Appendix A. WinBUGS Codes

```
#BUGS code for LAREhom

model {

  # Pain
  sw1[1] <- 0
  for (i in 1:N1) {
    mean1[i] <- mu1[s1[i]] + delta1[i]*(1-equals(t1[i],b1[i]))
    y1[i] ~ dnorm(mean1[i], prec1[i])
    se1[i] <- sd1[i]/sqrt(n1[i])
    prec1[i] <- 1/pow(se1[i],2)
    delta1[i] ~ dnorm(md1[i], taud1[i])
    taud1[i] <- tau1 * (1+equals(m1[i],3)/3)
    md1[i] <- d1[t1[i]] - d1[b1[i]] + equals(m1[i],3)*sw1[i]

    fitted1[i] <- mean1[i]
    res1[i] <- y1[i] - mean1[i]
    dev1[i] <- (y1[i] - mean1[i])*(y1[i] - mean1[i])*prec1[i]
  }

  resdev1 <- sum(dev1[])

  for (i in 2:N1) { sw1[i] <- (delta1[i-1] - d1[t1[i-1]] + d1[b1[i-1]])/2 }
  for (j in 1:NS1) { mu1[j] ~ dnorm(0, 0.0001) }

  d1[1] <- 0
  for (k in 2:NT) { d1[k] ~ dnorm(0, 0.0001) }

  sig1 ~ dunif(0.01, 10)
  tau1 <- 1/pow(sig1, 2)

  # Disability
  sw2[1] <- 0
  for (i in 1:N2) {
    mean2[i] <- mu2[s2[i]] + delta2[i]*(1-equals(t2[i],b2[i]))
    y2[i] ~ dnorm(mean2[i], prec2[i])
    se2[i] <- sd2[i]/sqrt(n2[i])
    prec2[i] <- 1/pow(se2[i],2)
    delta2[i] ~ dnorm(md2[i], taud2[i])
    taud2[i] <- tau2 * (1+equals(m2[i],3)/3)
    md2[i] <- d2[t2[i]] - d2[b2[i]] + equals(m2[i],3)*sw2[i]

    fitted2[i] <- mean2[i]
    res2[i] <- y2[i] - mean2[i]
    dev2[i] <- (y2[i] - mean2[i])*(y2[i] - mean2[i])*prec2[i]
```

```
}

resdev2 <- sum(dev2[])

for (i in 2:N2) { sw2[i] <- (delta2[i-1] - d2[t2[i-1]] + d2[b2[i-1]])/2 }
for (j in 1:NS2) { mu2[j] ~ dnorm(0, 0.0001) }

d2[1] <- 0
for (k in 2:NT) { d2[k] ~ dnorm(0, 0.0001) }

sig2 ~ dunif(0.01, 10)
tau2 <- 1/pow(sig2, 2)

resdev <- resdev1 + resdev2

# ranking
for (i in 1:nP1) { mmu1[i] <- mu1[study1[i]] }
for (i in 1:nP2) { mmu2[i] <- mu2[study2[i]] }
mP1 <- mean(mmu1[])        # Take average of mu1[]
SD1 <- sd(mmu1[])
mP2 <- mean(mmu2[])        # Take average of mu1[]
SD2 <- sd(mmu2[])

for (k in 1:NT) {
  T.1[k] <- mP1 + d1[k]
  rk.1[k] <- rank(T.1[], k)
  best1.1[k] <- equals(rk.1[k], 1)
  best2.1[k] <- equals(rk.1[k], 2)
  best12.1[k] <- best1.1[k] + best2.1[k]

  T.2[k] <- mP2 + d2[k]
  rk.2[k] <- rank(T.2[], k)
  best1.2[k] <- equals(rk.2[k], 1)
  best2.2[k] <- equals(rk.2[k], 2)
  best12.2[k] <- best1.2[k] + best2.2[k]

  T.eq[k] <- 0.5*T.1[k] + 0.5*T.2[k]
  rk.eq[k] <- rank(T.eq[], k)
  best1.eq[k] <- equals(rk.eq[k], 1)
  best2.eq[k] <- equals(rk.eq[k], 2)
  best12.eq[k] <- best1.eq[k] + best2.eq[k]

  T.pain[k] <- 0.8*T.1[k] + 0.2*T.2[k]
  rk.pain[k] <- rank(T.pain[], k)
  best1.pain[k] <- equals(rk.pain[k], 1)
  best2.pain[k] <- equals(rk.pain[k], 2)
```

```
    best12.pain[k] <- best1.pain[k] + best2.pain[k]

    T.dis[k] <- 0.2*T.1[k] + 0.8*T.2[k]
    rk.dis[k] <- rank(T.dis[], k)
    best1.dis[k] <- equals(rk.dis[k], 1)
    best2.dis[k] <- equals(rk.dis[k], 2)
    best12.dis[k] <- best1.dis[k] + best2.dis[k]
 }
}

# Data

list(N1=111, NS1=51, N2=57, NS2=26, NT=11,
      study1 = c(1,2,5,7,10, 12,14,15,16,21, 28,31,32,35,36, 40,42,43,44,45,
                    46,47,48,49,51), nP1=25,
       study2 = c(1,2,5,7,10, 12,14,15,16,21, 24), nP2=11)

# Data - pain

s1[]    t1[]    y1[]    sd1[]   n1[]    b1[]    m1[]
1       1       7.700   2.300   17      1       1
1       7       0.700   1.000   17      1       2
2       1       4.770   2.120   50      1       1
2       7       3.770   1.730   52      1       2
3       2       2.800   1.220   149     2       1
3       7       2.280   1.200   144     2       2
3       9       2.420   1.440   146     2       3
4       2       5.500   2.070   50      2       1
4       7       4.590   2.400   52      2       2
5       1       4.873   2.063   124     1       1
5       8       4.607   2.457   125     1       2
6       2       3.780   2.345   23      2       1
6       9       2.560   1.962   23      2       2
7       1       5.070   2.530   193     1       1
7       9       4.060   2.530   199     1       2
8       3       6.300   1.900   10      3       1
8       4       5.000   3.200   10      3       2
8       5       5.500   2.700   10      3       3
9       3       4.440   3.510   33      3       1
9       4       4.730   3.480   38      3       2
9       5       4.030   3.300   32      3       3
10      1       4.250   2.367   94      1       1
10      7       3.858   2.201   88      1       2
11      2       3.750   2.500   23      2       1
11      8       3.650   0.350   22      2       2
12      1       3.370   1.780   19      1       1
12      10      2.410   2.050   22      1       2
13      3       4.180   1.659   19      3       1
```

13	6	3.740	2.360	39	3	2
14	1	2.959	2.394	70	1	1
14	9	2.255	2.068	72	1	2
15	1	2.380	1.403	27	1	1
15	8	1.810	1.403	27	1	2
15	9	1.560	1.400	25	1	3
16	1	3.459	2.355	50	1	1
16	7	3.523	2.193	45	1	2
17	3	4.000	2.600	33	3	1
17	11	3.900	2.000	34	3	2
18	3	3.178	1.784	20	3	1
18	6	2.834	2.136	20	3	2
19	2	3.940	2.220	68	2	1
19	7	3.090	1.540	69	2	2
20	9	1.800	1.600	24	9	1
20	10	1.300	1.200	24	9	2
21	1	1.686	1.057	15	1	1
21	10	1.314	1.143	29	1	2
22	3	6.670	1.780	30	3	1
22	11	5.250	1.900	30	3	2
23	3	6.900	2.000	23	3	1
23	4	3.800	2.200	32	3	2
23	5	4.600	2.500	31	3	3
24	2	2.560	1.800	36	2	1
24	7	2.300	1.954	33	2	2
25	3	3.500	2.900	20	3	1
25	6	2.200	2.800	20	3	2
26	3	6.750	2.375	10	3	1
26	6	5.250	1.425	10	3	2
27	1	3.300	1.500	34	1	1
27	9	3.000	1.500	34	1	2
28	2	2.080	2.090	17	2	1
28	7	2.190	1.670	17	2	2
29	7	3.800	1.600	23	7	1
29	8	2.400	1.600	23	7	2
30	1	4.000	0.743	6	1	1
30	9	1.471	0.643	9	1	2
31	1	5.385	1.528	35	1	1
31	9	5.190	1.657	32	1	2
32	2	1.397	1.242	21	2	1
32	7	1.660	1.463	19	2	2
33	2	1.426	1.026	18	2	1
33	6	2.094	1.712	20	2	2
34	1	3.095	2.030	78	1	1
34	2	2.550	1.945	82	1	2
34	7	3.110	2.010	80	1	3

35	1	4.030	2.080	18	1	1
35	2	4.000	1.560	18	1	2
35	7	3.190	1.850	16	1	3
36	2	5.990	2.400	18	2	1
36	6	5.180	2.110	20	2	2
37	3	4.100	2.600	10	3	1
37	6	0.700	0.700	12	3	2
38	6	0.600	0.100	25	6	1
38	9	1.040	0.270	25	6	2
39	1	4.180	1.950	108	1	1
39	7	3.755	2.400	109	1	2
40	7	3.840	2.750	32	7	1
40	8	3.700	1.810	32	7	2
41	1	3.550	1.700	34	1	1
41	9	2.400	1.750	34	1	2
42	1	4.930	2.020	8	1	1
42	6	5.350	0.970	8	1	2
43	1	2.764	2.252	14	1	1
43	7	1.422	2.202	14	1	2
44	1	1.320	0.760	30	1	1
44	10	1.280	0.740	30	1	2
45	1	3.360	1.540	28	1	1
45	9	2.280	1.690	27	1	2
46	1	2.096	1.844	12	1	1
46	6	1.165	1.671	11	1	2
47	1	3.650	1.700	36	1	1
47	9	2.100	1.500	36	1	2
47	10	2.150	1.150	36	1	3
48	1	4.400	1.400	66	1	1
48	9	3.600	0.700	66	1	2
48	10	2.700	1.900	66	1	3
49	2	7.200	6.630	98	2	1
49	7	6.710	6.880	100	2	2
50	1	3.250	1.650	44	1	1
50	9	2.450	1.650	45	1	2
51	2	4.556	0.467	36	2	1
51	9	3.667	0.422	37	2	2

END

#Data - disability

s2[]	t2[]	y2[]	sd2[]	n2[]	b2[]	m2[]
1	1	6.360	0.790	17	1	1
1	7	1.280	0.970	17	1	2
2	1	5.960	2.320	50	1	1
2	7	3.740	2.690	52	1	2
3	2	2.250	1.225	149	2	1

3	7	1.800	1.200	144	2	2
3	9	1.850	1.225	146	2	3
4	2	6.180	2.750	50	2	1
4	7	6.070	2.950	52	2	2
5	1	3.757	2.237	124	1	1
5	8	3.110	1.833	125	1	2
6	2	3.920	2.936	23	2	1
6	9	3.660	2.869	23	2	2
7	1	3.045	1.720	193	1	1
7	9	2.695	1.805	199	1	2
8	3	5.100	1.700	10	3	1
8	4	5.500	3.000	10	3	2
8	5	5.100	2.300	10	3	3
9	3	4.630	3.540	33	3	1
9	4	4.930	3.630	38	3	2
9	5	4.400	3.440	32	3	3
10	1	1.850	1.374	94	1	1
10	7	1.958	1.539	88	1	2
11	2	5.980	1.940	23	2	1
11	8	6.600	2.150	22	2	2
12	1	2.990	1.630	19	1	1
12	10	2.806	2.327	22	1	2
13	3	4.590	1.681	19	3	1
13	6	3.960	2.425	39	3	2
14	1	3.660	1.636	70	1	1
14	9	2.881	1.633	72	1	2
15	1	3.860	1.351	27	1	1
15	8	3.700	1.351	27	1	2
15	9	3.610	1.350	25	1	3
16	1	0.357	0.280	50	1	1
16	7	0.428	0.368	45	1	2
17	3	4.010	1.583	33	3	1
17	11	3.469	1.615	34	3	2
18	3	3.420	1.654	20	3	1
18	6	2.933	2.034	20	3	2
19	2	1.930	1.880	68	2	1
19	7	1.850	2.260	69	2	2
20	9	1.800	1.300	24	9	1
20	10	1.200	1.000	24	9	2
21	1	4.490	1.750	15	1	1
21	10	3.560	2.090	29	1	2
22	3	4.618	1.331	30	3	1
22	11	4.525	1.717	30	3	2
23	3	4.850	1.750	23	3	1
23	4	3.850	2.030	32	3	2
23	5	3.680	1.650	31	3	3

```
24      1       2.500  0.850  10      1       1
24      9       2.300  0.840  10      1       2
25      3       3.893  2.723  41      3       1
25      7       1.927  1.826  42      3       2
26      2       6.559  0.899  78      2       1
26      7       6.286  1.038  80      2       2
END


# Inits1

list(mu1 = c(0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0,
                 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0),
     d1 = c(NA,0,0,0,0, 0,0,0,0,0, 0),  sig1 = 1,
     mu2 = c(0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0),
     d2 = c(NA,0,0,0,0, 0,0,0,0,0, 0), sig2=1)

# Inits2

list(mu1 = c(1,1,1,1,1, 1,1,1,1,1, 1,1,1,1,1, 1,1,1,1,1, 1,1,1,1,1, 1,1,1,1,1,
                 1,1,1,1,1, 1,1,1,1,1, 1,1,1,1,1, 1,1,1,1,1, 1),
     d1 = c(NA,1,1,1,1, 1,1,1,1,1, 1),  sig1 = 0.5,
     mu2 = c(1,1,1,1,1, 1,1,1,1,1, 1,1,1,1,1, 1,1,1,1,1, 1,1,1,1,1, 1),
     d2 = c(NA,1,1,1,1, 1,1,1,1,1, 1), sig2=0.5)


#BUGS code for CBREhom

model {

  for (j in 1:NS) {
    for (h in 1:NO) {
      delta[j, 1, h] <- 0
    }
  }

  for (i in 1:N) {
    mean[i] <- mu[s[i],l[i]] + delta[s[i], t[i], l[i]]
    y[i] ~ dnorm(mean[i], prec[i])
    se[i] <- sd[i]/sqrt(n[i])
    prec[i] <- 1/pow(se[i],2)
    fitted[i] <- mean[i]
    res[i] <- y[i] - mean[i]
    dev[i] <- (y[i] - mean[i])*(y[i] - mean[i])*prec[i]
  }

  for (j in 1:NS) {
```

```
  for (k in 2:NT) {
    delta[j, k, 1:NO] ~ dmnorm(d[k-1, 1:NO], invR[1:NO, 1:NO])
  }
}

for (j in 1:NS) {
  for (h in 1:NO) {
    mu[j,h] ~ dnorm(mP[h], tauP[h])
  }
}

for (h in 1:NO) {
  mP[h] ~ dnorm(0, 0.0001)
  tauP[h] <- 1/pow(sdP[h], 2)
  sdP[h] ~ dunif(0.01, 10)
}

for (k in 1:(NT-1)) { d[k, 1:NO] ~ dmnorm(md[1:NO], invcovd[1:NO, 1:NO]) }
invR[1:NO, 1:NO] ~ dwish(Omega[1:NO, 1:NO], 2)
R[1:NO, 1:NO] <- inverse(invR[ , ])

sig1 <- sqrt(R[1,1])
sig2 <- sqrt(R[2,2])
rho <- R[1,2]/sqrt(R[1,1]*R[2,2])

sumdev <- sum(dev[])

# ranking
dd1[1] <- 0
dd2[1] <- 0
for (k in 2:NT) {
  dd1[k] <- d[k-1,1]
  dd2[k] <- d[k-1, 2]
}

for (k in 1:NT) {
  T.1[k] <- mP[1] + dd1[k]
  rk.1[k] <- rank(T.1[], k)
  best1.1[k] <- equals(rk.1[k], 1)
  best2.1[k] <- equals(rk.1[k], 2)
  best12.1[k] <- best1.1[k] + best2.1[k]

  T.2[k] <- mP[2] + dd2[k]
  rk.2[k] <- rank(T.2[], k)
  best1.2[k] <- equals(rk.2[k], 1)
  best2.2[k] <- equals(rk.2[k], 2)
```

```
    best12.2[k] <- best1.2[k] + best2.2[k]

    T.eq[k] <- 0.5*T.1[k] + 0.5*T.2[k]
    rk.eq[k] <- rank(T.eq[], k)
    best1.eq[k] <- equals(rk.eq[k], 1)
    best2.eq[k] <- equals(rk.eq[k], 2)
    best12.eq[k] <- best1.eq[k] + best2.eq[k]

    T.pain[k] <- 0.8*T.1[k] + 0.2*T.2[k]
    rk.pain[k] <- rank(T.pain[], k)
    best1.pain[k] <- equals(rk.pain[k], 1)
    best2.pain[k] <- equals(rk.pain[k], 2)
    best12.pain[k] <- best1.pain[k] + best2.pain[k]

    T.dis[k] <- 0.2*T.1[k] + 0.8*T.2[k]
    rk.dis[k] <- rank(T.dis[], k)
    best1.dis[k] <- equals(rk.dis[k], 1)
    best2.dis[k] <- equals(rk.dis[k], 2)
    best12.dis[k] <- best1.dis[k] + best2.dis[k]
 }
}


#Data

list(N=168, NT=11, NS=54, NO=2,
md=c(0,0),
invcovd = structure(.Data = c(0.0001,0,
                                        0,0.0001), .Dim = c(2, 2)),
Omega = structure(.Data = c(10,0,0,10), .Dim = c(2, 2))
)
```

s[]	t[]	y[]	sd[]	n[]	l[]
1	1	7.700	2.300	17	1
1	7	0.700	1.000	17	1
1	1	6.360	0.790	17	2
1	7	1.280	0.970	17	2
2	1	4.770	2.120	50	1
2	7	3.770	1.730	52	1
2	1	5.960	2.320	50	2
2	7	3.740	2.690	52	2
3	2	2.800	1.220	149	1
3	7	2.280	1.200	144	1
3	9	2.420	1.440	146	1
3	2	2.250	1.225	149	2
3	7	1.800	1.200	144	2
3	9	1.850	1.225	146	2

4	2	5.500	2.070	50	1
4	7	4.590	2.400	52	1
4	2	6.180	2.750	50	2
4	7	6.070	2.950	52	2
5	1	4.873	2.063	124	1
5	8	4.607	2.457	125	1
5	1	3.757	2.237	124	2
5	8	3.110	1.833	125	2
6	2	3.780	2.345	23	1
6	9	2.560	1.962	23	1
6	2	3.920	2.936	23	2
6	9	3.660	2.869	23	2
7	1	5.070	2.530	193	1
7	9	4.060	2.530	199	1
7	1	3.045	1.720	193	2
7	9	2.695	1.805	199	2
8	3	6.300	1.900	10	1
8	4	5.000	3.200	10	1
8	5	5.500	2.700	10	1
8	3	5.100	1.700	10	2
8	4	5.500	3.000	10	2
8	5	5.100	2.300	10	2
9	3	4.440	3.510	33	1
9	4	4.730	3.480	38	1
9	5	4.030	3.300	32	1
9	3	4.630	3.540	33	2
9	4	4.930	3.630	38	2
9	5	4.400	3.440	32	2
10	1	4.250	2.367	94	1
10	7	3.858	2.201	88	1
10	1	1.850	1.374	94	2
10	7	1.958	1.539	88	2
11	2	3.750	2.500	23	1
11	8	3.650	0.350	22	1
11	2	5.980	1.940	23	2
11	8	6.600	2.150	22	2
12	1	3.370	1.780	19	1
12	10	2.410	2.050	22	1
12	1	2.990	1.630	19	2
12	10	2.806	2.327	22	2
13	3	4.180	1.659	19	1
13	6	3.740	2.360	39	1
13	3	4.590	1.681	19	2
13	6	3.960	2.425	39	2
14	1	2.959	2.394	70	1
14	9	2.255	2.068	72	1

14	1	3.660	1.636	70	2
14	9	2.881	1.633	72	2
15	1	2.380	1.403	27	1
15	8	1.810	1.403	27	1
15	9	1.560	1.400	25	1
15	1	3.860	1.351	27	2
15	8	3.700	1.351	27	2
15	9	3.610	1.350	25	2
16	1	3.459	2.355	50	1
16	7	3.523	2.193	45	1
16	1	0.357	0.280	50	2
16	7	0.428	0.368	45	2
17	3	4.000	2.600	33	1
17	11	3.900	2.000	34	1
17	3	4.010	1.583	33	2
17	11	3.469	1.615	34	2
18	3	3.178	1.784	20	1
18	6	2.834	2.136	20	1
18	3	3.420	1.654	20	2
18	6	2.933	2.034	20	2
19	2	3.940	2.220	68	1
19	7	3.090	1.540	69	1
19	2	1.930	1.880	68	2
19	7	1.850	2.260	69	2
20	9	1.800	1.600	24	1
20	10	1.300	1.200	24	1
20	9	1.800	1.300	24	2
20	10	1.200	1.000	24	2
21	1	1.686	1.057	15	1
21	10	1.314	1.143	29	1
21	1	4.490	1.750	15	2
21	10	3.560	2.090	29	2
22	3	6.670	1.780	30	1
22	11	5.250	1.900	30	1
22	3	4.618	1.331	30	2
22	11	4.525	1.717	30	2
23	3	6.900	2.000	23	1
23	4	3.800	2.200	32	1
23	5	4.600	2.500	31	1
23	3	4.850	1.750	23	2
23	4	3.850	2.030	32	2
23	5	3.680	1.650	31	2
24	2	2.560	1.800	36	1
24	7	2.300	1.954	33	1
25	3	3.500	2.900	20	1
25	6	2.200	2.800	20	1

26	3	6.750	2.375	10	1
26	6	5.250	1.425	10	1
27	1	3.300	1.500	34	1
27	9	3.000	1.500	34	1
28	2	2.080	2.090	17	1
28	7	2.190	1.670	17	1
29	7	3.800	1.600	23	1
29	8	2.400	1.600	23	1
30	1	4.000	0.743	6	1
30	9	1.471	0.643	9	1
31	1	5.385	1.528	35	1
31	9	5.190	1.657	32	1
32	2	1.397	1.242	21	1
32	7	1.660	1.463	19	1
33	2	1.426	1.026	18	1
33	6	2.094	1.712	20	1
34	1	3.095	2.030	78	1
34	2	2.550	1.945	82	1
34	7	3.110	2.010	80	1
35	1	4.030	2.080	18	1
35	2	4.000	1.560	18	1
35	7	3.190	1.850	16	1
36	2	5.990	2.400	18	1
36	6	5.180	2.110	20	1
37	3	4.100	2.600	10	1
37	6	0.700	0.700	12	1
38	6	0.600	0.100	25	1
38	9	1.040	0.270	25	1
39	1	4.180	1.950	108	1
39	7	3.755	2.400	109	1
40	7	3.840	2.750	32	1
40	8	3.700	1.810	32	1
41	1	3.550	1.700	34	1
41	9	2.400	1.750	34	1
42	1	4.930	2.020	8	1
42	6	5.350	0.970	8	1
43	1	2.764	2.252	14	1
43	7	1.422	2.202	14	1
44	1	1.320	0.760	30	1
44	10	1.280	0.740	30	1
45	1	3.360	1.540	28	1
45	9	2.280	1.690	27	1
46	1	2.096	1.844	12	1
46	6	1.165	1.671	11	1
47	1	3.650	1.700	36	1
47	9	2.100	1.500	36	1

```
47      10      2.150  1.150  36      1
48      1       4.400  1.400  66      1
48      9       3.600  0.700  66      1
48      10      2.700  1.900  66      1
49      2       7.200  6.630  98      1
49      7       6.710  6.880  100     1
50      1       3.250  1.650  44      1
50      9       2.450  1.650  45      1
51      2       4.556  0.467  36      1
51      9       3.667  0.422  37      1
52      1       2.500  0.850  10      2
52      9       2.300  0.840  10      2
53      3       3.893  2.723  41      2
53      7       1.927  1.826  42      2
54      2       6.559  0.899  78      2
54      7       6.286  1.038  80      2
END


#Inits1

list(
d=structure(.Data = c(0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0), .Dim=c(10,2)),
mP = c(0,0), sdP = c(1,1),
invR=structure(.Data=c(1,0,0,1), .Dim=c(2,2)),
mu=structure(.Data=c(0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0), .Dim=c(54,2))
)

#Inits2

list(
d=structure(.Data = c(1,1,1,1,1,1,1,1,1,1, 1,1,1,1,1,1,1,1,1,1), .Dim=c(10,2)),
mP = c(1,1), sdP = c(0.5,0.5),
invR=structure(.Data=c(2,0,0,2), .Dim=c(2,2)),
mu=structure(.Data=c(1,1,1,1,1,1,1,1,1,1, 1,1,1,1,1,1,1,1,1,1,
                     1,1,1,1,1,1,1,1,1,1, 1,1,1,1,1,1,1,1,1,1,
                     1,1,1,1,1,1,1,1,1,1, 1,1,1,1,1,1,1,1,1,1,
                     1,1,1,1,1,1,1,1,1,1, 1,1,1,1,1,1,1,1,1,1,
                     1,1,1,1,1,1,1,1,1,1, 1,1,1,1,1,1,1,1,1,1,
                     1,1,1,1,1,1,1,1,1,1, 1,1,1,1,1,1,1,1,1,1,
                     1,1,1,1,1,1,1,1), .Dim=c(54,2))
)
```

```
#BUGS code for ABREhom

model {

  for (i in 1:N) {
    y[i] ~ dnorm(mean[s[i],t[i],l[i]], prec[i])
    prec[i] <- n[i]/pow(sd[i], 2)
    mean[s[i],t[i],l[i]] <- mu[t[i], l[i]] + v[s[i], t[i], l[i]]
    dev[i] <- (y[i]-mean[s[i],t[i],l[i]])*(y[i]-mean[s[i],t[i],l[i]])*prec[i]
    res[i] <- y[i] - mean[s[i],t[i],l[i]]
    fitted[i] <- mean[s[i],t[i],l[i]]
  }
  resdev <- sum(dev[])

  for (j in 1:NS) {
    for (k in 1:NT) {
      v[j, k, 1:NO] ~ dmnorm(meano[1:NO], invRo[1:NO,1:NO])
     }
  }


  invRo[1:NO,1:NO] ~ dwish(Omegao[1:NO, 1:NO], NO)
  Ro[1:NO, 1:NO] <- inverse(invRo[ , ])

  corro <- Ro[1,2]/sqrt(Ro[1,1]*Ro[2,2])
  for (h in 1:NO) { sig[h] <- sqrt(Ro[h,h]) }

  for (k in 1:NT) {
    for (h in 1:NO) {
      mu[k,h] ~ dnorm(0, 0.0001)
    }
  }

  # To compare with CB models
  for (h in 1:NO) {
    for (k in 1:NT) {
      d[k,h] <- mu[k,h] - mu[1,h]
    }
  }

  #rank
  for (k in 1:NT) {
    T.1[k] <- mu[k,1]
    rk.1[k] <- rank(T.1[], k)
    best1.1[k] <- equals(rk.1[k],1)
    best2.1[k] <- equals(rk.1[k],2)
    best12.1[k] <- best1.1[k] + best2.1[k]
```

```
    T.2[k] <- mu[k,2]
    rk.2[k] <- rank(T.2[], k)
    best1.2[k] <- equals(rk.2[k],1)
    best2.2[k] <- equals(rk.2[k],2)
    best12.2[k] <- best1.2[k] + best2.2[k]

    T.eq[k] <- 0.5*T.1[k] + 0.5*T.2[k]
    rk.eq[k] <- rank(T.eq[], k)
    best1.eq[k] <- equals(rk.eq[k],1)
    best2.eq[k] <- equals(rk.eq[k], 2)
    best12.eq[k] <- best1.eq[k] + best2.eq[k]

    T.pain[k] <- 0.8*T.1[k] + 0.2*T.2[k]
    rk.pain[k] <- rank(T.pain[], k)
    best1.pain[k] <- equals(rk.pain[k], 1)
    best2.pain[k] <- equals(rk.pain[k], 2)
    best12.pain[k] <- best1.pain[k] + best2.pain[k]

    T.dis[k] <- 0.2*T.1[k] + 0.8*T.2[k]
    rk.dis[k] <- rank(T.dis[], k)
    best1.dis[k] <- equals(rk.dis[k], 1)
    best2.dis[k] <- equals(rk.dis[k], 2)
    best12.dis[k] <- best1.dis[k] + best2.dis[k]
  }
}


#Data

list(
N=168, NT=11, NS=54, NO=2,
meano= c(0,0),
Omegao = structure(.Data = c(10,0,0,10), .Dim = c(2,2))
)
```

s[]	t[]	y[]	sd[]	n[]	l[]
1	1	7.700	2.300	17	1
1	7	0.700	1.000	17	1
1	1	6.360	0.790	17	2
1	7	1.280	0.970	17	2
2	1	4.770	2.120	50	1
2	7	3.770	1.730	52	1
2	1	5.960	2.320	50	2
2	7	3.740	2.690	52	2
3	2	2.800	1.220	149	1
3	7	2.280	1.200	144	1

3	9	2.420	1.440	146	1
3	2	2.250	1.225	149	2
3	7	1.800	1.200	144	2
3	9	1.850	1.225	146	2
4	2	5.500	2.070	50	1
4	7	4.590	2.400	52	1
4	2	6.180	2.750	50	2
4	7	6.070	2.950	52	2
5	1	4.873	2.063	124	1
5	8	4.607	2.457	125	1
5	1	3.757	2.237	124	2
5	8	3.110	1.833	125	2
6	2	3.780	2.345	23	1
6	9	2.560	1.962	23	1
6	2	3.920	2.936	23	2
6	9	3.660	2.869	23	2
7	1	5.070	2.530	193	1
7	9	4.060	2.530	199	1
7	1	3.045	1.720	193	2
7	9	2.695	1.805	199	2
8	3	6.300	1.900	10	1
8	4	5.000	3.200	10	1
8	5	5.500	2.700	10	1
8	3	5.100	1.700	10	2
8	4	5.500	3.000	10	2
8	5	5.100	2.300	10	2
9	3	4.440	3.510	33	1
9	4	4.730	3.480	38	1
9	5	4.030	3.300	32	1
9	3	4.630	3.540	33	2
9	4	4.930	3.630	38	2
9	5	4.400	3.440	32	2
10	1	4.250	2.367	94	1
10	7	3.858	2.201	88	1
10	1	1.850	1.374	94	2
10	7	1.958	1.539	88	2
11	2	3.750	2.500	23	1
11	8	3.650	0.350	22	1
11	2	5.980	1.940	23	2
11	8	6.600	2.150	22	2
12	1	3.370	1.780	19	1
12	10	2.410	2.050	22	1
12	1	2.990	1.630	19	2
12	10	2.806	2.327	22	2
13	3	4.180	1.659	19	1
13	6	3.740	2.360	39	1

13	3	4.590	1.681	19	2
13	6	3.960	2.425	39	2
14	1	2.959	2.394	70	1
14	9	2.255	2.068	72	1
14	1	3.660	1.636	70	2
14	9	2.881	1.633	72	2
15	1	2.380	1.403	27	1
15	8	1.810	1.403	27	1
15	9	1.560	1.400	25	1
15	1	3.860	1.351	27	2
15	8	3.700	1.351	27	2
15	9	3.610	1.350	25	2
16	1	3.459	2.355	50	1
16	7	3.523	2.193	45	1
16	1	0.357	0.280	50	2
16	7	0.428	0.368	45	2
17	3	4.000	2.600	33	1
17	11	3.900	2.000	34	1
17	3	4.010	1.583	33	2
17	11	3.469	1.615	34	2
18	3	3.178	1.784	20	1
18	6	2.834	2.136	20	1
18	3	3.420	1.654	20	2
18	6	2.933	2.034	20	2
19	2	3.940	2.220	68	1
19	7	3.090	1.540	69	1
19	2	1.930	1.880	68	2
19	7	1.850	2.260	69	2
20	9	1.800	1.600	24	1
20	10	1.300	1.200	24	1
20	9	1.800	1.300	24	2
20	10	1.200	1.000	24	2
21	1	1.686	1.057	15	1
21	10	1.314	1.143	29	1
21	1	4.490	1.750	15	2
21	10	3.560	2.090	29	2
22	3	6.670	1.780	30	1
22	11	5.250	1.900	30	1
22	3	4.618	1.331	30	2
22	11	4.525	1.717	30	2
23	3	6.900	2.000	23	1
23	4	3.800	2.200	32	1
23	5	4.600	2.500	31	1
23	3	4.850	1.750	23	2
23	4	3.850	2.030	32	2
23	5	3.680	1.650	31	2

24	2	2.560	1.800	36	1
24	7	2.300	1.954	33	1
25	3	3.500	2.900	20	1
25	6	2.200	2.800	20	1
26	3	6.750	2.375	10	1
26	6	5.250	1.425	10	1
27	1	3.300	1.500	34	1
27	9	3.000	1.500	34	1
28	2	2.080	2.090	17	1
28	7	2.190	1.670	17	1
29	7	3.800	1.600	23	1
29	8	2.400	1.600	23	1
30	1	4.000	0.743	6	1
30	9	1.471	0.643	9	1
31	1	5.385	1.528	35	1
31	9	5.190	1.657	32	1
32	2	1.397	1.242	21	1
32	7	1.660	1.463	19	1
33	2	1.426	1.026	18	1
33	6	2.094	1.712	20	1
34	1	3.095	2.030	78	1
34	2	2.550	1.945	82	1
34	7	3.110	2.010	80	1
35	1	4.030	2.080	18	1
35	2	4.000	1.560	18	1
35	7	3.190	1.850	16	1
36	2	5.990	2.400	18	1
36	6	5.180	2.110	20	1
37	3	4.100	2.600	10	1
37	6	0.700	0.700	12	1
38	6	0.600	0.100	25	1
38	9	1.040	0.270	25	1
39	1	4.180	1.950	108	1
39	7	3.755	2.400	109	1
40	7	3.840	2.750	32	1
40	8	3.700	1.810	32	1
41	1	3.550	1.700	34	1
41	9	2.400	1.750	34	1
42	1	4.930	2.020	8	1
42	6	5.350	0.970	8	1
43	1	2.764	2.252	14	1
43	7	1.422	2.202	14	1
44	1	1.320	0.760	30	1
44	10	1.280	0.740	30	1
45	1	3.360	1.540	28	1
45	9	2.280	1.690	27	1

```
46	1	2.096	1.844	12	1
46	6	1.165	1.671	11	1
47	1	3.650	1.700	36	1
47	9	2.100	1.500	36	1
47	10	2.150	1.150	36	1
48	1	4.400	1.400	66	1
48	9	3.600	0.700	66	1
48	10	2.700	1.900	66	1
49	2	7.200	6.630	98	1
49	7	6.710	6.880	100	1
50	1	3.250	1.650	44	1
50	9	2.450	1.650	45	1
51	2	4.556	0.467	36	1
51	9	3.667	0.422	37	1
52	1	2.500	0.850	10	2
52	9	2.300	0.840	10	2
53	3	3.893	2.723	41	2
53	7	1.927	1.826	42	2
54	2	6.559	0.899	78	2
54	7	6.286	1.038	80	2
END

#Inits1

list(
mu = structure(.Data = c(0,0,0,0,0, 0,0,0,0,0, 0, 0,0,0,0,0, 0,0,0,0,0, 0), .Dim=c(11,2)),
invRo = structure(.Data=c(1,0,0,1), .Dim=c(2,2))
)

#Inits2

list(
mu = structure(.Data = c(1,1,1,1,1, 1,1,1,1,1, 1, 1,1,1,1,1, 1,1,1,1,1, 1), .Dim=c(11,2)),
invRo = structure(.Data=c(2,0,0,2), .Dim=c(2,2))
)
```