

dim=6,origin=0,tol=1.0E-4, init=c(1,0,0, 0,1,0),n.par=6,

index=structure(.Data=c(1,2,3,

 4,5,6), .Dim=c(2,3))

)

**# Initial values - 1**

list(

**# Prospective**

phi = 0.23, lambda.c = 0.74, delta = c(0.2,0.1),

alpha = c(0.01,0.01,0.01,0.01),

pi.sch = 0.051, pi.ost = 0.07, dur.symp = 0.1

)

**# Initial values - 2**

list(

**# Prospective**

phi = 0.7, lambda.c = 2, delta = c(0.1,0.6),

alpha = c(0.15,0.15,0.15,0.15),

pi.sch = 0.2, pi.ost = 0.2, dur.symp = 0.145

)

***WBDiff Program to calculate the transition probabilities***

MODULE WBDiffThreeState;

 IMPORT

 WBDiffODEMath,

 Math;

 TYPE

 Equations = POINTER TO RECORD (WBDiffODEMath.Equations) END;

 Factory = POINTER TO RECORD (WBDiffODEMath.Factory) END;

 CONST

 nEq = 6;

 nSt = 4; (\* one higher as arrays start at zero\*)

 VAR

 **fact**-: WBDiffODEMath.Factory;

 PROCEDURE (e: Equations) Derivatives (IN theta, C: ARRAY OF REAL; n: INTEGER; t: REAL;

 OUT dCdt: ARRAY OF REAL);

 VAR

 index: ARRAY nSt, nSt OF INTEGER;

 BEGIN

(\* define index of parameters (look-up table) \*)

 index[1,1] := 0;

 index[1,2] := 1;

 index[1,3] := 2;

 index[2,1] := 3;

 index[2,2] := 4;

 index[2,3] := 5;

(\* define system of nEq Differential Equations \*)

 dCdt[index[1,1]]:= C[index[1,1]]\*theta[index[1,1]] + C[index[1,2]]\*theta[index[2,1]];

 dCdt[index[1,2]]:= C[index[1,1]]\*theta[index[1,2]] + C[index[1,2]]\*theta[index[2,2]];

 dCdt[index[1,3]]:= C[index[1,1]]\*theta[index[1,3]] + C[index[1,2]]\*theta[index[2,3]];

 dCdt[index[2,1]]:= C[index[2,1]]\*theta[index[1,1]] + C[index[2,2]]\*theta[index[2,1]];

 dCdt[index[2,2]]:= C[index[2,1]]\*theta[index[1,2]] + C[index[2,2]]\*theta[index[2,2]];

 dCdt[index[2,3]]:= C[index[2,1]]\*theta[index[1,3]] + C[index[2,2]]\*theta[index[2,3]];

 END Derivatives;

 PROCEDURE (equations: Equations) SecondDerivatives (IN theta, x: ARRAY OF REAL;

 numEq: INTEGER; t: REAL; OUT d2xdt2: ARRAY OF REAL);

 BEGIN

 **HALT**(126)

 END SecondDerivatives;

PROCEDURE (equations: Equations) Jacobian (IN theta, x: ARRAY OF REAL;

 numEq: INTEGER; t: REAL; OUT jacob: ARRAY OF ARRAY OF REAL);

 BEGIN

 **HALT**(126)

 END Jacobian;

PROCEDURE (f: Factory) New (option: INTEGER): WBDiffODEMath.GraphNode;

 VAR

 equations: Equations;

 node: WBDiffODEMath.GraphNode;

 BEGIN

 NEW(equations);

 node := WBDiffODEMath.New(equations, nEq);

 **RETURN** node

 END New;

 PROCEDURE **Install**\*;

 BEGIN

 WBDiffODEMath.Install(fact)

 END Install;

 PROCEDURE Init;

 VAR

 f: Factory;

 BEGIN

 NEW(f); fact := f

 END Init;

 BEGIN

 Init

 END WBDiffThreeState.

***WBDEV code to calcuate the Binomial likelihood probabilities of PID for the 2-rate model***

MODULE WBDevgeneratep;

 IMPORT

 WBDevScalar,

 Math;

 TYPE

 Function = POINTER TO RECORD (WBDevScalar.Node) END;

 Factory = POINTER TO RECORD (WBDevScalar.Factory) END;

 VAR

fact-: WBDevScalar.Factory;

 PROCEDURE (func: Function) DeclareArgTypes (OUT args: ARRAY OF CHAR);

 BEGIN

 args := "v";

 END DeclareArgTypes;

 PROCEDURE calculation (func: Function; OUT output: REAL);

 VAR

 term1, term2, omega, omega\_cum,pi\_hi, pi\_low: ARRAY 366 OF REAL;

 pi: ARRAY 4,366+1 OF REAL;

 p\_hi, p\_low: ARRAY 3,4 OF REAL;

 H,B,h,i,b: INTEGER;

 Hin, Bin, lambdaC, lambdaT, phi, clinicorscreen, caseprop: REAL;

 BEGIN

 (\* Read in the parameter values \*)

 p\_hi[1,1] := func.arguments[0][0].Value();

 p\_hi[1,2] := func.arguments[0][1].Value();

 p\_hi[1,3] := func.arguments[0][2].Value();

 p\_hi[2,1] := func.arguments[0][3].Value();

 p\_hi[2,2] := func.arguments[0][4].Value();

 p\_hi[2,3] := func.arguments[0][5].Value();

 p\_low[1,1] := func.arguments[0][6].Value();

 p\_low[1,2] := func.arguments[0][7].Value();

 p\_low[1,3] := func.arguments[0][8].Value();

 p\_low[2,1] := func.arguments[0][9].Value();

 p\_low[2,2] := func.arguments[0][10].Value();

 p\_low[2,3] := func.arguments[0][11].Value();

 Hin := func.arguments[0][12].Value();

 Bin := func.arguments[0][13].Value();

 lambdaC := func.arguments[0][14].Value();

 lambdaT := func.arguments[0][15].Value();

 phi := func.arguments[0][16].Value();

 clinicorscreen := func.arguments[0][17].Value();

 caseprop := func.arguments[0][18].Value();

 (\* Converts H and B to integer format \*)

 FOR i:= 1 TO 5000 DO

 IF (Hin = i) THEN;

 H := i;

 END;

 END;

 FOR i:= 1 TO 5000 DO

 IF (Bin = i) THEN;

 B := i;

 END;

 END;

 (\* Sets B to equal H if follow-up time is shorter than B - Note the program would need changing if a

 screening study with a follow-up time shorter than B was included \*)

 IF (H < B) THEN;

 B := H;

 END;

 (\* Calculates the proportion of cases in screening studies infected in the last c = 1 to C days \*)

 IF (clinicorscreen = 1) THEN;

 omega\_cum[0] := 0;

 FOR b := 1 TO B DO

 omega[b] := phi \* (Math.Exp( - lambdaT \* (b - 1) / 365) - Math.Exp( - lambdaT \* b / 365)) +

 (1 - phi) \* (Math.Exp( - lambdaC \* (b - 1) / 365) - Math.Exp( - lambdaC \* b / 365)); omega\_cum[b] := omega\_cum[b-1] + omega[b];

 END;

 (\* specifies the proportion of women in each state at time zero \*)

 pi\_hi[0] := caseprop \* omega\_cum[B];

 ELSE

 pi\_hi[0] := caseprop;

 END;

 pi[1,0] := caseprop;

 pi[2,0]:= 1 - pi[1,0];

 pi[3,0] := 0;

 pi\_low[0] := pi[1,0] - pi\_hi[0];

 (\* main analysis \*)

 FOR h := 1 TO B DO;

 pi[1,h] := pi\_low[h-1] \* p\_low[1,1] + pi\_hi[h-1] \* p\_hi[1,1] + pi[2,h-1] \* p\_hi[2,1];

 pi[2,h] := pi[2,h-1] \* p\_hi[2,2] + pi\_low[h-1] \* p\_low[1,2] + pi\_hi[h-1] \* p\_hi[1,2];

 pi[3,h]:= 1 - pi[2,h] - pi[1,h];

 IF (clinicorscreen =0) THEN;

 pi\_hi[h] := pi[1,h];

 pi\_low[h] := 0

 END;

 IF (clinicorscreen =1) THEN;

 term1[h] := 0;

 FOR i := h TO B DO

 term1[h] := term1[h] + omega[B+1-i] \* pi[1,0] \* Math.Power(p\_hi[1,1],h);

 END;

 term2[1] := 0;

 IF (h >= 2) THEN;

 FOR i := 1 TO h DO

 term2[h] := term2[h-1] + (pi[2,i-1] \* p\_hi[2,1]) \* Math.Power(p\_hi[1,1],h-i);

 END;

 END;

 pi\_hi[h] := term1[h] + term2[h];

 pi\_low[h] := pi[1,h] - pi\_hi[h];

 END;

 END;

IF (H > B) THEN;

 FOR h := B+1 TO H DO

 pi[1,h] := pi\_low[h-1] \* p\_low[1,1] + pi\_hi[h-1] \* p\_hi[1,1] + pi[2,h-1] \* p\_hi[2,1];

 pi[2,h] := pi[2,h-1] \* p\_hi[2,2] + pi\_low[h-1] \* p\_low[1,2] + pi\_hi[h-1] \* p\_hi[1,2];

 pi[3,h]:= 1 - pi[2,h] - pi[1,h];

 term1[h] := 0;

 FOR i := h+1-B TO h DO

 term2[h] := term2[h-1] + (pi[2,i-1] \* p\_hi[2,1]) \* Math.Power(p\_hi[1,1],h-i);

 END;

 pi\_hi[h] := term1[h] + term2[h];

 pi\_low[h] := pi[1,h] - pi\_hi[h];

 END;

 END;

 output := pi[3,H];

 END calculation;

 PROCEDURE (func: Function) Evaluate (OUT value: REAL);

 VAR

 output: REAL;

 BEGIN

 calculation(func, output);

 value := output;

 END Evaluate;

 PROCEDURE (f: Factory) New (option: INTEGER): Function;

 VAR

 func: Function;

 BEGIN

 NEW(func); func.Initialize; **RETURN** func;

 END New;

 PROCEDURE **Install**\*;

 BEGIN

 WBDevScalar.Install(fact);

 END Install;

 PROCEDURE Init;

 VAR

 f: Factory;

 BEGIN

 NEW(f); fact := f;

 END Init;

 BEGIN

 Init;

 END WBDevgeneratep.

# Appendix 10

#### WinBUGS code for PID incidence synthesis

model {

# Routine data

for (ag in 1:4) {

 r.routine[ag] ~ dbin(p.routine[ag],N.routine[ag])

 p.routine[ag] <- 1 - exp(-lambda.PID.diag[ag])

 lambda.PID.diag[ag] ~ dexp(0.0001)

 range.temp[ag] ~ dunif(0,range.max[ag])

 range[ag] <- cut(range.temp[ag])

 lambda.PID[ag] <- (lambda.PID.diag[ag] + (range[ag] / N.routine[ag])) /

 (1 – psi[1])

# output for Multivariate Normal #prior

 lnlambda.PID[ag] <- log(lambda.PID[ag])

 }

N.routine[1] <- sum(N[16:19])

N.routine[2] <- sum(N[20:24])

N.routine[3] <- sum(N[25:34])

N.routine[4] <- sum(N[35:44])

lambda.PID1624 <- (lambda.PID[1] \* N.routine[1] +

 lambda.PID[2] \* N.routine[2]) / sum(N[16:24])

lambda.PID2544 <- (lambda.PID[3] \* N.routine[3] +

 lambda.PID[4] \* N.routine[4]) / sum(N[25:44])

lambda.PID1644 <- (lambda.PID[1] \* sum(N[16:19]) +

 lambda.PID[2] \* sum(N[20:24]) +

 lambda.PID[3] \* sum(N[25:34]) +

 lambda.PID[4] \* sum(N[35:44])) /

 sum(N[16:44])

RatioofPIDnums <- (lambda.PID2544 \* (N.routine[3] + N.routine[4])) /

 (lambda.PID1624 \* (N.routine[1] + N.routine[2]) +

 lambda.PID2544 \* (N.routine[3] + N.routine[4]))

# Wolner Hanssen

r.wh.undiagpop ~ dbin(psi[1],n.wh.all)

r.wh.asymp ~ dbin(psi[2],n.wh.undiag)

log(lgpsi[1]) <- psi[1]

log(lgpsi[2]) <- psi[2]

logit(lgtpsi[1]) <- psi[1]

logit(lgtpsi[2]) <- psi[2]

psi[1] ~ dbeta(1,1)

psi[2] ~ dbeta(1,1)

# POPI data

r.POPI ~ dbin(p.POPI,n.POPI)

p.POPI <- 1 - exp(-lambda.POPI)

lambda.POPI <- lambda.PID1624 \* (1 - (psi[1] \* psi[2]))

# Residual Deviance

for (ag in 1:4) {

 dev[ag] <- 2 \* (r.routine[ag] \* log(r.routine[ag] / (p.routine[ag] \*

 N.routine[ag])) + (N.routine[ag] - r.routine[ag]) \*

 log((N.routine[ag] - r.routine[ag])

 / (N.routine[ag] - (N.routine[ag] \* p.routine[ag]))))

 }

dev[5] <- 2 \* (r.wh.undiagpop \* log(r.wh.undiagpop / (psi[1] \* n.wh.all)) +

 (n.wh.all - r.wh.undiagpop) \* log((n.wh.all - r.wh.undiagpop)

 /(n.wh.all - (n.wh.all \* psi[1]))))

dev[6] <- 2 \* (r.wh.asymp \* log(r.wh.asymp / (psi[2] \* n.wh.undiag)) +

 (n.wh.undiag - r.wh.asymp) \* log((n.wh.undiag - r.wh.asymp) /

 (n.wh.undiag - (n.wh.undiag \* psi[2]))))

dev[7] <- 2 \* (r.POPI \* log(r.POPI / (p.POPI \* n.POPI)) +

 (n.POPI - r.POPI) \* log((n.POPI - r.POPI) /

 (n.POPI - (n.POPI \* p.POPI))))

sumdev <- sum(dev[ ])

}

# Data

list(

# Routine data

r.routine = c(8295,13241,18851,11914),

range.max = c(1233,3101,9756,9609),

# Census data - 2002

N=c(NA,NA,NA,NA,NA, NA,NA,NA,NA,NA, NA,NA,NA,NA,NA,

305500,306300,296400,291400,294800, 310100,313900,305600,294700,295000,

304100,317000,329600,349600,370300, 380900,376900,387800,390900,399400,

401200,402600,398700,391900,381900, 370900,356200,349000,343800),

# Wolner-Hansenn

r.wh.undiagpop = 25, n.wh.all = 36

r.wh.asymp = 4, n.wh.undiag = 25

# POPI

r.POPI = 23,n.POPI = 1186

)

# Initial values 1

list(

# population PID incidence

lambda.PID.diag = c(0.01,0.01,0.01,0.01), range.temp = c(1,1,1,1),

# Proportion of PID cases diagnosed

psi = c(0.4,0.5)

)

# Initial values 2

list(

# population PID incidence

lambda.PID.diag = c(0.1,0.1,0.1,0.1), range.temp = c(1000,1000,1000,1000),

# Proportion of PID cases diagnosed

psi = c(0.1,0.1)

)

# Appendix 11

#### WinBUGS code for cumulative PID exposure

This Appendix provides the programming code used to estimate the distribution of salpingitis and number of subsequent PID episodes in the population. The appendix is set out in three sections. (A) provides the WBDev functions written in Component Pascal which are called by the WinBUGS code shown in sections B and C. (B) is the WinBUGS code that calculates the numbers of PIDs, CT-related and non-CT-related. (C) is the WinBUGS code that develops the comparisons to the Lund data.

**(A). WBDEV code**

Note that in appendix A the philap parameter can be removed to generate estimates of the distribution of clinical PID and diagnosed clinical PID. To generate estimates for the distribution of salpingitis it can be included as a multiplier in the exponents for all of the transition probabilities.

MODULE WBDevCumulativePIDlap;

 IMPORT

 WBDevVector,

 Math;

 TYPE

 Function = POINTER TO RECORD (WBDevVector.Node) END;

 Factory = POINTER TO RECORD (WBDevVector.Factory) END;

 VAR

 fact-: WBDevVector.Factory;

 PROCEDURE (func: Function) DeclareArgTypes (OUT args: ARRAY OF CHAR);

 BEGIN

 args := "vv";

 END DeclareArgTypes;

 PROCEDURE readindata1 (func: Function; OUT lambda\_PID: ARRAY OF REAL);

 VAR

 index,a: INTEGER;

 BEGIN

 index := 0;

 FOR a := 1 TO 4 DO

 lambda\_PID[a] := func.arguments[0][index].Value();

 INC(index);

 END;

 END readindata1;

PROCEDURE readindata2 (func: Function; OUT psi: REAL);

 BEGIN

 psi := func.arguments[0][4].Value();

 END readindata2;

 PROCEDURE readindata3 (func: Function; OUT eta1: REAL);

 BEGIN

 eta1 := func.arguments[0][5].Value();

 END readindata3;

 PROCEDURE readindata4 (func: Function; OUT EF: ARRAY OF REAL);

 VAR

 index,a: INTEGER;

 BEGIN

 index := 6;

 FOR a := 1 TO 4 DO

 EF[a] := func.arguments[0][index].Value();

 INC(index);

 END;

 END readindata4;

 PROCEDURE readindata5 (func: Function; OUT philap: REAL);

 BEGIN

 philap := func.arguments[0][10].Value();

 END readindata5;

 PROCEDURE readindata6 (func: Function; OUT N: ARRAY OF REAL);

 VAR

 index,a: INTEGER;

 BEGIN

 index := 0;

 FOR a := 16 TO 44 DO

 N[a] := func.arguments[1][index].Value();

 INC(index);

 END;

 END readindata6;

PROCEDURE tranratesall (lambda\_PID: ARRAY OF REAL; eta1: REAL;

 OUT lambda\_PID2: ARRAY OF ARRAY OF REAL);

 VAR

 a : INTEGER;

 BEGIN

 FOR a := 16 TO 19 DO

 lambda\_PID2[1,a] := lambda\_PID[1] \* 0.85;

 lambda\_PID2[2,a] := lambda\_PID2[1,a] \* eta1;

 END;

 FOR a := 20 TO 24 DO

 lambda\_PID2[1,a] := lambda\_PID[2] \* 0.85;

 lambda\_PID2[2,a] := lambda\_PID2[1,a] \* eta1;

 END;

 FOR a := 25 TO 34 DO

 lambda\_PID2[1,a] := lambda\_PID[3] \* 0.85;

 lambda\_PID2[2,a] := lambda\_PID2[1,a] \* eta1;

 END;

 FOR a := 35 TO 44 DO

 lambda\_PID2[1,a] := lambda\_PID[4] \* 0.85;

 lambda\_PID2[2,a] := lambda\_PID2[1,a] \* eta1;

 END;

 END tranratesall;

PROCEDURE tranratesnct (lambda\_PID: ARRAY OF REAL; eta1: REAL; EF: ARRAY OF REAL;

 OUT lambda\_PID2nct: ARRAY OF ARRAY OF REAL);

 VAR

 a : INTEGER;

 BEGIN

 FOR a := 16 TO 19 DO

 lambda\_PID2nct[1,a] := (lambda\_PID[1] - lambda\_PID[1] \* EF[1]) \* 0.85;

 lambda\_PID2nct[2,a] := lambda\_PID2nct[1,a] \* eta1;

 END;

 FOR a := 20 TO 24 DO

 lambda\_PID2nct[1,a] := (lambda\_PID[2] - lambda\_PID[2] \* EF[2]) \* 0.85;

 lambda\_PID2nct[2,a] := lambda\_PID2nct[1,a] \* eta1;

 END;

 FOR a := 25 TO 34 DO

 lambda\_PID2nct[1,a] := (lambda\_PID[3] - lambda\_PID[3] \* EF[3]) \* 0.85;

 lambda\_PID2nct[2,a] := lambda\_PID2nct[1,a] \* eta1;

 END;

 FOR a := 35 TO 44 DO

 lambda\_PID2nct[1,a] := (lambda\_PID[4] - lambda\_PID[4] \* EF[4]) \* 0.85;

 lambda\_PID2nct[2,a] := lambda\_PID2nct[1,a] \* eta1;

 END;

 END tranratesnct;

 PROCEDURE tranproball (lambda\_PID2: ARRAY OF ARRAY OF REAL; **philap: REAL**;

 OUT p: ARRAY OF ARRAY OF ARRAY OF REAL);

 VAR

 a: INTEGER;

 BEGIN

 FOR a := 14 TO 15 DO

 p[1,2,a] := 0;

 p[1,1,a] := 0;

 p[2,2,a] := 0;

 p[2,5,a] := 0;

 p[2,3,a] := 0;

 p[3,3,a] := 0;

 p[3,5,a] := 0;

 p[3,4,a] := 0;

 p[4,5,a] := 0;

 p[4,4,a] := 0;

 p[5,5,a] := 0;

 p[5,8,a] := 0;

 p[5,6,a] := 0;

 p[6,6,a] := 0;

 p[6,8,a] := 0;

 p[6,7,a] := 0;

 p[7,8,a] := 0;

 p[7,7,a] := 0;

 p[8,8,a] := 0;

 END;

 FOR a := 16 TO 44 DO

 p[1,2,a] := 1 - Math.Exp(-lambda\_PID2[1,a] **\* philap**);

 p[1,1,a] := 1 - p[1,2,a];

 p[2,2,a] := 0;

 p[2,5,a] := 1 - Math.Exp(-lambda\_PID2[2,a]);

 p[2,3,a] := 1 - p[2,5,a];

 p[3,3,a] := 0;

 p[3,5,a] := 1 - Math.Exp(-lambda\_PID2[2,a]);

 p[3,4,a] := 1 - p[3,5,a];

 p[4,5,a] := 1 - Math.Exp(-lambda\_PID2[1,a]);

 p[4,4,a] := 1 - p[4,5,a];

 p[5,5,a] := 0;

 p[5,8,a] := 1 - Math.Exp(-lambda\_PID2[2,a]);

 p[5,6,a] := 1 - p[5,8,a];

 p[6,6,a] := 0;

 p[6,8,a] := 1 - Math.Exp(-lambda\_PID2[2,a]);

 p[6,7,a] := 1 - p[6,8,a];

 p[7,8,a] := 1 - Math.Exp(-lambda\_PID2[1,a]);

 p[7,7,a] := 1 - p[7,8,a];

 p[8,8,a] := 1;

 END;

 END tranproball;

 PROCEDURE tranprobnct (lambda\_PID2nct: ARRAY OF ARRAY OF REAL; **philap: REAL;**

 OUT pnct: ARRAY OF ARRAY OF ARRAY OF REAL);

 VAR

 a : INTEGER;

 BEGIN

 FOR a := 16 TO 44 DO

 pnct[1,2,a] := 1 - Math.Exp(-lambda\_PID2nct[1,a] **\* philap**);

 pnct[1,1,a] := 1 - pnct[1,2,a];

 pnct[2,2,a] := 0;

 pnct[2,5,a] := 1 - Math.Exp(-lambda\_PID2nct[2,a]);

 pnct[2,3,a] := 1 - pnct[2,5,a];

 pnct[3,3,a] := 0;

 pnct[3,5,a] := 1 - Math.Exp(-lambda\_PID2nct[2,a]);

 pnct[3,4,a] := 1 - pnct[3,5,a];

 pnct[4,5,a] := 1 - Math.Exp(-lambda\_PID2nct[1,a]);

 pnct[4,4,a] := 1 - pnct[4,5,a];

 pnct[5,5,a] := 0;

 pnct[5,8,a] := 1 - Math.Exp(-lambda\_PID2nct[2,a]);

 pnct[5,6,a] := 1 - pnct[5,8,a];

 pnct[6,6,a] := 0;

 pnct[6,8,a] := 1 - Math.Exp(-lambda\_PID2nct[2,a]);

 pnct[6,7,a] := 1 - pnct[6,8,a];

 pnct[7,8,a] := 1 - Math.Exp(-lambda\_PID2nct[1,a]);

 pnct[7,7,a] := 1 - pnct[7,8,a];

 pnct[8,8,a] :=1;

 END;

 END tranprobnct;

 PROCEDURE stateoccpropall (p: ARRAY OF ARRAY OF ARRAY OF REAL;

 OUT pi: ARRAY OF ARRAY OF REAL);

 VAR

 a,s: INTEGER;

 BEGIN

 FOR a := 13 TO 15 DO

 pi[1,a] := 1;

 FOR s := 2 TO 8 DO

 pi[s,a] := 0;

 END;

 END;

 FOR a := 16 TO 44 DO

 pi[1,a] := pi[1,a-1] \* p[1,1,a];

 pi[2,a] := pi[1,a-1] \* p[1,2,a];

 pi[3,a] := pi[2,a-1] \* p[2,3,a];

 pi[4,a] := pi[3,a-1] \* p[3,4,a] + pi[4,a-1] \* p[4,4,a];

 pi[5,a] := pi[2,a-1] \* p[2,5,a] + pi[3,a-1] \* p[3,5,a] + pi[4,a-1] \* p[4,5,a] + pi[5,a-1] \* p[5,5,a];

 pi[6,a] := pi[5,a-1] \* p[5,6,a];

 pi[7,a] := pi[6,a-1] \* p[6,7,a] + pi[7,a-1] \* p[7,7,a];

 pi[8,a] := pi[5,a-1] \* p[5,8,a] + pi[6,a-1] \* p[6,8,a] + pi[7,a-1] \* p[7,8,a] + pi[8,a-1];

 END;

 END stateoccpropall;

 PROCEDURE stateoccpropnct (pnct: ARRAY OF ARRAY OF ARRAY OF REAL;

 OUT pinct: ARRAY OF ARRAY OF REAL);

 VAR

 a,s: INTEGER;

 BEGIN

 FOR a := 13 TO 15 DO

 pinct[1,a] := 1;

 FOR s := 2 TO 8 DO

 pinct[s,a] := 0;

 END;

 END;

 FOR a := 16 TO 44 DO

 pinct[1,a] := pinct[1,a-1] \* pnct[1,1,a];

 pinct[2,a] := pinct[1,a-1] \* pnct[1,2,a];

 pinct[3,a] := pinct[2,a-1] \* pnct[2,3,a];

 pinct[4,a] := pinct[3,a-1] \* pnct[3,4,a] + pinct[4,a-1] \* pnct[4,4,a];

 pinct[5,a] := pinct[2,a-1] \* pnct[2,5,a] + pinct[3,a-1] \* pnct[3,5,a] + pinct[4,a-1] \* pnct[4,5,a] + pinct[5,a-1] \* pnct[5,5,a];

 pinct[6,a] := pinct[5,a-1] \* pnct[5,6,a];

 pinct[7,a] := pinct[6,a-1] \* pnct[6,7,a] + pinct[7,a-1] \* pnct[7,7,a];

 pinct[8,a] := pinct[5,a-1] \* pnct[5,8,a] + pinct[6,a-1] \* pnct[6,8,a] + pinct[7,a-1] \* pnct[7,8,a] +

 pinct[8,a-1];

 END;

 END stateoccpropnct;

 PROCEDURE numberPIDsageall (pi: ARRAY OF ARRAY OF REAL; N: ARRAY OF REAL;

 OUT PIDnum: ARRAY OF ARRAY OF REAL);

 VAR

 a: INTEGER;

 BEGIN

 FOR a := 16 TO 44 DO

 PIDnum[1,a] := (pi[2,a] + pi[3,a] + pi[4,a]);

 PIDnum[2,a] := (pi[5,a] + pi[6,a] + pi[7,a]);

 PIDnum[3,a] := pi[8,a];

 END;

 END numberPIDsageall;

PROCEDURE numberPIDsagegpall (PIDnum: ARRAY OF ARRAY OF REAL; N: ARRAY OF REAL;

 OUT numPIDs: ARRAY OF ARRAY OF REAL);

 VAR

 n: INTEGER;

 BEGIN

 FOR n := 1 TO 3 DO

 numPIDs[n,1] := (PIDnum[n,16] \* N[16] + PIDnum[n,17] \* N[17] + PIDnum[n,18] \* N[18] +

 PIDnum[n,19] \* N[19]) /

 (N[16] + N[17] + N[18] + N[19]);

 numPIDs[n,2] := (PIDnum[n,20] \* N[20] + PIDnum[n,21] \* N[21] + PIDnum[n,22] \* N[22] +

 PIDnum[n,23] \* N[23] + PIDnum[n,24] \* N[24]) /

 (N[20] + N[21] + N[22] + N[23] + N[24]);

 numPIDs[n,3] := (PIDnum[n,25] \* N[25] + PIDnum[n,26] \* N[26] + PIDnum[n,27] \* N[27] +

 PIDnum[n,28] \* N[28] + PIDnum[n,29] \* N[29] + PIDnum[n,30] \* N[30] +

 PIDnum[n,31] \* N[31] + PIDnum[n,32] \* N[32] + PIDnum[n,33] \* N[33]+

 PIDnum[n,34] \* N[34]) /

 (N[25] + N[26] + N[27] + N[28] + N[29] + N[30] + N[31] + N[32] + N[33] + N[34]);

numPIDs[n,4] := (PIDnum[n,35] \* N[35] + PIDnum[n,36] \* N[36] + PIDnum[n,37] \* N[37] +

 PIDnum[n,38] \* N[38] + PIDnum[n,39] \* N[39] + PIDnum[n,40] \* N[40] +

 PIDnum[n,41] \* N[41] + PIDnum[n,42] \* N[42] + PIDnum[n,43] \* N[43] +

 PIDnum[n,44] \* N[44]) /

 (N[35] + N[36] + N[37] + N[38] + N[39] + N[40] + N[41] + N[42] + N[43] + N[44]);

 END;

 END numberPIDsagegpall;

 PROCEDURE numberPIDsobsageall (PIDnum: ARRAY OF ARRAY OF REAL; psi: REAL;

 OUT PIDnumobs: ARRAY OF ARRAY OF REAL);

 VAR

 a: INTEGER;

 BEGIN

 FOR a := 16 TO 44 DO

 PIDnumobs[1,a] := PIDnum[1,a] \* psi + 2 \* PIDnum[2,a] \* psi \* (1 - psi) +

 3 \* PIDnum[3,a] \* psi \* (1 - psi) \* (1 - psi);

 PIDnumobs[2,a] := PIDnum[2,a] \* psi \* psi + 3 \* PIDnum[3,a] \* psi \* psi \* (1 - psi);

 PIDnumobs[3,a] := PIDnum[3,a] \* psi \* psi \* psi

 END;

 END numberPIDsobsageall;

PROCEDURE numberPIDsagenct (pinct: ARRAY OF ARRAY OF REAL; N: ARRAY OF REAL;

 OUT PIDnumnct: ARRAY OF ARRAY OF REAL);

 VAR

 a: INTEGER;

 BEGIN

 FOR a := 16 TO 44 DO

 PIDnumnct[1,a] := (pinct[2,a] + pinct[3,a] + pinct[4,a]);

 PIDnumnct[2,a] := (pinct[5,a] + pinct[6,a] + pinct[7,a]);

 PIDnumnct[3,a] := pinct[8,a];

 END;

 END numberPIDsagenct;

 PROCEDURE numberPIDsagegpnct (PIDnumnct: ARRAY OF ARRAY OF REAL; N: ARRAY OF

 REAL;

 OUT numPIDsnct: ARRAY OF ARRAY OF REAL);

 VAR

 n: INTEGER;

 BEGIN

 FOR n := 1 TO 3 DO

 numPIDsnct[n,1] := (PIDnumnct[n,16] \* N[16] + PIDnumnct[n,17] \* N[17] +

 PIDnumnct[n,18] \* N[18] + PIDnumnct[n,19] \* N[19]) /

 (N[16] + N[17] + N[18] + N[19]);

 numPIDsnct[n,2] := (PIDnumnct[n,20] \* N[20] + PIDnumnct[n,21] \* N[21] +

 PIDnumnct[n,22] \* N[22] + PIDnumnct[n,23] \* N[23] +

 PIDnumnct[n,24] \* N[24]) /

 (N[20] + N[21] + N[22] + N[23] + N[24]);

 numPIDsnct[n,3] := (PIDnumnct[n,25] \* N[25] + PIDnumnct[n,26] \* N[26] +

 PIDnumnct[n,27] \* N[27] + PIDnumnct[n,28] \* N[28] +

 PIDnumnct[n,29] \* N[29] + PIDnumnct[n,30] \* N[30] +

 PIDnumnct[n,31] \* N[31] + PIDnumnct[n,32] \* N[32] +

 PIDnumnct[n,33] \* N[33] + PIDnumnct[n,34] \* N[34]) /

 (N[25] + N[26] + N[27] + N[28] + N[29] + N[30] + N[31] + N[32] + N[33] + N[34]);

 numPIDsnct[n,4] := (PIDnumnct[n,35] \* N[35] + PIDnumnct[n,36] \* N[36] +

 PIDnumnct[n,37] \* N[37] + PIDnumnct[n,38] \* N[38] +

 PIDnumnct[n,39] \* N[39] + PIDnumnct[n,40] \* N[40] +

 PIDnumnct[n,41] \* N[41] + PIDnumnct[n,42] \* N[32] +

 PIDnumnct[n,43] \* N[43] + PIDnumnct[n,44] \* N[44]) /

 (N[35] + N[36] + N[37] + N[38] + N[39] + N[40] + N[41] + N[42] + N[43] + N[44]);

 END;

 END numberPIDsagegpnct;

 PROCEDURE numberPIDsobsageallnct (PIDnumnct: ARRAY OF ARRAY OF REAL; psi: REAL;

 OUT PIDnumobsnct: ARRAY OF ARRAY OF REAL);

 VAR

 a: INTEGER;

 BEGIN

 FOR a := 16 TO 44 DO

 PIDnumobsnct[1,a] := PIDnumnct[1,a] \* psi + 2 \* PIDnumnct[2,a] \* psi \* (1 - psi) +

 3 \* PIDnumnct[3,a] \* psi \* (1 - psi) \* (1 - psi);

 PIDnumobsnct[2,a] := PIDnumnct[2,a] \* psi \* psi + 3 \* PIDnumnct[3,a] \* psi \* psi \* (1 - psi);

 PIDnumobsnct[3,a] := PIDnumnct[3,a] \* psi \* psi \* psi

 END;

 END numberPIDsobsageallnct;

PROCEDURE eventsin8 (pi: ARRAY OF ARRAY OF REAL; p: ARRAY OF ARRAY OF ARRAY OF

 REAL; lambda\_PID2: ARRAY OF ARRAY OF REAL; N: ARRAY OF REAL;

 OUT PIDsin8: ARRAY OF REAL);

 VAR

 a: INTEGER;

 last2yrs: ARRAY 45 OF REAL;

 BEGIN

 PIDsin8[14] := 0;

 PIDsin8[15] := 0;

 FOR a := 16 TO 44 DO

 last2yrs[a] := pi[5,a-3] \* p[5,8,a-2] + pi[5,a-2] \* p[5,8,a-1] +

 pi[6,a-3] \* p[6,8,a-2] + pi[6,a-2] \* p[6,8,a-1] +

 pi[7,a-3] \* p[7,8,a-2] + pi[7,a-2] \* p[7,8,a-1];

 PIDsin8[a] := last2yrs[a] \* ( 1 - Math.Exp(-lambda\_PID2[2,a])) +

 (pi[8,a] - last2yrs[a]) \* (1 - Math.Exp(-lambda\_PID2[1,a]))

 END;

 END eventsin8;

PROCEDURE eventsin8nct (pinct: ARRAY OF ARRAY OF REAL; pnct: ARRAY OF ARRAY OF

 ARRAY OF REAL; lambda\_PID2nct: ARRAY OF ARRAY OF REAL; N:

 ARRAY OF REAL;

 OUT PIDsin8nct: ARRAY OF REAL);

 VAR

 a: INTEGER;

 last2yrs: ARRAY 45 OF REAL;

 BEGIN

 PIDsin8nct[14] := 0;

 PIDsin8nct[15] := 0;

 FOR a := 16 TO 44 DO

 last2yrs[a] := pinct[5,a-3] \* pnct[5,8,a-2] + pinct[5,a-2] \* pnct[5,8,a-1] +

 pinct[6,a-3] \* pnct[6,8,a-2] + pinct[6,a-2] \* pnct[6,8,a-1] +

 pinct[7,a-3] \* pnct[7,8,a-2] + pinct[7,a-2] \* pnct[7,8,a-1];

 PIDsin8nct[a] := last2yrs[a] \* ( 1 - Math.Exp(-lambda\_PID2nct[2,a])) +

 (pinct[8,a] - last2yrs[a]) \* (1 - Math.Exp(-lambda\_PID2nct[1,a]));

 END;

 END eventsin8nct;

PROCEDURE predincforpop (pi: ARRAY OF ARRAY OF REAL; p: ARRAY OF ARRAY OF ARRAY

 OF REAL; PIDsin8: ARRAY OF REAL;

 OUT PIDinc: ARRAY OF ARRAY OF REAL);

 VAR

 a: INTEGER;

 BEGIN

 FOR a := 16 TO 44 DO

 PIDinc[1,a] := pi[1,a-1] \* p[1,2,a];

 PIDinc[2,a] := pi[2,a-1] \* p[2,5,a] + pi[3,a-1] \* p[3,5,a] + pi[4,a-1] \* p[4,5,a];

 PIDinc[3,a] := pi[5,a-1] \* p[5,8,a] + pi[6,a-1] \* p[6,8,a] + pi[7,a-1] \* p[7,8,a];

 PIDinc[4,a] := PIDsin8[a];

 END;

 END predincforpop;

PROCEDURE predincforpopnct (pinct: ARRAY OF ARRAY OF REAL; pnct: ARRAY OF ARRAY OF

 ARRAY OF REAL; PIDsin8nct: ARRAY OF REAL;

 OUT PIDincnct: ARRAY OF ARRAY OF REAL);

 VAR

 a: INTEGER;

 BEGIN

 FOR a := 16 TO 44 DO

 PIDincnct[1,a] := pinct[1,a-1] \* pnct[1,2,a];

 PIDincnct[2,a] := pinct[2,a-1] \* pnct[2,5,a] + pinct[3,a-1] \* pnct[3,5,a] +

 pinct[4,a-1] \* pnct[4,5,a];

 PIDincnct[3,a] := pinct[5,a-1] \* pnct[5,8,a] + pinct[6,a-1] \* pnct[6,8,a] +

 pinct[7,a-1] \* pnct[7,8,a];

 PIDincnct[4,a] := PIDsin8nct[a];

 END;

 END predincforpopnct;

PROCEDURE predincgpforpop (PIDinc: ARRAY OF ARRAY OF REAL; N: ARRAY OF REAL;

 PIDsin8: ARRAY OF REAL;

 OUT pred\_prop\_ag: ARRAY OF REAL);

 VAR

 a: INTEGER;

 pred\_pop: ARRAY 45 OF REAL;

 sumN: REAL;

BEGIN

 FOR a := 16 TO 44 DO

 pred\_pop[a] := (PIDinc[1,a] + PIDinc[2,a] + PIDinc[3,a] + PIDinc[4,a]) \* N[a];

 END;

 pred\_prop\_ag[1] := (pred\_pop[16] + pred\_pop[17] + pred\_pop[18] + pred\_pop[19]) /

 (N[16] + N[17] + N[18] + N[19]);

 pred\_prop\_ag[2] := (pred\_pop[20] + pred\_pop[21] + pred\_pop[22] + pred\_pop[23] + pred\_pop[24]) /

 (N[20] + N[21] + N[22] + N[23] + N[24]);

 pred\_prop\_ag[3] := (pred\_pop[25] + pred\_pop[26] + pred\_pop[27] + pred\_pop[28] + pred\_pop[29] +

 pred\_pop[30] + pred\_pop[31] + pred\_pop[32] + pred\_pop[33] + pred\_pop[34]) /

 (N[25] + N[26] + N[27] + N[28] + N[29] + N[30] + N[31] + N[32] + N[33] + N[34]);

 pred\_prop\_ag[4] := (pred\_pop[35] + pred\_pop[36] + pred\_pop[37] + pred\_pop[38] + pred\_pop[39] +

 pred\_pop[40] + pred\_pop[41] + pred\_pop[42] + pred\_pop[43] + pred\_pop[44]) /

 (N[35] + N[36] + N[37] + N[38] + N[39] + N[40] + N[41] + N[42] + N[43] + N[44]);

 pred\_prop\_ag[0] := 0;

 sumN := 0;

 FOR a := 16 TO 44 DO

 pred\_prop\_ag[0] := pred\_prop\_ag[0] + pred\_pop[a];

 sumN := sumN + N[a];

 END;

 pred\_prop\_ag[0] := pred\_prop\_ag[0] / sumN;

 END predincgpforpop;

PROCEDURE predincgpforpopnct (PIDincnct: ARRAY OF ARRAY OF REAL; N: ARRAY OF REAL;

 PIDsin8nct: ARRAY OF REAL;

 OUT pred\_prop\_agnct: ARRAY OF REAL);

 VAR

 a: INTEGER;

 pred\_pop: ARRAY 45 OF REAL;

 sumN: REAL;

 BEGIN

 FOR a := 16 TO 44 DO

 pred\_pop[a] := (PIDincnct[1,a] + PIDincnct[2,a] + PIDincnct[3,a] + PIDincnct[4,a]) \* N[a];

 END;

 pred\_prop\_agnct[1] := (pred\_pop[16] + pred\_pop[17] + pred\_pop[18] + pred\_pop[19]) /

 (N[16] + N[17] + N[18] + N[19]);

 pred\_prop\_agnct[2] := (pred\_pop[20] + pred\_pop[21] + pred\_pop[22] + pred\_pop[23] +

 pred\_pop[24]) / (N[20] + N[21] + N[22] + N[23] + N[24]);

 pred\_prop\_agnct[3] := (pred\_pop[25] + pred\_pop[26] + pred\_pop[27] + pred\_pop[28] +

 pred\_pop[29] + pred\_pop[30] + pred\_pop[31] + pred\_pop[32] +

 pred\_pop[33] + pred\_pop[34]) /

 (N[25] + N[26] + N[27] + N[28] + N[29] + N[30] + N[31] + N[32] + N[33] + N[34]);

 pred\_prop\_agnct[4] := (pred\_pop[35] + pred\_pop[36] + pred\_pop[37] + pred\_pop[38] +

 pred\_pop[39] + pred\_pop[40] + pred\_pop[41] + pred\_pop[42] +

 pred\_pop[43] + pred\_pop[44]) /

 (N[35] + N[36] + N[37] + N[38] + N[39] + N[40] + N[41] + N[42] + N[43] + N[44]);

 pred\_prop\_agnct[0] := 0;

 sumN := 0;

 FOR a := 16 TO 44 DO

 pred\_prop\_agnct[0] := pred\_prop\_agnct[0] + pred\_pop[a];

 sumN := sumN + N[a];

 END;

 pred\_prop\_agnct[0] := pred\_prop\_agnct[0] / sumN;

 END predincgpforpopnct;

 PROCEDURE (func: Function) Evaluate (OUT values: ARRAY OF REAL);

 VAR

 a, n, i: INTEGER;

 eta1, psi, philap: REAL;

 lambda\_PID, EF, pred\_prop\_ag, pred\_prop\_agnct: ARRAY 5 OF REAL;

 PIDsin8,PIDsin8nct,N: ARRAY 45 OF REAL;

 lambda\_PID2, lambda\_PID2nct: ARRAY 3,45 OF REAL;

 p, pnct: ARRAY 9,9,45 OF REAL;

 pi, pinct: ARRAY 9,45 OF REAL;

 numPIDs, numPIDsnct: ARRAY 4,5 OF REAL;

 PIDinc, PIDincnct: ARRAY 5,45 OF REAL;

 PIDnum, PIDnumnct, PIDnumobs,PIDnumobsnct: ARRAY 4,45 OF REAL;

 BEGIN

 readindata1(func, lambda\_PID);

 readindata2(func, psi);

 readindata3(func, eta1);

 readindata4(func, EF);

 readindata5(func, philap);

 readindata6(func, N);

 tranratesall(lambda\_PID, eta1, lambda\_PID2);

 tranratesnct(lambda\_PID, eta1, EF, lambda\_PID2nct);

 tranproball(lambda\_PID2, **philap,** p);

 tranprobnct(lambda\_PID2nct, **philap,** pnct);

 stateoccpropall(p, pi );

 stateoccpropnct(pnct, pinct );

 numberPIDsageall(pi, N, PIDnum);

 numberPIDsagenct(pinct, N, PIDnumnct);

 numberPIDsagegpall(PIDnum, N, numPIDs);

 numberPIDsagegpnct(PIDnumnct, N, numPIDsnct);

 numberPIDsobsageall(PIDnum, psi, PIDnumobs);

 numberPIDsobsageallnct(PIDnumnct, psi, PIDnumobsnct);

 eventsin8(pi, p, lambda\_PID2, N, PIDsin8);

 predincforpop(pi, p, PIDsin8, PIDinc);

 predincgpforpop(PIDinc, N, PIDsin8, pred\_prop\_ag);

 eventsin8nct(pinct, pnct, lambda\_PID2nct, N, PIDsin8nct);

 predincforpopnct(pinct, pnct, PIDsin8nct, PIDincnct);

 predincgpforpopnct(PIDincnct, N, PIDsin8nct, pred\_prop\_agnct);

 i := 0;

 FOR n := 1TO 3 DO

 FOR a := 1 TO 4 DO

 values[i] := numPIDs[n,a];

 values[i+12] := numPIDsnct[n,a];

 INC(i);

 END;

 END;

 FOR i := 24 TO 28 DO

 values[i] := pred\_prop\_ag[i-24];

 END;

 i := 29;

 FOR n := 1 TO 3 DO

 FOR a := 16 TO 44 DO

 values[i] := PIDnum[n,a];

 values[i+87] := PIDnumnct[n,a];

 INC(i);

 END;

 END;

 i := 203;

 FOR n := 1 TO 4 DO

 FOR a := 16 TO 44 DO

 values[i] := PIDinc[n,a];

 values[i+116] := PIDincnct[n,a];

 INC(i);

 END;

 END;

 i := 435;

 FOR n := 1 TO 3 DO

 FOR a := 16 TO 44 DO

 values[i] := PIDnumobs[n,a];

 values[i+87] := PIDnumobsnct[n,a];

 INC(i);

 END;

 END;

 END Evaluate;

 PROCEDURE (f: Factory) New (option: INTEGER): Function;

 VAR

 func: Function;

 BEGIN

 NEW(func); func.Initialize; RETURN func;

 END New;

 PROCEDURE Install\*;

 BEGIN

 WBDevVector.Install(fact);

 END Install;

 PROCEDURE Init;

 VAR

 f: Factory;

 BEGIN

 NEW(f); fact := f;

 END Init;

 BEGIN

 Init;

 END WBDevCumulativePIDlap.

**(B). WinBUGS code to calculate numbers of PIDs and non-CT related PIDs**

**model {**

**# PID incidence and ct GP re-infection rate ratio informative priors**

Y[1:10] ~ dmnorm(mu[], Omega[ , ])

for (ag in 1:4) {

 log(lambda.PID[ag]) <- Y[ag]

 EF[ag] <- Y[ag+5] \* Y[ag+5]

 }

logit(psi) <- Y[5]

log(eta1) <- Y[10]

**# WBDEV call**

for (ag in 1:4) {

 input1[ag] <- lambda.PID[ag]

 input1[ag+6] <- EF[ag]

 }

input1[5] <- psi

input1[6] <- eta1

for (i in 1:29) {

 input2[i] <- N[i+15]

 }

philap ~ dbeta(12,16)

input1[11] <- philap

solution[1:609] <- cumulativePIDlap(input1[1:11],input2[1:29])

for (ag in 1:4) {

 Expect.prop[ag] <- 1 - exp(-lambda.PID[ag])

 }

total.expect.prop <- (Expect.prop[1] \* sum(N[16:19]) +

 Expect.prop[2] \* sum(N[20:24]) +

 Expect.prop[3] \* sum(N[25:34]) +

 Expect.prop[4] \* sum(N[35:44])) /

 sum(N[16:44])

**}**

**# Data**

**list(**

**# PID incidence, r-infection rate, Etological fractions and re-infection rate**

**mu = c(-3.865, -3.595, -3.964, -4.402, -0.5856, 0.7104, 0.4815, 0.3117, 0.3277, 1.919),**

Omega = structure(.Data =c(

**573.855, -49.264, -10.817, -3.992, 323.371, 97.687, -112.351, -30.198, -5.401, -2.713,**

**-49.264, 301.419, -14.194, -3.299, 147.591, -78.069, 176.480, -72.284, -11.551, 2.012,**

**-10.817, -14.194, 105.908, -20.063, 37.684, -13.275, -46.636, 236.305, -118.245, 0.515,**

**-3.992, -3.299, -20.063, 57.687, 19.172, -2.664, -8.418, -125.677, 138.728, 0.088,**

**323.371, 147.591, 37.684, 19.172, 351.620, 1.363, -1.443, -1.164, 1.537, 0.030,**

**97.687, -78.069, -13.275, -2.664, 1.363, 285.706, -324.389, -85.846, -16.864, -7.957,**

**-112.351, 176.480, -46.636, -8.418, -1.443, -324.389, 734.182, -295.254, -50.046, 8.090,**

**-30.198, -72.284, 236.305, -125.677, -1.164, -85.846, -295.254, 1512.713, -758.618, 3.088,**

**-5.401, -11.551, -118.245, 138.728, 1.537, -16.864, -50.046, -758.618, 839.634, 0.705,**

**-2.713, 2.012, 0.515, 0.088, 0.030, -7.957, 8.090, 3.088, 0.705, 13.711**

),

.Dim = c(10,10)),

**# Population sizes from census, age =1...44 - 2002**

N=c(NA,NA,NA,NA,NA, NA,NA,NA,NA,NA, NA,NA,NA,NA,NA,

305500,306300,296400,291400,294800, 310100,313900,305600,294700,295000,

304100,317000,329600,349600,370300, 380900,376900,387800,390900,399400,

401200,402600,398700,391900,381900, 370900,356200,349000,343800)

**)**

**# Initial values 1**

**list(**

Y = c(-5,-5,-5,-5,-1,-5,-5,-5,-5,-1)

**)**

**# Initial values 2**

**list(**

Y = c(-1,-1,-1,-1,3,-1,-1,-1,-1,3)

**)**

**(C). WinBUGS code comparisons to Lund data**

**# Westrom proportions**

model {

r1[1:3] ~ dmulti(p1[1:3],N1)

r2[1:3] ~ dmulti(p2[1:3],N2)

p1[1:3] ~ ddirch(a1[1:3])

p2[1:3] ~ ddirch(a2[1:3])

}

# Data

list(r1 = c(771,158,61),r2 = c(220,27,4), N1 = 990, N2 = 251,

 a1 =c(1,1,1), a2 = c(1,1,1))

# Initial Values

list(p1 = c(0.7,0.2,0.1),p2 = c(0.7,0.2,0.1))

# twelve - comparison of cumulative PID to Westrom

model {

 Y[1:10] ~ dmnorm(mu[], Omega[ , ])

 for (ag in 1:4) {

 log(lambda.PID[ag]) <- Y[ag]

 }

 Logit(psi) <- Y[5]

 log(eta1) <- Y[10]

# transition rates

 for (a in 16:19) {

 lambda.PID2[1,a] <- lambda.PID[1] \* 0.85

 lambda.PID2[2,a] <- lambda.PID2[1,a] \* eta1

 }

 for (a in 20:24) {

 lambda.PID2[1,a] <- lambda.PID[2] \* 0.85

 lambda.PID2[2,a] <- lambda.PID2[1,a] \* eta1

 }

 for (a in 25:34) {

 lambda.PID2[1,a] <- lambda.PID[3] \* 0.85

 lambda.PID2[2,a] <- lambda.PID2[1,a] \* eta1

 }

 for (a in 35:44) {

 lambda.PID2[1,a] <- lambda.PID[4] \* 0.85

 lambda.PID2[2,a] <- lambda.PID2[1,a] \* eta1

 }

 # transition probabilities

 # state 1 = 1 PID <1 year

 # state 2 = 1 PID 1-2 years

 # state 3 = 1 PID 2+ years

 # state 4 = 2 PIDs <1 year

 # state 5 = 2 PIDs 1-2 years

 # state 6 = 2 PIDs 2+ years

 # state 7 = 3 PIDs < 1 year

 # state 8 = 3 PIDs 1-2 years

 # state 9 = 3 PIDs 2+ years

 # state 10 = 4 PIDs < 1 year

 # state 11 = 4 PIDs < 1-2 years

 # state 12 = 4 PIDs < 2+ years

 # state 13 = 5 PIDs assume no more than 5 PIDs

 for (a in 16:44) {

 p[1,1,a] <- 0

 p[1,2,a] <- 1 - p[1,4,a]

 p[1,4,a] <- 1 - exp(-lambda.PID2[2,a])

 p[2,2,a] <- 0

 p[2,3,a] <- 1 - p[2,4,a]

 p[2,4,a] <- 1 - exp(-lambda.PID2[2,a])

 p[3,3,a] <- 1 - p[3,4,a]

 p[3,4,a] <- 1 - exp(-lambda.PID2[1,a])

 p[4,4,a] <- 0

 p[4,5,a] <- 1 - p[4,7,a]

 p[4,7,a] <- 1 - exp(-lambda.PID2[2,a])

 p[5,5,a] <- 0

 p[5,6,a] <- 1 - p[5,7,a]

 p[5,7,a] <- 1 - exp(-lambda.PID2[2,a])

 p[6,6,a] <- 1 - p[6,7,a]

 p[6,7,a] <- 1 - exp(-lambda.PID2[1,a])

 p[7,7,a] <- 0

 p[7,8,a] <- 1 - p[7,10,a]

 p[7,10,a] <- 1 - exp(-lambda.PID2[2,a])

 p[8,8,a] <- 0

 p[8,9,a] <- 1 - p[8,10,a]

 p[8,10,a] <- 1 - exp(-lambda.PID2[2,a])

 p[9,9,a] <- 1 - p[9,10,a]

 p[9,10,a] <- 1 - exp(-lambda.PID2[1,a])

 p[10,10,a] <- 0

 p[10,11,a] <- 1 - p[10,13,a]

 p[10,13,a] <- 1 - exp(-lambda.PID2[2,a])

 p[11,11,a] <- 0

 p[11,12,a] <- 1 - p[11,13,a]

 p[11,13,a] <- 1 - exp(-lambda.PID2[2,a])

 p[12,12,a] <- 1 - p[9,10,a]

 p[12,13,a] <- 1 - exp(-lambda.PID2[1,a])

 p[13,13,a] <- 1

 }

 # State occupancy proportions

 for (i in 16:37) {

 pi[i,1,i-1] <- 1

 pi[i,2,i-1] <- 0

 pi[i,3,i-1] <- 0

 pi[i,4,i-1] <- 0

 pi[i,5,i-1] <- 0

 pi[i,6,i-1] <- 0

 pi[i,7,i-1] <- 0

 pi[i,8,i-1] <- 0

 pi[i,9,i-1] <- 0

 pi[i,10,i-1] <- 0

 pi[i,11,i-1] <- 0

 pi[i,12,i-1] <- 0

 pi[i,13,i-1] <- 0

 }

 for (i in 16:37) {

 for (a in i:i+7) {

 pi[i,1,a] <- pi[i,1,a-1] \* p[1,1,a]

 pi[i,2,a] <- pi[i,1,a-1] \* p[1,2,a] + pi[i,2,a-1] \* p[2,2,a]

 pi[i,3,a] <- pi[i,2,a-1] \* p[2,3,a] + pi[i,3,a-1] \* p[3,3,a]

 pi[i,4,a] <- pi[i,1,a-1] \* p[1,4,a] + pi[i,2,a-1] \* p[2,4,a] +

 pi[i,3,a-1] \* p[3,4,a] + pi[i,4,a-1] \* p[4,4,a]

 pi[i,5,a] <- pi[i,4,a-1] \* p[4,5,a] + pi[i,5,a-1] \* p[5,5,a]

 pi[i,6,a] <- pi[i,5,a-1] \* p[5,6,a] + pi[i,6,a-1] \* p[6,6,a]

 pi[i,7,a] <- pi[i,4,a-1] \* p[4,7,a] + pi[i,5,a-1] \* p[5,7,a] +

 pi[i,6,a-1] \* p[6,7,a] + pi[i,7,a-1] \* p[7,7,a]

 pi[i,8,a] <- pi[i,7,a-1] \* p[7,8,a] + pi[i,8,a-1] \* p[8,8,a]

 pi[i,9,a] <- pi[i,8,a-1] \* p[8,9,a] + pi[i,9,a-1] \* p[9,9,a]

 pi[i,10,a] <- pi[i,7,a-1] \* p[7,10,a] + pi[i,8,a-1] \* p[8,10,a] +

 pi[i,9,a-1] \* p[9,10,a] + pi[i,10,a-1] \* p[10,10,a]

 pi[i,11,a] <- pi[i,10,a-1] \* p[10,11,a] + pi[i,11,a-1] \* p[11,11,a]

 pi[i,12,a] <- pi[i,11,a-1] \* p[11,12,a] + pi[i,12,a-1] \* p[12,12,a]

 pi[i,13,a] <- pi[i,10,a-1] \* p[10,13,a] + pi[i,11,a-1] \* p[11,13,a] +

 pi[i,12,a-1] \* p[12,13,a] + pi[i,13,a-1] \* p[13,13,a]

 }

 }

 # Proportion in each state by age

 for (i in 16:37) {

 prop[i,1] <- sum(pi[i,1:3,i+7])

 prop[i,2] <- sum(pi[i,4:6,i+7])

 prop[i,3] <- sum(pi[i,7:9,i+7])

 prop[i,4] <- sum(pi[i,10:12,i+7])

 prop[i,5] <- pi[i,13,i+7]

 propcomp[i,1] <- prop[i,1]

 propcomp[i,2] <- prop[i,2]

 propcomp[i,3] <- sum(prop[i,3:5])

 }

 propunder25[1] <- sum(propcomp[16:24,1]) / 9

 propunder25[2] <- sum(propcomp[16:24,2]) / 9

 propunder25[3] <- sum(propcomp[16:24,3]) / 9

 propover25[1] <- sum(propcomp[25:37,1]) / 13

 propover25[2] <- sum(propcomp[25:37,2]) / 13

 propover25[3] <- sum(propcomp[25:37,3]) / 13

 # Proportion expected to be observed in state by age

 for (i in 16:37) {

 prop.obs[i,1] <- prop[i,1] +

 prop[i,2] \* (1 - psi) +

 prop[i,3] \* (1 - psi) \* (1 - psi) +

 prop[i,4] \* (1 - psi) \* (1 - psi) \* (1 - psi) +

 prop[i,5] \* (1 - psi) \* (1 - psi) \* (1 - psi) \* (1 - psi)

 prop.obs[i,2] <- prop[i,2] \* psi +

 2 \* prop[i,3] \* psi \* (1 - psi) +

 3 \* prop[i,4] \* psi \* (1 - psi) \* (1 - psi) +

 4 \* prop[i,5] \* psi \* (1 - psi) \* (1 - psi) \* (1 - psi)

 prop.obs[i,3] <- prop[i,3] \* psi \* psi +

 3 \* prop[i,4] \* psi \* psi \* (1 - psi) +

 6 \* prop[i,5] \* psi \* psi \* (1 - psi) \* (1 - psi)

 prop.obs[i,4] <- prop[i,4] \* psi \* psi \* psi +

 4 \* prop[i,5] \* psi \* psi \* psi \* (1 - psi)

 prop.obs[i,5] <- prop[i,5] \* psi \* psi \* psi \* psi

 prop.obscomp[i,1] <- prop.obs[i,1]

 prop.obscomp[i,2] <- prop.obs[i,2]

 prop.obscomp[i,3] <- sum(prop.obs[i,3:5])

 }

 prop.obsunder25[1] <- sum(prop.obscomp[16:24,1]) / 9

 prop.obsunder25[2] <- sum(prop.obscomp[16:24,2]) / 9

 prop.obsunder25[3] <- sum(prop.obscomp[16:24,3]) / 9

 prop.obsover25[1] <- sum(prop.obscomp[25:37,1]) / 13

 prop.obsover25[2] <- sum(prop.obscomp[25:37,2]) / 13

 prop.obsover25[3] <- sum(prop.obscomp[25:37,3]) / 13

}

**# Data**

**list(**

**# PID incidence, r-infection raqte, Etological fractions and re-infection rate**

**mu = c(-3.865, -3.595, -3.964, -4.402, -0.5856, 0.7104, 0.4815, 0.3117, 0.3277, 1.919),**

Omega = structure(.Data =c(

**573.855, -49.264, -10.817, -3.992, 323.371, 97.687, -112.351, -30.198, -5.401, -2.713,**

**-49.264, 301.419, -14.194, -3.299, 147.591, -78.069, 176.480, -72.284, -11.551, 2.012,**

**-10.817, -14.194, 105.908, -20.063, 37.684, -13.275, -46.636, 236.305, -118.245, 0.515,**

**-3.992, -3.299, -20.063, 57.687, 19.172, -2.664, -8.418, -125.677, 138.728, 0.088,**

**323.371, 147.591, 37.684, 19.172, 351.620, 1.363, -1.443, -1.164, 1.537, 0.030,**

**97.687, -78.069, -13.275, -2.664, 1.363, 285.706, -324.389, -85.846, -16.864, -7.957,**

**-112.351, 176.480, -46.636, -8.418, -1.443, -324.389, 734.182, -295.254, -50.046, 8.090,**

**-30.198, -72.284, 236.305, -125.677, -1.164, -85.846, -295.254, 1512.713, -758.618, 3.088,**

**-5.401, -11.551, -118.245, 138.728, 1.537, -16.864, -50.046, -758.618, 839.634, 0.705,**

**-2.713, 2.012, 0.515, 0.088, 0.030, -7.957, 8.090, 3.088, 0.705, 13.711**

),

.Dim = c(10,10))

**)**

**# Initial values 1**

**list(**

Y = c(-5,-5,-5,-5,-1,-5,-5,-5,-5,-1)

**)**

**# Initial values 2**

**list(**

Y = c(-1,-1,-1,-1,3,-1,-1,-1,-1,3)

**)**

# Appendix 12

#### WinBUGS code for prediction of EP rates in Table 35

**model {**

**# Population (disecrete time) model - 2 year higher rate**

**# PID incidence and ct GP re-infection rate ratio informative priors**

Y[1:10] ~ dmnorm(mu[], Omega[ , ])

for (ag in 1:4) {

 log(lambda.PID[ag]) <- Y[ag]

 EF[ag] <- Y[ag+5] \* Y[ag+5]

 }

logit(psi) <- Y[5]

log(eta1) <- Y[10]

**# WBDEV call**

for (ag in 1:4) {

 input1[ag] <- lambda.PID[ag]

 input1[ag+6] <- EF[ag]

 }

input1[5] <- psi

input1[6] <- eta1

philap ~ dbeta(12,16)

input1[11] <- philap

for (i in 1:29) {

 input2[i] <- N[i+15]

 }

solution[1:609] <- cumulativePIDlap(input1[1:11],input2[1:29])

**# number of previous PIDs [n,a] n = number: 0,1,2,3; a = age-group**

for (a in 1:4) {

 numPIDs[1,a] <- 1 - numPIDs[2,a] - numPIDs[3,a] - numPIDs[4,a]

 numPIDs[2,a] <- solution[a]

 numPIDs[3,a] <- solution[a+4]

 numPIDs[4,a] <- solution[a+8]

 numPIDsnct[1,a] <- 1 - numPIDsnct[2,a] - numPIDsnct[3,a] - numPIDsnct[4,a]

 numPIDsnct[2,a] <- solution[a+12]

 numPIDsnct[3,a] <- solution[a+16]

 numPIDsnct[4,a] <- solution[a+20]

 numPIDsct[1,a] <- numPIDs[1,a] - numPIDsnct[1,a]

 numPIDsct[2,a] <- numPIDs[2,a] - numPIDsnct[2,a]

 numPIDsct[3,a] <- numPIDs[3,a] - numPIDsnct[3,a]

 numPIDsct[4,a] <- numPIDs[4,a] - numPIDsnct[4,a]

 }

**# population denominators - 2002**

N1619 <- sum(N[16:19])

N2024 <- sum(N[20:24])

N2534 <- sum(N[25:34])

N3544 <- sum(N[35:44])

N1644 <- sum(N[16:44])

N1617 <- sum(N[16:17])

N1820 <- sum(N[18:20])

N2124 <- sum(N[21:24])

N2529 <- sum(N[25:29])

N3044 <- sum(N[30:44])

N1819 <- sum(N[18:19])

N3034 <- sum(N[30:34])

**# Distribution of PID severity - 2002**

**# hospital**

r.HESPID[1] ~ dbin(HESPID[1],N1619)

r.HESPID[2] ~ dbin(HESPID[2],N2024)

r.HESPID[3] ~ dbin(HESPID[3],N2534)

r.HESPID[4] ~ dbin(HESPID[4],N3544)

**# priors**

for (a in 1:4) {

 HESPID[a] ~ dbeta(1,1)

 }

**# kc60**

r.kc602008[1] ~ dbin(kc602008[1],N1619)

r.kc602008[2] ~ dbin(kc602008[2],N2024)

r.kc602008[3] ~ dbin(kc602008[3],N2534)

r.kc602008[4] ~ dbin(kc602008[4],N3544)

**# priors**

for (a in 1:4) {

 kc602008[a] ~ dbeta(1,1)

 }

**# data by age not available until 2008**

**# 13421 / 12117 is the ratio of totals for 2002 and GUMCAD data for 2008**

**# assumes GUMCAD data representative of all kc60**

**# assumes age distribution is the same in 2002 as 2008**

for (a in 1:4) {

 kc60[a] <- kc602008[a] \* 13421 / 12117

 }

**# GPRD**

r.GPRDPID[1] ~ dbin(GPRDPID[1],N1619)

r.GPRDPID[2] ~ dbin(GPRDPID[2],N2024)

r.GPRDPID[3] ~ dbin(GPRDPID[3],N2534)

r.GPRDPID[4] ~ dbin(GPRDPID[4],N3544)

**# priors**

for (a in 1:4) {

 GPRDPID[a] ~ dbeta(1,1)

 }

**# Distribution**

for (a in 1:4) {

 pmin[a] <- kc60[a] + max(HESPID[a],GPRDPID[a])

 pmax[a] <- kc60[a] + HESPID[a] + GPRDPID[a]

 X[a] ~ dunif(pmin[a],pmax[a])

 hospdiag[a] <- psi \* HESPID[a] / X[a]

 milddiag[a] <- 1 - hospdiag[a] - undiag[a]

 undiag[a] <- (1 - psi)

 }

**# proportion with PIDs by each age, severity, and number**

**# all PID**

# n = 1

for (s in 1:3) {

 for (a in 1:4) {

 PIDcat[1,s,a] <- 1 - (PIDcat[2,s,a] + PIDcat[3,s,a] + PIDcat[4,s,a])

 }

 }

# n = 2-4

for (n in 2:4) {

 for (a in 1:4) {

 PIDcat[n,1,a] <- numPIDs[n,a] \* undiag[a]

 PIDcat[n,2,a] <- numPIDs[n,a] \* milddiag[a]

 PIDcat[n,3,a] <- numPIDs[n,a] \* hospdiag[a]

 }

 }

for (n in 2:4) {

 for (s in 1:3) {

 PIDcat1624\_2544[n,s,1] <- (PIDcat[n,s,1] \* sum(N[16:19]) +

 PIDcat[n,s,2] \* sum(N[20:24]) +

 PIDcat[n,s,3] \* sum(N[25:29])) /

 sum(N[16:29])

 PIDcat1624\_2544[n,s,2] <- (PIDcat[n,s,3] \* sum(N[30:34]) +

 PIDcat[n,s,4] \* sum(N[35:44])) /

 sum(N[30:44])

 PIDcat1624\_2544[n,s,3] <- (PIDcat[n,s,1] \* sum(N[16:19]) +

 PIDcat[n,s,2] \* sum(N[20:24]) +

 PIDcat[n,s,3] \* sum(N[25:34]) +

 PIDcat[n,s,4] \* sum(N[35:44])) /

 sum(N[16:44])

 }

 }

for (a in 1:2) {

 for (n in 2:4) {

 sumsevPIDcat1624\_2544[n,a] <- sum(PIDcat1624\_2544[n, ,a])

 }

 sum2sevPIDcat1624\_2544[a] <-sum(sumsevPIDcat1624\_2544[2:4,a])

 for (s in 1:3) {

 sumnumPIDcat1624\_2544[s,a] <- sum(PIDcat1624\_2544[2:4,s,a])

 }

 sum2numPIDcat1624\_2544[a] <- sum(sumnumPIDcat1624\_2544[ ,a])

 }

**# non-CT related PID**

# n = 1

for (s in 1:3) {

 for (a in 1:4) {

 PIDcatnct[1,s,a] <- 1 - (PIDcatnct[2,s,a] + PIDcatnct[3,s,a] +

 PIDcatnct[4,s,a])

 }

 }

# n = 2-4

for (n in 2:4) {

 for (a in 1:4) {

 PIDcatnct[n,1,a] <- numPIDsnct[n,a] \* undiag[a]

 PIDcatnct[n,2,a] <- numPIDsnct[n,a] \* milddiag[a]

 PIDcatnct[n,3,a] <- numPIDsnct[n,a] \* hospdiag[a]

 }

 }

for (n in 2:4) {

 for (s in 1:3) {

 PIDcat1624\_2544nct[n,s,1] <- (PIDcatnct[n,s,1] \* sum(N[16:19]) +

 PIDcatnct[n,s,2] \* sum(N[20:24]) +

 PIDcatnct[n,s,3] \* sum(N[25:29])) /

 sum(N[16:29])

 PIDcat1624\_2544nct[n,s,2] <- (PIDcatnct[n,s,3] \* sum(N[30:34]) +

 PIDcatnct[n,s,4] \* sum(N[35:44])) /

 sum(N[30:44])

 PIDcat1624\_2544nct[n,s,3] <- (PIDcatnct[n,s,1] \* sum(N[16:19]) +

 PIDcatnct[n,s,2] \* sum(N[20:24]) +

 PIDcatnct[n,s,3] \* sum(N[25:34]) +

 PIDcatnct[n,s,4] \* sum(N[35:44])) /

 sum(N[16:44])

 }

 }

for (a in 1:3) {

 for (n in 2:4) {

 sumsevPIDcat1624\_2544nct[n,a] <- sum(PIDcat1624\_2544nct[n, ,a])

 }

 sum2sevPIDcat1624\_2544nct[a] <-sum(sumsevPIDcat1624\_2544nct[2:4,a])

 for (s in 1:3) {

 sumnumPIDcat1624\_2544nct[s,a] <- sum(PIDcat1624\_2544nct[2:4,s,a])

 }

 sum2numPIDcat1624\_2544nct[a] <- sum(sumnumPIDcat1624\_2544nct[ ,a])

 }

# CT related PID

# n = 1

for (s in 1:3) {

 for (a in 1:4) {

 PIDcatct[1,s,a] <- 1 - (PIDcatct[2,s,a] + PIDcatct[3,s,a] +

 PIDcatct[4,s,a])

 }

 }

# n = 2-4

for (n in 2:4) {

 for (a in 1:4) {

 PIDcatct[n,1,a] <- numPIDsct[n,a] \* undiag[a]

 PIDcatct[n,2,a] <- numPIDsct[n,a] \* milddiag[a]

 PIDcatct[n,3,a] <- numPIDsct[n,a] \* hospdiag[a]

 }

 }

for (n in 2:4) {

 for (s in 1:3) {

 PIDcat1624\_2544ct[n,s,1] <- (PIDcatct[n,s,1] \* sum(N[16:19]) +

 PIDcatct[n,s,2] \* sum(N[20:24]) +

 PIDcatct[n,s,3] \* sum(N[25:29])) /

 sum(N[16:29])

 PIDcat1624\_2544ct[n,s,2] <- (PIDcatct[n,s,3] \* sum(N[30:34]) +

 PIDcatct[n,s,4] \* sum(N[35:44])) /

 sum(N[30:44])

 PIDcat1624\_2544ct[n,s,3] <- (PIDcatct[n,s,1] \* sum(N[16:19]) +

 PIDcatct[n,s,2] \* sum(N[20:24]) +

 PIDcatct[n,s,3] \* sum(N[25:34]) +

 PIDcatct[n,s,4] \* sum(N[35:44])) /

 sum(N[16:44])

 }

 }

for (a in 1:3) {

 for (n in 2:4) {

 sumsevPIDcat1624\_2544ct[n,a] <- sum(PIDcat1624\_2544ct[n, ,a])

 }

 sum2sevPIDcat1624\_2544ct[a] <-sum(sumsevPIDcat1624\_2544ct[2:4,a])

 for (s in 1:3) {

 sumnumPIDcat1624\_2544ct[s,a] <- sum(PIDcat1624\_2544ct[2:4,s,a])

 }

 sum2numPIDcat1624\_2544ct[a] <- sum(sumnumPIDcat1624\_2544ct[ ,a])

 }

**# Probability of EP given PID and conception**

**# Likelihood**

**# Westrom progression to EP by number of episodes**

for (n in 1:4) {

 r.EPnum[n] ~ dbin(EPnum[n],n.EPnum[n])

 }

**# Westrom progression to EP by severity in women with 1 PID**

for (s in 1:3) {

 r.EPsev[s] ~ dbin(EPsev[s],n.EPsev[s])

 }

**# Westrom progression to EP by age in women with 1 PID**

for (a in 1:2) {

 r.EPage[a] ~ dbin(EPage[a],n.EPage[a])

 }

**# functional retionship between model (see below) parameters and likelihood**

**EPnum[1] <- (control[1] \* 713 + control[2] \* 199) / 912**

for (n in 2:4) {

 EPnum[n] <- (((PIDconctoEP[n,1,1] \* n.EPsev[1] +

 PIDconctoEP[n,2,1] \* n.EPsev[2] +

 PIDconctoEP[n,3,1] \* n.EPsev[3]) /

 sum(n.EPsev[1:3])) \* n.EPage[1]

 +

 ((PIDconctoEP[n,1,2] \* n.EPsev[1] +

 PIDconctoEP[n,2,2] \* n.EPsev[2] +

 PIDconctoEP[n,3,2] \* n.EPsev[3]) /

 sum(n.EPsev[1:3])) \* n.EPage[2]) /

 sum(n.EPage[1:2])

 }

for (s in 1:3) {

 EPsev[s] <- (PIDconctoEP[2,s,1] \* n.EPage[1] +

 PIDconctoEP[2,s,2] \* n.EPage[2]) /

 sum(n.EPage[1:2])

 }

for (a in 1:2) {

 EPage[a] <- (PIDconctoEP[2,1,a] \* n.EPsev[1] +

 PIDconctoEP[2,2,a] \* n.EPsev[2] +

 PIDconctoEP[2,3,a] \* n.EPsev[3]) /

 sum(n.EPsev[1:3])

 }

**# probability of EP by age - all PID**

**# assume EPs occur 5yrs into Westrom (8.9 year foll) for distribution of EP #age**

**# Progression probabilities by age, severity, and number**

# progress[n,s,a] n: number of PIDs 0,1,2,3+, sev mild,mod,sev, a: age<=29, #30+

**# model**

for (s in 1:3) {

 for (a in 1:2) {

 PIDconctoEP[1,s,a] <- EPnum[1] # constant across age!!!

 for (n in 2:4) {

 logit(PIDconctoEP[n,s,a]) <- beta0 + beta1[n] + beta2[s] + beta3[a]

 }

 }

 }

control[1] <- alphatemp \* propEP.1624 / (propEP.1624 + propEP.2544)

control[2] <- alphatemp \* propEP.2544 / (propEP.1624 + propEP.2544)

logit(alphatemp) <- alpha

**# Priors**

# alpha control group thing

alpha ~ dnorm(0,0.0001)

# beta0 = 1 PID, mild, young,

beta0 ~ dnorm(0,0.0001)

# beta1[n] = effect of 0,1,2or3 PIDs beta1[1] = 0, beta1[2] = 0

beta1[1] <- 0

beta1[2] <- 0

for (n in 3:4) {

 beta1[n] ~ dnorm(0,0.0001)

 }

# beta2[s] = effect of severity: beta2[1] = 0

beta2[1] <- 0

for (s in 2:3) {

 beta2[s] ~ dnorm(0,0.0001)

 }

# beta3[a] = effect of age: beta3[1] = 0

beta3[1] <- 0

beta3[2] ~ dnorm(0,0.0001)

**# RESIDUAL DEVIANCE**

for (m in 1:4) {

 dev1[m] <- 2 \* (r.EPnum[m] \* log(r.EPnum[m] / (EPnum[m] \* n.EPnum[m])) +

 (n.EPnum[m] - r.EPnum[m]) \* log((n.EPnum[m] - r.EPnum[m]) /

 (n.EPnum[m] - (n.EPnum[m] \* EPnum[m]))))

 }

for (s in 1:3) {

 dev2[s] <- 2 \* (r.EPsev[s] \* log(r.EPsev[s] / (EPsev[s] \* n.EPsev[s])) +

 (n.EPsev[s] - r.EPsev[s]) \* log((n.EPsev[s] - r.EPsev[s]) /

 (n.EPsev[s] - (n.EPsev[s] \* EPsev[s]))))

 }

for (a in 1:2) {

 dev3[a] <- 2 \* (r.EPage[a] \* log(r.EPage[a] / (EPage[a] \* n.EPage[a])) +

 (n.EPage[a] - r.EPage[a]) \* log((n.EPage[a] - r.EPage[a]) /

 (n.EPage[a] - (n.EPage[a] \* EPage[a]))))

 }

sumdev1 <- sum(dev1[])

sumdev2 <- sum(dev2[])

sumdev3 <- sum(dev3[])

sumdev.tot <- sumdev1 + sumdev2 + sumdev3

**# Progression probabilities by age, diagnostic status, and number**

# progress[n,s,a] n: number of PIDs 0,1,2,3+,

# 1: undiagnosed (mild),

# 2: diagnosed outside of Hospital (mild),

# 3: hospital diagnosed(overall Westrom)

# a: age<=29, 30+

**# model 1**

#for (n in 2:4) {

# for (a in 1:2) {

# PIDconctoEP**2**[n,1,a] <- ((PIDconctoEP[n,1,a] \* n.EPsev[1] +

# PIDconctoEP[n,2,a] \* n.EPsev[2] +

# PIDconctoEP[n,3,a] \* n.EPsev[3]) /

# sum(n.EPsev[1:3])) - control[a]

# PIDconctoEP**2**[n,2,a] <- ((PIDconctoEP[n,1,a] \* n.EPsev[1] +

# PIDconctoEP[n,2,a] \* n.EPsev[2] +

# PIDconctoEP[n,3,a] \* n.EPsev[3]) /

# sum(n.EPsev[1:3])) - control[a]

# PIDconctoEP**2**[n,3,a] <- ((PIDconctoEP[n,1,a] \* n.EPsev[1] +

# PIDconctoEP[n,2,a] \* n.EPsev[2] +

# PIDconctoEP[n,3,a] \* n.EPsev[3]) /

# sum(n.EPsev[1:3])) - control[a]

# }

# }

**# model 2**

#for (n in 2:4) {

# for (a in 1:2) {

# PIDconctoEP**2**[n,1,a] <- PIDconctoEP[n,1,a] - control[a]

# PIDconctoEP**2**[n,2,a] <- (PIDconctoEP[n,1,a] \* n.EPsev[1] +

# PIDconctoEP[n,2,a] \* n.EPsev[2] +

# PIDconctoEP[n,3,a] \* n.EPsev[3]) /

# sum(n.EPsev[1:3]) - control[a]

# PIDconctoEP**2**[n,3,a] <- (PIDconctoEP[n,1,a] \* n.EPsev[1] +

# PIDconctoEP[n,2,a] \* n.EPsev[2] +

# PIDconctoEP[n,3,a] \* n.EPsev[3]) /

# sum(n.EPsev[1:3]) - control[a]

# }

# }

**# model 3**

#for (n in 2:4) {

# for (a in 1:2) {

# PIDconctoEP**2**[n,1,a] <- PIDconctoEP[n,1,a] - control[a]

# PIDconctoEP**2**[n,2,a] <- PIDconctoEP[n,1,a] - control[a]

# PIDconctoEP**2**[n,3,a] <- (PIDconctoEP[n,1,a] \* n.EPsev[1] +

# PIDconctoEP[n,2,a] \* n.EPsev[2] +

# PIDconctoEP[n,3,a] \* n.EPsev[3]) /

# sum(n.EPsev[1:3]) - control[a]

# }

# }

**# model 4**

for (n in 2:4) {

 for (a in 1:2) {

 A[n,a] <- PIDconctoEP[n,1,a] - control[a]

 PIDconctoEP**2**[n,1,a] ~ dunif(0,A[n,a])

 PIDconctoEP**2**[n,2,a] <- PIDconctoEP[n,1,a] - control[a]

 PIDconctoEP**2**[n,3,a] <- (PIDconctoEP[n,1,a] \* n.EPsev[1] +

 PIDconctoEP[n,2,a] \* n.EPsev[2] +

 PIDconctoEP[n,3,a] \* n.EPsev[3]) /

 sum(n.EPsev[1:3]) - control[a]

 }

 }

**# model 5**

#for (n in 2:4) {

# for (a in 1:2) {

# PIDconctoEP**2**[n,1,a] <- 0

# PIDconctoEP**2**[n,2,a] <- PIDconctoEP[n,1,a] - control[a]

# PIDconctoEP**2**[n,3,a] <- (PIDconctoEP[n,1,a] \* n.EPsev[1] +

# PIDconctoEP[n,2,a] \* n.EPsev[2] +

# PIDconctoEP[n,3,a] \* n.EPsev[3]) /

# sum(n.EPsev[1:3]) - control[a]

# }

# }

**# Risk a pregnancy will be ectopic due to PID by age**

**# all PID**

for (n in 2:4) {

 for (s in 1:3) {

 EPs[n,s,1] <- PIDcat[n,s,1] \* PIDconctoEP**2**[n,s,1]

 EPs[n,s,2] <- PIDcat[n,s,2] \* PIDconctoEP**2**[n,s,1]

 EPs[n,s,3] <- PIDcat[n,s,3] \* (

 (PIDconctoEP**2**[n,s,1] \* N2529 + PIDconctoEP**2**[n,s,2] \* N3034)

 / N2534)

 EPs[n,s,4] <- PIDcat[n,s,4] \* PIDconctoEP**2**[n,s,2]

 }

 }

for (a in 1:4) {

 EPbyage[a] <- sum(EPs[2:4 , ,a])

 }

EPbyage.tot <- (EPbyage[1] \* sum(N[16:19]) +

 EPbyage[2] \* sum(N[20:24]) +

 EPbyage[3] \* sum(N[25:34]) +

 EPbyage[4] \* sum(N[35:44])) /

 sum(N[16:44])

**# non-CT related PID**

for (n in 2:4) {

 for (s in 1:3) {

 EPsnct[n,s,1] <- PIDcatnct[n,s,1] \* PIDconctoEP**2**[n,s,1]

 EPsnct[n,s,2] <- PIDcatnct[n,s,2] \* PIDconctoEP**2**[n,s,1]

 EPsnct[n,s,3] <- PIDcatnct[n,s,3] \* (

 (PIDconctoEP**2**[n,s,1] \* N2529 + PIDconctoEP**2**[n,s,2] \* N3034)

 / N2534)

 EPsnct[n,s,4] <- PIDcatnct[n,s,4] \* PIDconctoEP**2**[n,s,2]

 }

 }

for (a in 1:4) {

 EPbyagenct[a] <- sum(EPsnct[2:4 , ,a])

 }

EPbyagenct.tot <- (EPbyagenct[1] \* sum(N[16:19]) +

 EPbyagenct[2] \* sum(N[20:24]) +

 EPbyagenct[3] \* sum(N[25:34]) +

 EPbyagenct[4] \* sum(N[35:44])) /

 sum(N[16:44])

for (a in 1:4) {

 EPduetoCTbyage[a] <- EPbyage[a] - EPbyagenct[a]

 }

EPduetoCTbyage.tot <- (EPduetoCTbyage[1] \* sum(N[16:19]) +

 EPduetoCTbyage[2] \* sum(N[20:24]) +

 EPduetoCTbyage[3] \* sum(N[25:34]) +

 EPduetoCTbyage[4] \* sum(N[35:44])) /

 sum(N[16:44])

**# proportion of PID related EPs due to CT**

for (a in 1:4) {

 propCTofPIDEPs[a] <- 1 - (EPbyagenct[a] / EPbyage[a])

 }

propCTofPIDEPs.tot <- (propCTofPIDEPs[1] \* sum(N[16:19]) +

 propCTofPIDEPs[2] \* sum(N[20:24]) +

 propCTofPIDEPs[3] \* sum(N[25:34]) +

 propCTofPIDEPs[4] \* sum(N[35:44])) /

 sum(N[16:44])

propCTofPIDEPs.tot2 <- (propCTofPIDEPs[1] \* r.HESEP[1] +

 propCTofPIDEPs[2] \* r.HESEP[2] +

 propCTofPIDEPs[3] \* r.HESEP[3] +

 propCTofPIDEPs[4] \* r.HESEP[4]) /

 sum(r.HESEP[1:4])

**# Population EP rate by age - 2002**

**# Conception rate by age - 2002**

concdata[1] ~ dbin(conc[1],N1619)

concdata[2] ~ dbin(conc[2],N2024)

concdata[3] ~ dbin(conc[3],N2534)

concdata[4] ~ dbin(conc[4],N3544)

**# priors**

for (a in 1:4) {

 conc[a] ~ dbeta(1,1)

 }

**# EP rate by age - 2002**

r.HESEP[1] ~ dbin(HESEP[1],N1619)

r.HESEP[2] ~ dbin(HESEP[2],N2024)

r.HESEP[3] ~ dbin(HESEP[3],N2534)

r.HESEP[4] ~ dbin(HESEP[4],N3544)

**# priors**

for (a in 1:4) {

 HESEP[a] ~ dbeta(1,1)

 }

**# Analysis of retrospective data**

**# Odds Ratio**

ln.OR.l ~ dnorm(1.22,28.5)

ln.OR.u ~ dnorm(1.69,48.5)

OR.l <- exp(ln.OR.l)

OR.u <- exp(ln.OR.u)

**# Population attributable fraction**

for (a in 1:4) {

 pi.PID[a] <- 1 - numPIDs[1,a]

 pi.PIDnct[a] <- 1 - numPIDsnct[1,a]

 gamma.l[a] <- (pi.PID[a] \* (OR.l - 1)) / (pi.PID[a] \* (OR.l - 1) + 1)

 gamma.u[a] <- (pi.PID[a] \* (OR.u - 1)) / (pi.PID[a] \* (OR.u - 1) + 1)

 gamma.lnct[a] <- (pi.PIDnct[a] \* (OR.l - 1)) /

 (pi.PIDnct[a] \* (OR.l - 1) + 1)

 gamma.unct[a] <- (pi.PIDnct[a] \* (OR.u - 1)) /

 (pi.PIDnct[a] \* (OR.u - 1) + 1)

 gamma.lct[a] <- gamma.l[a] - gamma.lnct[a]

 gamma.uct[a] <- gamma.u[a] - gamma.unct[a]

 }

pi.PID1624 <- (pi.PID[1] \* sum(N[16:19]) + pi.PID[2] \* sum(N[20:24])) /

 sum(N[16:24])

pi.PID2544 <- (pi.PID[3] \* sum(N[25:34]) + pi.PID[4] \* sum(N[35:44])) /

 sum(N[25:44])

pi.PID1644 <- (pi.PID[1] \* sum(N[16:19]) + pi.PID[2] \* sum(N[20:24]) +

 pi.PID[3] \* sum(N[25:34]) + pi.PID[4] \* sum(N[35:44])) /

 sum(N[16:44])

pi.PID1624nct <- (pi.PIDnct[1] \* sum(N[16:19]) +

 pi.PIDnct[2] \* sum(N[20:24])) / sum(N[16:24])

pi.PID2544nct <- (pi.PIDnct[3] \* sum(N[25:34]) +

 pi.PIDnct[4] \* sum(N[35:44])) / sum(N[25:44])

pi.PID1644nct <- (pi.PIDnct[1] \* sum(N[16:19]) +

 pi.PIDnct[2] \* sum(N[20:24]) +

 pi.PIDnct[3] \* sum(N[25:34]) +

 pi.PIDnct[4] \* sum(N[35:44])) /

 sum(N[16:44])

gamma.l.1624 <- (pi.PID1624 \* (OR.l - 1)) / (pi.PID1624 \* (OR.l - 1) + 1)

gamma.u.1624 <- (pi.PID1624 \* (OR.u - 1)) / (pi.PID1624 \* (OR.u - 1) + 1)

gamma.l.2544 <- (pi.PID2544 \* (OR.l - 1)) / (pi.PID2544 \* (OR.l - 1) + 1)

gamma.u.2544 <- (pi.PID2544 \* (OR.u - 1)) / (pi.PID2544 \* (OR.u - 1) + 1)

gamma.l.1644 <- (pi.PID1644 \* (OR.l - 1)) / (pi.PID1644 \* (OR.l - 1) + 1)

gamma.u.1644 <- (pi.PID1644 \* (OR.u - 1)) / (pi.PID1644 \* (OR.u - 1) + 1)

gamma.l.1624nct <- (pi.PID1624nct \* (OR.l - 1)) /

 (pi.PID1624nct \* (OR.l - 1) + 1)

gamma.u.1624nct <- (pi.PID1624nct \* (OR.u - 1)) /

 (pi.PID1624nct \* (OR.u - 1) + 1)

gamma.l.2544nct <- (pi.PID2544nct \* (OR.l - 1)) /

 (pi.PID2544nct \* (OR.l - 1) + 1)

gamma.u.2544nct <- (pi.PID2544nct \* (OR.u - 1)) /

 (pi.PID2544nct \* (OR.u - 1) + 1)

gamma.l.1644nct <- (pi.PID1644nct \* (OR.l - 1)) /

 (pi.PID1644nct \* (OR.l - 1) + 1)

gamma.u.1644nct <- (pi.PID1644nct \* (OR.u - 1)) /

 (pi.PID1644nct \* (OR.u - 1) + 1)

gamma.l.1624ct <- gamma.l.1624 - gamma.l.1624nct

gamma.u.1624ct <- gamma.u.1624 - gamma.u.1624nct

gamma.l.2544ct <- gamma.l.2544 - gamma.l.2544nct

gamma.u.2544ct <- gamma.u.2544 - gamma.u.2544nct

gamma.l.1644ct <- gamma.l.1644 - gamma.l.1644nct

gamma.u.1644ct <- gamma.u.1644 - gamma.u.1644nct

numEPSduePID.l <- gamma.l.1644 \* sum(r.HESEP[ ])

numEPSduePID.u <- gamma.u.1644 \* sum(r.HESEP[ ])

numEPSdueCT.l <- gamma.l.1644 \* sum(r.HESEP[ ]) \* propCTofPIDEPs.tot2

numEPSdueCT.u <- gamma.u.1644 \* sum(r.HESEP[ ]) \* propCTofPIDEPs.tot2

**# Proportion of conceptions that are Ectopic - 2002**

for (a in 1:4) {

 propEP[a] <- HESEP[a] / conc[a]

 }

propEP.1624 <- (propEP[1] \* sum(N[16:19]) + propEP[2] \* sum(N[20:24])) /

 sum(N[16:24])

propEP.2544 <- (propEP[3] \* sum(N[25:34]) + propEP[4] \* sum(N[35:44])) /

 sum(N[25:44])

propEP.tot <- (propEP[1] \* sum(N[16:19]) +

 propEP[2] \* sum(N[20:24]) +

 propEP[3] \* sum(N[25:34]) +

 propEP[4] \* sum(N[35:44])) /

 sum(N[16:44])

**# Retrospective estimate of prop EPS due to PID - 2002**

for (a in 1:4) {

 retroEPfromPID.l[a] <- propEP[a] \* gamma.l[a]

 retroEPfromPID.u[a] <- propEP[a] \* gamma.u[a]

 }

retroEPfromPID.l.tot <- propEP.tot \* gamma.l.1644

retroEPfromPID.u.tot <- propEP.tot \* gamma.u.1644

**# Proportion due to PID and CT - 2002**

for (a in 1:4) {

 propPID[a] <- EPbyage[a] / propEP[a]

 propifnoct[a] <- EPbyagenct[a] / propEP[a]

 propnct[a] <- propPID[a] - propifnoct[a]

 }

propPID.tot <- (propPID[1] \* sum(N[16:19]) +

 propPID[2] \* sum(N[20:24]) +

 propPID[3] \* sum(N[25:34]) +

 propPID[4] \* sum(N[35:44])) /

 sum(N[16:44])

propifnoct.tot <- (propifnoct[1] \* sum(N[16:19]) +

 propifnoct[2] \* sum(N[20:24]) +

 propifnoct[3] \* sum(N[25:34]) +

 propifnoct[4] \* sum(N[35:44])) /

 sum(N[16:44])

propnct.tot <- (propnct[1] \* sum(N[16:19]) +

 propnct[2] \* sum(N[20:24]) +

 propnct[3] \* sum(N[25:34]) +

 propnct[4] \* sum(N[35:44])) /

 sum(N[16:44])

**}**

# Data

list(

# PID incidence, r-infection rate, Etological fractions and re-infection rate

mu = c(-3.865, -3.595, -3.964, -4.402, -0.5856, 0.7104, 0.4815, 0.3117, 0.3277, 1.919),

Omega = structure(.Data =c(

573.855, -49.264, -10.817, -3.992, 323.371, 97.687, -112.351, -30.198, -5.401, -2.713,

-49.264, 301.419, -14.194, -3.299, 147.591, -78.069, 176.480, -72.284, -11.551, 2.012,

-10.817, -14.194, 105.908, -20.063, 37.684, -13.275, -46.636, 236.305, -118.245, 0.515,

-3.992, -3.299, -20.063, 57.687, 19.172, -2.664, -8.418, -125.677, 138.728, 0.088,

323.371, 147.591, 37.684, 19.172, 351.620, 1.363, -1.443, -1.164, 1.537, 0.030,

97.687, -78.069, -13.275, -2.664, 1.363, 285.706, -324.389, -85.846, -16.864, -7.957,

-112.351, 176.480, -46.636, -8.418, -1.443, -324.389, 734.182, -295.254, -50.046, 8.090,

-30.198, -72.284, 236.305, -125.677, -1.164, -85.846, -295.254, 1512.713, -758.618, 3.088,

-5.401, -11.551, -118.245, 138.728, 1.537, -16.864, -50.046, -758.618, 839.634, 0.705,

-2.713, 2.012, 0.515, 0.088, 0.030, -7.957, 8.090, 3.088, 0.705, 13.711),

.Dim = c(10,10)),

**# Population sizes from census, age =1...44 - 2002**

N=c(NA,NA,NA,NA,NA, NA,NA,NA,NA,NA, NA,NA,NA,NA,NA,

305500,306300,296400,291400,294800, 310100,313900,305600,294700,295000,

304100,317000,329600,349600,370300, 380900,376900,387800,390900,399400,

401200,402600,398700,391900,381900, 370900,356200,349000,343800),

**# Routine PID data**

**# HES PID data - 2002**

r.HESPID = c(1233,3101,9756,10526), # 16-19, 20-24, 25-34, 35-44

**# KC-60 PID data**

r.kc602008 = c(2900,3972,3538,1253), #16-19, 20-24, 25-34, 35-44

**# predicted GPRD data - 2002**

r.GPRDPID = c(5083,8842,14932,9609), # 16-19, 20-24, 25-34, 35-44

**# Prospective PID to EP data (Westrom 560238)**

**# NB data are for first pregnancy**

# results by number of episodes

r.EPnum = c(6,61,24,15), # 0,1,2,3 diagnosed PIDs

n.EPnum = c(439,912,148,39), # 0,1,2,3 diagnosed PIDs

# number by severity, in women with one PID episode only - read from graph!

r.EPsev = c(7,19,35), # mild, moderate, severe

n.EPsev = c(309,420,183), # mild, moderate, severe

# number by age group, in women with one PID episode only

r.EPage = c(39,22), # <25, 25-35 at index?

n.EPage = c(713,199), # <25, 25-35 at index?

**# HES EP data - 2002**

r.HESEP = c(291,1154,4340,1840), # 16-19, 20-24, 25-34, 35-44

**# Conceptions data - 2002**

# 13-15, 15-17, 15-19, 20-24, 25-29, 30-34, 35-39, 40+

#conc = c(7900,42000,**97100**,167800,199400,204300,98900,**19600**),

# 16-19, 20-24, 25-34, 35-44 **- Guessed!**

concdata = c(**90000**,167800,403700,**113900**)

**)**

**# Initial values 1**

**list(**

Y = c(-5,-5,-5,-5,-1,-5,-5,-5,-5,-1),

HESPID = c(0.01,0.01,0.01,0.01),

kc602008 = c(0.01,0.01,0.01,0.01),

GPRDPID = c(0.01,0.01,0.01,0.01),

alpha = -2,

beta0 = -2,

beta1 = c(NA,NA,0.5,0.5),

beta2 = c(NA,0.05,0.05),

beta3 = c(NA,0.5),

HESEP = c(0.01,0.01,0.01,0.01),

conc = c(0.01,0.01,0.01,0.01),

philap = 0.3

**)**

**# Initial values 2**

**list(**

Y = c(-1,-1,-1,-1,-1,-1,-1,-1,-1,-1),

HESPID = c(0.1,0.1,0.1,0.1),

kc602008 = c(0.1,0.1,0.1,0.1),

GPRDPID = c(0.1,0.1,0.1,0.1),

alpha = -5,

beta0 = -5,

beta1 = c(NA,NA,0.05,0.05),

beta2 = c(NA,0.5,0.5),

beta3 = c(NA,0.05),

HESEP = c(0.1,0.1,0.1,0.1),

conc = c(0.1,0.1,0.1,0.1),

philap = 0.6

**)**

# Appendix 14

#### WinBUGS code and data files for prediction of TFI prevalence

The following WinBUGS code processes the survey data on infertility (Table 37) and delivers estimates of the prevalence on primary, secondary, and total infertility, and the prevalence of TFI (Table 38). The data input for the Oakley study is based on a WinBUGs simulation exercise, shown later in this Appendix.

**model {**

for (i in 1:4) {

 p[i] ~ dbeta(x[1],x[2]) # common beta for pr(1ary infertility)

 }

for (i in 5:7) {

 p[i] ~ dbeta(x[3],x[4]) # common beta for pr(2ndary infertility)

 }

for (i in 8:8) {

 p[i] ~ dbeta(.5,.5) # Jeffreys priors p[8]

 }

f ~ dunif(0.89,1) # adjustment for length of follow-up

for (i in 1:4) {

 x[i] ~ dexp(.001) # priors for beta parameters

 }

for (i in 1:8) {

 r[i] ~ dbin(p[i],n[i]) # likelihood

 rhat[i] <- p[i] \* n[i] # expected value of the numerators

 dev[i] <- 2 \* (r[i] \* (log(r[i])-log(rhat[i])) + (n[i]-r[i]) \*

 (log(n[i]- r[i]) - log(n[i]-rhat[i]))) # Deviance contribution

 }

x[5] <- x[1]/sum(x[1:2]) # estimate of pr(1ary infertility)

x[6] <- x[3]/sum(x[3:4]) # estimate of pr(2ndary infertility)

x[7] <- x[6] \* f # adjusted pr(2ndary)

x[8] <- x[5] + x[7] # total infertility

x[9] <- x[8] \* p[8] # total TFI

dev[9] <- sum(dev[1:8]) # total residual deviance

dev[10] <- sum(dev[1:7]) # totall res dev for fertility data

}

**# Initial Values 1**

list(x=c(4,6,4,6,NA,NA,NA,NA,NA),p=c(.4,.4,.4,.4,.4,.4,.4,.4),f=.92))

**# Initial values 2**

list(x=c(20,20,20,20,NA,NA,NA,NA,NA),p=c(.2,.2,.2,.2,.2,.2,.2,.2),f=.96))

**# Data**

r[] n[]

# primary (Bhattacharya, Templeton, Gunnell, Oakley (adjusted))

79 2347

27 766

31 1609

158.3 6128

# secondary ( Bhattacharya, Templeton, Gunnell)

5 2347

17 766

41 1609

# Proportion of total infertility (including males) due to TFI (Maheshwari)

442 1782

END

This simulation model to adjust the Oakley % primary infertility data for the proportion of women who were involuntarily childless.

model {

p[i] ~ dbeta(.5,.5)

r[i] ~ dbin(p[i],n[i]) }

p[3] <- p[1]/p[2]

}

# Initial values

list(p=c(.5,.5,NA))

# Data

list(r=c(159,2910),n=c(6584,3113))

Results:

**node mean sd MC error 2.5% median 97.5% start sample**

p[1] 0.02422 0.001891 7.665E-6 0.02068 0.02418 0.02806 10001 60000

p[2] 0.9347 0.004425 1.77E-5 0.9257 0.9348 0.943 10001 60000

p[3] 0.02592 0.002027 8.251E-6 0.02211 0.02587 0.03004 10001 60000

The output is used as follows: the probability of primary infertility from the Oakley study is taken to be the reported probability divided by the proportion who were not involuntarily childless: (159/6584) / (2910/3113) = 0.02583. The variance of this estimate, taken from the posterior estimate of p[3] is 0.0020272. The effective denominator is therefore the solution to *n* in 0.0025834 x (1-0.0025834)/*n =* 0.0020272, which gives *n*=6128, with numerator 158.3. These figures are used as “data” in the previous code.

**Appendix 15**

*WinBUGS code for the predictions from prospective models (Table 40)*

**# TFI analysis**

**model {**

**# Population (disecrete time) model - 2 year higher rate**

**# PID incidence and ct GP re-infection rate ratio informative priors**

Y[1:10] ~ dmnorm(mu[], Omega[ , ])

for (ag in 1:4) {

 log(lambda.PID[ag]) <- Y[ag]

 EF[ag] <- Y[ag+5] \* Y[ag+5]

 }

logit(psi) <- Y[5]

log(eta1) <- Y[10]

**# WBDEV call**

for (ag in 1:4) {

 input1[ag] <- lambda.PID[ag]

 input1[ag+6] <- EF[ag]

 }

input1[5] <- psi

input1[6] <- eta1

philap ~ dbeta(12,16)

input1[11] <- philap

for (i in 1:29) {

 input2[i] <- N[i+15]

 }

solution[1:609] <- cumulativePIDlap(input1[1:11],input2[1:29])

**# number of previous PIDs [n,a] n = number: 0,1,2,3; a = age-group**

for (i in 1:4) {

 numPIDs[2,i] <- solution[i]

 numPIDs[3,i] <- solution[i+4]

 numPIDs[4,i] <- solution[i+8]

 numPIDsnct[2,i] <- solution[i+12]

 numPIDsnct[3,i] <- solution[i+16]

 numPIDsnct[4,i] <- solution[i+20]

 }

**# number of previous PIDs in women aged 44, n = number: 0,1,2,3;**

num44[2] <- solution[58]

num44[3] <- solution[87]

num44[4] <- solution[116]

num44nct[2] <- solution[145]

num44nct[3] <- solution[174]

num44nct[4] <- solution[203]

**# population denominators - 2002**

N1619 <- sum(N[16:19])

N2024 <- sum(N[20:24])

N2534 <- sum(N[25:34])

N3544 <- sum(N[35:44])

N1644 <- sum(N[16:44])

N1617 <- sum(N[16:17])

N1820 <- sum(N[18:20])

N2124 <- sum(N[21:24])

N2529 <- sum(N[25:29])

N3044 <- sum(N[30:44])

N1819 <- sum(N[18:19])

N3034 <- sum(N[30:34])

**# Distribution of PID severity - 2002**

**# hospital**

r.HESPID[1] ~ dbin(HESPID[1],N1619)

r.HESPID[2] ~ dbin(HESPID[2],N2024)

r.HESPID[3] ~ dbin(HESPID[3],N2534)

r.HESPID[4] ~ dbin(HESPID[4],N3544)

**# priors**

for (a in 1:4) {

 HESPID[a] ~ dbeta(1,1)

 }

**# kc60**

r.kc602008[1] ~ dbin(kc602008[1],N1619)

r.kc602008[2] ~ dbin(kc602008[2],N2024)

r.kc602008[3] ~ dbin(kc602008[3],N2534)

r.kc602008[4] ~ dbin(kc602008[4],N3544)

**# priors**

for (a in 1:4) {

 kc602008[a] ~ dbeta(1,1)

 }

**# data by age not available until 2008**

**# 13421 / 12117 is the ratio of totals for 2002 and GUMCAD data for 2008**

**# assumes GUMCAD data representative of all kc60**

**# assumes age distribution is the same in 2002 as 2008**

for (a in 1:4) {

 kc60[a] <- kc602008[a] \* 13421 / 12117

 }

**# GPRD**

r.GPRDPID[1] ~ dbin(GPRDPID[1],N1619)

r.GPRDPID[2] ~ dbin(GPRDPID[2],N2024)

r.GPRDPID[3] ~ dbin(GPRDPID[3],N2534)

r.GPRDPID[4] ~ dbin(GPRDPID[4],N3544)

**# priors**

for (a in 1:4) {

 GPRDPID[a] ~ dbeta(1,1)

 }

# Distribution

for (a in 1:4) {

pmin[a] <- kc60[a] + max(HESPID[a],GPRDPID[a])

pmax[a] <- kc60[a] + HESPID[a] + GPRDPID[a]

X[a] ~ dunif(pmin[a],pmax[a])

 hospdiag[a] <- psi \* HESPID[a] / X[a]

 milddiag[a] <- 1 - hospdiag[a] - undiag[a]

 undiag[a] <- (1 - psi)

 }

**# proportion with PIDs by each age, severity, and number**

**# all PID**

# n = 1

for (s in 1:3) {

 for (a in 1:4) {

 PIDcat[1,s,a] <- 1 - (PIDcat[2,s,a] + PIDcat[3,s,a] + PIDcat[4,s,a])

 }

 PIDcat44[1,s] <- 1 - (PIDcat44[2,s] + PIDcat44[3,s] + PIDcat44[4,s])

 }

# n = 2-4

for (n in 2:4) {

 for (a in 1:4) {

 PIDcat[n,1,a] <- numPIDs[n,a] \* undiag[a]

 PIDcat[n,2,a] <- numPIDs[n,a] \* milddiag[a]

 PIDcat[n,3,a] <- numPIDs[n,a] \* hospdiag[a]

 }

 PIDcat44[n,1] <- num44[n] \* undiag[4]

 PIDcat44[n,2] <- num44[n] \* milddiag[4]

 PIDcat44[n,3] <- num44[n] \* hospdiag[4]

 }

**# non-CT related PID**

# n = 1

for (s in 1:3) {

 for (a in 1:4) {

 PIDcatnct[1,s,a] <- 1 - (PIDcatnct[2,s,a] + PIDcatnct[3,s,a] +

 PIDcatnct[4,s,a])

 }

 PIDcat44nct[1,s] <- 1 - (PIDcat44nct[2,s] + PIDcat44nct[3,s] +

 PIDcat44nct[4,s])

 }

# n = 2-4

for (n in 2:4) {

 for (a in 1:4) {

 PIDcatnct[n,1,a] <- numPIDsnct[n,a] \* undiag[a]

 PIDcatnct[n,2,a] <- numPIDsnct[n,a] \* milddiag[a]

 PIDcatnct[n,3,a] <- numPIDsnct[n,a] \* hospdiag[a]

 }

 PIDcat44nct[n,1] <- num44nct[n] \* undiag[4]

 PIDcat44nct[n,2] <- num44nct[n] \* milddiag[4]

 PIDcat44nct[n,3] <- num44nct[n] \* hospdiag[4]

 }

**# Probability of TFI given PID by age, severity, number of PIDs-Westrom**

**# Likelihood**

for (a in 1:2) {

 for (s in 1:3) {

 onePIDsev[a,s] <- p[a,s,1]

 r.onePIDsev[a,s] ~ dbin(onePIDsev[a,s],n.onePIDsev[a,s])

 }

 twoPID[a] <- (p[a,1,2] \* n.onePIDsev[a,1] +

 p[a,2,2] \* n.onePIDsev[a,2] +

 p[a,3,2] \* n.onePIDsev[a,3]) / sum(n.onePIDsev[a, ])

 r.twoPID[a] ~ dbin(twoPID[a],n.twoPID[a])

 threePID[a] <- (p[a,1,3] \* n.onePIDsev[a,1] +

 p[a,2,3] \* n.onePIDsev[a,2] +

 p[a,3,3] \* n.onePIDsev[a,3]) / sum(n.onePIDsev[a, ])

 r.threePID[a] ~ dbin(threePID[a],n.threePID[a])

 }

**# model**

for (a in 1:2) {

 for (s in 1:3) {

 for (n in 1:3) {

 logit(p[a,s,n]) <- beta0 + beta1[a] + beta2[s] + beta3[n]

 PIDtoTFI[n+1,a,s] <- p[a,s,n]

 }

 }

 }

for (a in 1:2) {

 onePID[a] <- (onePIDsev[a,1] \* n.onePIDsev[a,1] +

 onePIDsev[a,2] \* n.onePIDsev[a,2] +

 onePIDsev[a,3] \* n.onePIDsev[a,3]) /

 sum(n.onePIDsev[a, ])

 }

**# priors**

beta0 ~ dnorm(0,0.0001)

beta1[1] <- 0

beta1[2] ~ dnorm(0,0.0001)

beta2[1] <- 0

beta2[2] ~ dnorm(0,0.0001)

beta2[3] ~ dnorm(0,0.0001)

beta3[1] <- 0

beta3[2] ~ dnorm(0,0.0001)

beta3[3] ~ dnorm(0,0.0001)

**# RESIDUAL DEVIANCE**

for (a in 1:2) {

 for (s in 1:3) {

 dev1[a,s] <- 2 \* (r.onePIDsev[a,s] \* log(r.onePIDsev[a,s] /

 (onePIDsev[a,s] \* n.onePIDsev[a,s])) +

 (n.onePIDsev[a,s] - r.onePIDsev[a,s]) \*

 log((n.onePIDsev[a,s] - r.onePIDsev[a,s]) /

 (n.onePIDsev[a,s] - (n.onePIDsev[a,s] \*

 onePIDsev[a,s]))))

 }

 dev2[a] <- 2 \* (r.twoPID[a] \* log(r.twoPID[a] /

 (twoPID[a] \* n.twoPID[a])) +

 (n.twoPID[a] - r.twoPID[a]) \*

 log((n.twoPID[a] - r.twoPID[a]) /

 (n.twoPID[a] - (n.twoPID[a] \* twoPID[a]))))

 dev3[a] <- 2 \* (r.threePID[a] \* log(r.threePID[a] /

 (threePID[a] \* n.threePID[a])) +

 (n.threePID[a] - r.threePID[a]) \*

 log((n.threePID[a] - r.threePID[a]) /

 (n.threePID[a] - (n.threePID[a] \* threePID[a]))))

 }

sumdev1 <- sum(dev1[ , ])

sumdev2 <- sum(dev2[])

sumdev3 <- sum(dev3[])

sumdev.tot <- sumdev1 + sumdev2 + sumdev3

**# Progression probabilities by age, diagnostic status, and number**

# progress[n,s,a] n: number of PIDs 0,1,2,3+,

# 1: undiagnosed (mild),

# 2: diagnosed outside of Hospital (mild),

# 3: hospital diagnosed(overall Westrom)

# a: age<=29, 30+

**# model 1**

for (n in 1:3) {

 for (a in 1:2) {

 PIDtoTFI2[n,1,a] <- (PIDtoTFI[2,a,1] \* n.onePIDsev[a,1] +

 PIDtoTFI[2,a,2] \* n.onePIDsev[a,2] +

 PIDtoTFI[2,a,3] \* n.onePIDsev[a,3]) /

 sum(n.onePIDsev[a, ])

 PIDtoTFI2[n,2,a] <- (PIDtoTFI[2,a,1] \* n.onePIDsev[a,1] +

 PIDtoTFI[2,a,2] \* n.onePIDsev[a,2] +

 PIDtoTFI[2,a,3] \* n.onePIDsev[a,3]) /

 sum(n.onePIDsev[a, ])

 PIDtoTFI2[n,3,a] <- (PIDtoTFI[2,a,1] \* n.onePIDsev[a,1] +

 PIDtoTFI[2,a,2] \* n.onePIDsev[a,2] +

 PIDtoTFI[2,a,3] \* n.onePIDsev[a,3]) /

 sum(n.onePIDsev[a, ])

 }

 }

**# model 2**

#for (n in 1:3) {

# for (a in 1:2) {

# PIDtoTFI2[n,1,a] <- PIDtoTFI[2,a,1]

# PIDtoTFI2[n,2,a] <- (PIDtoTFI[2,a,1] \* n.onePIDsev[a,1] +

# PIDtoTFI[2,a,2] \* n.onePIDsev[a,2] +

# PIDtoTFI[2,a,3] \* n.onePIDsev[a,3]) /

# sum(n.onePIDsev[a, ])

# PIDtoTFI2[n,3,a] <- (PIDtoTFI[2,a,1] \* n.onePIDsev[a,1] +

# PIDtoTFI[2,a,2] \* n.onePIDsev[a,2] +

# PIDtoTFI[2,a,3] \* n.onePIDsev[a,3]) /

# sum(n.onePIDsev[a, ])

# }

# }

**# model 3**

#for (n in 1:3) {

# for (a in 1:2) {

# PIDtoTFI2[n,1,a] <- PIDtoTFI[2,a,1]

# PIDtoTFI2[n,2,a] <- PIDtoTFI[2,a,1]

# PIDtoTFI2[n,3,a] <- (PIDtoTFI[2,a,1] \* n.onePIDsev[a,1] +

# PIDtoTFI[2,a,2] \* n.onePIDsev[a,2] +

# PIDtoTFI[2,a,3] \* n.onePIDsev[a,3]) /

# sum(n.onePIDsev[a, ])

# }

# }

**# model 4**

#for (n in 1:3) {

# for (a in 1:2) {

# A[n,a] ~ dunif(0,PIDtoTFI[2,a,1])

# PIDtoTFI2[n,1,a] <- A[n,a]

# PIDtoTFI2[n,2,a] <- PIDtoTFI[2,a,1]

# PIDtoTFI2[n,3,a] <- (PIDtoTFI[2,a,1] \* n.onePIDsev[a,1] +

# PIDtoTFI[2,a,2] \* n.onePIDsev[a,2] +

# PIDtoTFI[2,a,3] \* n.onePIDsev[a,3]) /

# sum(n.onePIDsev[a, ])

# }

# }

**# model 5**

#for (n in 1:3) {

# for (a in 1:2) {

# PIDtoTFI2[n,1,a] <- 0

# PIDtoTFI2[n,2,a] <- PIDtoTFI[2,a,1]

# PIDtoTFI2[n,3,a] <- (PIDtoTFI[2,a,1] \* n.onePIDsev[a,1] +

# PIDtoTFI[2,a,2] \* n.onePIDsev[a,2] +

# PIDtoTFI[2,a,3] \* n.onePIDsev[a,3]) /

# sum(n.onePIDsev[a, ])

# }

# }

**# TFIs by age**

**# all PID**

for (n in 2:4) {

 for (s in 1:3) {

 TFIs[n,s,1] <- PIDcat[n,s,1] \* PIDtoTFI2[n-1,s,1]

 TFIs[n,s,2] <- PIDcat[n,s,2] \* PIDtoTFI2[n-1,s,1]

 TFIs[n,s,3] <- PIDcat[n,s,3] \*

 (PIDtoTFI2[n-1,s,1] \* N2529 + PIDtoTFI2[n-1,s,2] \* N3034) /

 N2534

 TFIs[n,s,4] <- PIDcat[n,s,4] \* PIDtoTFI2[n-1,s,2]

 TFI44[n,s] <- PIDcat44[n,s] \* PIDtoTFI2[n-1,s,2]

 }

 }

for (a in 1:4) {

 TFIbyage[a] <- sum(TFIs[2:4 , ,a])

 }

TFI44tot <- sum(TFI44[2:4 , ])

TFIbyage.tot <- (TFIbyage[1] \* sum(N[16:19]) +

 TFIbyage[2] \* sum(N[20:24]) +

 TFIbyage[3] \* sum(N[25:34]) +

 TFIbyage[4] \* sum(N[35:44])) /

 sum(N[16:44])

**# non-CT related PID**

for (n in 2:4) {

 for (s in 1:3) {

 TFIsnct[n,s,1] <- PIDcatnct[n,s,1] \* PIDtoTFI2[n-1,s,1]

 TFIsnct[n,s,2] <- PIDcatnct[n,s,2] \* PIDtoTFI2[n-1,s,1]

 TFIsnct[n,s,3] <- PIDcatnct[n,s,3] \*

 (PIDtoTFI2[n-1,s,1] \* N2529 + PIDtoTFI2[n-1,s,2] \*

 N3034) / N2534

 TFIsnct[n,s,4] <- PIDcatnct[n,s,4] \* PIDtoTFI2[n-1,s,2]

 TFI44nct[n,s] <- PIDcat44nct[n,s] \* PIDtoTFI2[n-1,s,2]

 }

 }

for (a in 1:4) {

 TFIbyagenct[a] <- sum(TFIsnct[2:4 , ,a])

 }

TFI44totnct <- sum(TFI44nct[2:4 , ])

TFIbyagenct.tot <- (TFIbyagenct[1] \* sum(N[16:19]) +

 TFIbyagenct[2] \* sum(N[20:24]) +

 TFIbyagenct[3] \* sum(N[25:34]) +

 TFIbyagenct[4] \* sum(N[35:44])) /

 sum(N[16:44])

for (a in 1:4) {

 TFIduetoCTbyage[a] <- TFIbyage[a] - TFIbyagenct[a]

 }

TFIduetoCTbyage.tot <- (TFIduetoCTbyage[1] \* sum(N[16:19]) +

 TFIduetoCTbyage[2] \* sum(N[20:24]) +

 TFIduetoCTbyage[3] \* sum(N[25:34]) +

 TFIduetoCTbyage[4] \* sum(N[35:44])) /

 sum(N[16:44])

**# proportion of PID related TFIs due to CT**

for (a in 1:4) {

 propCTofPIDTFIs[a] <- 1 - (TFIbyagenct[a] / TFIbyage[a])

 }

propCTofPIDTFIs44 <- 1 - (TFI44totnct / TFI44tot)

propCTofPIDTFIs.tot <- (propCTofPIDTFIs[1] \* sum(N[16:19]) +

 propCTofPIDTFIs[2] \* sum(N[20:24]) +

 propCTofPIDTFIs[3] \* sum(N[25:34]) +

 propCTofPIDTFIs[4] \* sum(N[35:44])) /

 sum(N[16:44])

**}**

**# Data**

**list(**

**# PID incidence, r-infection rate, Etological fractions and re-infection #rate**

mu = c(-3.865, -3.595, -3.964, -4.402, -0.5856, 0.7104, 0.4815, 0.3117, 0.3277, 1.919),

Omega = structure(.Data =c(

573.855, -49.264, -10.817, -3.992, 323.371, 97.687, -112.351, -30.198, -5.401, -2.713,

-49.264, 301.419, -14.194, -3.299, 147.591, -78.069, 176.480, -72.284, -11.551, 2.012,

-10.817, -14.194, 105.908, -20.063, 37.684, -13.275, -46.636, 236.305, -118.245, 0.515,

-3.992, -3.299, -20.063, 57.687, 19.172, -2.664, -8.418, -125.677, 138.728, 0.088,

323.371, 147.591, 37.684, 19.172, 351.620, 1.363, -1.443, -1.164, 1.537, 0.030,

97.687, -78.069, -13.275, -2.664, 1.363, 285.706, -324.389, -85.846, -16.864, -7.957,

-112.351, 176.480, -46.636, -8.418, -1.443, -324.389, 734.182, -295.254, -50.046, 8.090,

-30.198, -72.284, 236.305, -125.677, -1.164, -85.846, -295.254, 1512.713, -758.618, 3.088,

-5.401, -11.551, -118.245, 138.728, 1.537, -16.864, -50.046, -758.618, 839.634, 0.705,

-2.713, 2.012, 0.515, 0.088, 0.030, -7.957, 8.090, 3.088, 0.705, 13.711),

.Dim = c(10,10)),

**# Population sizes from census, age =1...44 - 2002**

N=c(NA,NA,NA,NA,NA, NA,NA,NA,NA,NA, NA,NA,NA,NA,NA,

305500,306300,296400,291400,294800, 310100,313900,305600,294700,295000,

304100,317000,329600,349600,370300, 380900,376900,387800,390900,399400,

401200,402600,398700,391900,381900, 370900,356200,349000,343800),

**# Routine PID data**

**# HES PID data - 2002**

r.HESPID = c(1233,3101,9756,10526), # 16-19, 20-24, 25-34, 35-44

**# KC-60 PID data**

r.kc602008 = c(2900,3972,3538,1253), #16-19, 20-24, 25-34, 35-44

**# predicted GPRD data - 2002**

r.GPRDPID = c(5083,8842,14932,9609), # 16-19, 20-24, 25-34, 35-44

**# Westrom Progression data (A cohort study of 1,844)-assumes a=1 goes to 29**

r.onePIDsev = structure(.Data =c(

2,23,34,

0,5,15

),

.Dim = c(2,3)),

n.onePIDsev = structure(.Data =c(

241,361,169,

71,89,60

),

.Dim = c(2,3)),

r.twoPID = c(29,7), n.twoPID = c(158,27),

r.threePID = c(23,3), n.threePID = c(61,4),

# Note: 0 TFIs in 451 control women so assume zero.

**)**

**# Initial values 1**

**list(**

Y = c(-5,-5,-5,-5,-1,-5,-5,-5,-5,-1),

HESPID = c(0.01,0.01,0.01,0.01),

kc602008 = c(0.01,0.01,0.01,0.01),

GPRDPID = c(0.01,0.01,0.01,0.01),

philap = 0.3,

beta0 = -2,

beta1 = c(NA,0.5),

beta2 = c(NA,0.05,0.05),

beta3 = c(NA,0.5,0.5)

**)**

A = structure(.Data =c(

0.01,0.01,

0.01,0.01,

0.01,0.01

),

.Dim = c(3,2)),

**)**

**# Initial values 2**

**list(**

Y = c(-1,-1,-1,-1,-1,-1,-1,-1,-1,-1),

HESPID = c(0.1,0.1,0.1,0.1),

kc602008 = c(0.1,0.1,0.1,0.1),

GPRDPID = c(0.1,0.1,0.1,0.1),

philap = 0.6,

beta0 = -5,

beta1 = c(NA,0.05),

beta2 = c(NA,0.5,0.5),

beta3 = c(NA,0.05,0.05),

**)**

A = structure(.Data =c(

0.1,0.1,

0.1,0.1,

0.1,0.1

),

.Dim = c(3,2)),

**)**

# Appendix 17

#### WinBUGS code applied to serology data

**model {**

# cycles through the different combinations of input parameters

for (a in 1:**numobs**) {

 for (i in 1:17) {

 for (g in 1:2) {

 r[i,a,g] ~ dbin(p[g,a,i],n[i,a,g])

 rhat[i,a,g] <- p[g,a,i]\*n[i,a,g]

 dev[i,a,g] <- 2 \* (r[i,a,g] \* (log(r[i,a,g])-log(rhat[i,a,g])) +

 (n[i,a,g]-r[i,a,g]) \* (log(n[i,a,g]-r[i,a,g]) -

 log(n[i,a,g]-rhat[i,a,g])))

 }

 sumdev1[i,a] <- sum(dev[i,a,])

 p[1,a,i] <- pi[1,a] \* se[i,a] + (1-pi[1,a]) \* fp[i,a]

 p[2,a,i] <- pi[3,a]\*sec[i,a] + pi[2,a] \* se[i,a] + pi[4,a] \* fp[i,a]

 logit(se[i,a]) <- lse[i,a]

 logit(sec[i,a]) <- lse[i,a] + dse[a]

 lse[i,a] ~ dnorm(0,.01)

 lfp[i,a] <- lse[i,a] - res[test[i],a]

 logit(fp[i,a]) <- lfp[i,a]

 }

 for (j in 1:5) {

 res[j,a] ~ dnorm(mean[a],prec[a])I(0,)

 }

 z[a] ~ dbeta(1,1)

 x[a] ~ dbeta(1,1)

 # Ever exposed in control group

 pi[1,a] <- ctprev[a]

 # Ever exposed in Non CT caused TFI

 pi[2,a] <- (1 - pi[3,a]) \* (pi[1,a] + z[a] \* (1 - pi[1,a]))

 # Proportion of TFI caused by C

 pi[3,a] <- x[a]

 # the negatices in the tfi group

 pi[4,a] <- (1 - pi[3,a]) \* ((1 - z[a]) \* (1 - pi[1,a]))

 # sum check

 pi[5,a] <- pi[2,a] + pi[3,a] + pi[4,a]

 sumdev2[a] <- sum(sumdev1[,a])

 }

**}**