The Impact of Covariance Priors on Arm-based Bayesian Network Meta-Analyses with Binary Outcomes

Supplementary Materials

Zhenxun Wang* Lifeng Lin† James S. Hodges* Haitao Chu*

A Definition of restricted Wishart distribution and its properties

In this appendix, we want to show the equivalence of restricted Wishart distribution and LKJ distribution Lewandowski et al. (2009) in some sense. We start with the definition of Wishart distribution. As we know, if $T \times 1$ vectors $\boldsymbol{\nu_1}, \boldsymbol{\nu_2}, \ldots, \boldsymbol{\nu_K}$ are independent random variables that are multivariate normally distributed with mean $\mathbf{0}$ and variance $\boldsymbol{\Sigma}$, and we have a $T \times K$ matrix $\boldsymbol{\nu} = (\boldsymbol{\nu_1}, \boldsymbol{\nu_2}, \ldots, \boldsymbol{\nu_K})$, then the T-dimensional positive definite matrix $\mathbf{S} = \boldsymbol{\nu}\boldsymbol{\nu}'$ ($\boldsymbol{\nu}'$ is the transpose of $\boldsymbol{\nu}$) follows the Wishart distribution with degree of freedom $K > T - 1$ and positive definite scale matrix $\boldsymbol{\Sigma}$:

$$\mathbf{S} \sim \mathrm{W}_T(K, \boldsymbol{\Sigma}) \equiv \{2^{\frac{1}{2}KT}\Gamma_T(\frac{1}{2}K)\}^{-1}|\boldsymbol{\Sigma}|^{-\frac{1}{2}K}|\mathbf{S}|^{\frac{1}{2}(K-T-1)}\exp(-\frac{1}{2}tr(\boldsymbol{\Sigma}^{-1}\mathbf{S})) \tag{1}$$

---

*Division of Biostatistics, School of Public Health, University of Minnesota, Minneapolis, MN 55455, USA

†Department of Statistics, Florida State University, Tallahassee, FL 32304, USA

where $|\bullet|$ is the determinant, tr is the trace, and Γ_T is the multivariate (T-variate here) gamma function. After that, we can formally define the restricted Wishart distribution based on the Wishart distribution:

Definition 1. A positive definite correlation matrix $\mathbf{R}$ follows the restricted Wishart distribution $\mathrm{RW}_T(m)$, if $\mathbf{R} = \mathbf{\Delta \Sigma \Delta}$, where $\mathbf{\Sigma} \sim \mathrm{W}_T(m, \mathbf{\Psi})$ with $\mathbf{\Psi}$ being diagonal matrix with diagonal entries $\psi_{11}, \psi_{22}, \ldots, \psi_{TT}$, $\mathbf{\Delta}$ is a diagonal matrix with the i^{th} diagonal element $\sigma_{ii}^{-1/2}$ and σ_{ii} is the i^{th} diagonal element of $\mathbf{\Sigma}$.

Theorem 1. If R follows the restricted Wishart distribution $\mathrm{RW}_T(m)$, then
(a) the joint distribution of $\mathbf{R}$ is:

$$f(\mathbf{R}) = b_T(m)|\mathbf{R}|^{\frac{m-T-1}{2}} \tag{2}$$

where

$$b_T(m) = \frac{\Gamma^T(\frac{m}{2})}{\Gamma_T(\frac{m}{2})} \tag{3}$$

is the normalizing constant. When $m = T+1$, R becomes a jointly uniform distribution on a compact subspace of the $T(T-1)/2$ dimensional hypercube $[-1,1]^{T(T-1)/2}$.
(b) this normalizing constant $b_T(m)$ matches the following alternative formula given by Lewandowski et al.Lewandowski et al. (2009):

$$c_T = 2^{\sum_{k=1}^{T-1}(m-k-1)(k-T)} \times \prod_{k=1}^{T-1}[B(\frac{m-k}{2}, \frac{m-k}{2})]^{k-T} \tag{4}$$

where $B(\bullet, \bullet)$ is the beta function.

(c) the marginal distribution of each element ρ_{ij} $(i \neq j)$ in matrix $\mathbf{R}$ is

$$f(\rho_{ij}) = \frac{\Gamma(\frac{m}{2})}{\Gamma(\frac{m-1}{2})\Gamma(\frac{1}{2})}(1-\rho_{ij}^2)^{\frac{m-3}{2}}, \quad -1 \leq \rho_{ij} \leq 1 \tag{5}$$

which is exactly the $Beta(\frac{m-1}{2}, \frac{m-1}{2})$ distribution on $[-1,1]$, and will be $Beta(\frac{T}{2}, \frac{T}{2})$ when $f(\mathbf{R}) = b_T(T+1)$.

Proof of Theorem 1. (a). We first calculate the Jacobian matrix of the transformation $\mathbf{\Sigma} \rightarrow (\sigma_{11}, \ldots, \sigma_{TT}, \mathbf{R})$

$$\mathbf{J}_1 = \mathbf{J}(\sigma_{11} = \sigma_{11}, \ldots, \sigma_{TT} = \sigma_{TT}, \rho_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}}) = \left[\begin{array}{c|c} \mathbf{I}_T & 0 \\ \hline * & \mathbf{C} \end{array}\right] \tag{6}$$

where σ_{ij} is the entry of matrix $\mathbf{\Sigma}$, $\mathbf{I}_T$ is the T-dimensional identity matrix, $\mathbf{C}$ is the $T(T-1)/2$ dimensional diagonal matrix with entries $1/\sqrt{\sigma_{ii}\sigma_{jj}}$ $(i \neq j)$. Since J_1 is a lower triangular matrix, its determinant $|\mathbf{J}_1|$ equals $\prod_{i=1}^{T} \sigma_{ii}^{-(T-1)/2}$ and sub-matrix in (6) given as $*$ need not be derived. Then, given $\mathbf{\Sigma} \sim \mathrm{W}_T(m, \mathbf{\Psi})$, we can derive the joint distribution of $\mathbf{R}$ and $(\sigma_{11}, \ldots, \sigma_{TT})$ is

$$\begin{aligned} f(\mathbf{R}, \sigma_{11}, \ldots, \sigma_{TT}) &= f(\mathbf{\Sigma})|\mathbf{J}_1|^{-1} \\ &= \frac{\Gamma^T(\frac{m}{2})|\mathbf{R}|^{\frac{m-T-1}{2}}}{\Gamma_T(\frac{m}{2})} \prod_{i=1}^{T} \{ \frac{\sigma_{ii}^{\frac{m}{2}-1} \exp(-\frac{\sigma_{ii}}{\psi_{ii}})}{2^{\frac{m}{2}} \psi_{ii}^{\frac{m}{2}} \Gamma(\frac{m}{2})} \} \end{aligned} \tag{7}$$

Clearly, $\frac{\sigma_{ii}}{\psi_{ii}}, i = 1, \ldots, T$ are independently distributed as chi-square with m degrees of freedom. Also, the density of $\boldsymbol{P}$ is

$$f(\mathbf{R}) = \frac{\Gamma^T(\frac{m}{2})}{\Gamma_T(\frac{m}{2})}|\mathbf{R}|^{\frac{m-T-1}{2}} \tag{8}$$

(b). We just need to prove function $f(T,m) = b_T(m)/c_T$ equals 1 for any integer T (≥ 2)

and $m > T-1$. Given $\Gamma_T(\frac{m}{2}) = \pi^{T(T-1)/4}\prod_{k=1}^{T}\Gamma(\frac{m}{2}+\frac{1-k}{2})$ (JamesJames (1964), p. 483), we can simplify $f(T,m)$ as

$$\begin{aligned} f(T,m) &= \frac{\Gamma^T(\frac{m}{2})}{\Gamma_T(\frac{m}{2})}\prod_{k=1}^{T-1}[B(\frac{m-k}{2},\frac{m-k}{2})]^{T-k} \times 2^{\sum_{k=1}^{T-1}(m-k-1)(T-k)} \\ &= \frac{\Gamma^T(\frac{m}{2})}{\pi^{T(T-1)/4}\Gamma(\frac{m}{2})\prod_{k=1}^{T-1}\Gamma(\frac{m-k}{2})}\prod_{k=1}^{T-1}[\frac{\Gamma^2(\frac{m-k}{2})}{\Gamma(m-k)}]^{T-k} \times 2^{\sum_{k=1}^{T-1}(m-k-1)(T-k)} \\ &= \prod_{k=1}^{T-1}[2^{(m-k-1)(T-k)} \times \frac{\Gamma(\frac{m}{2})}{\pi^{T/4}}\frac{\Gamma^{2T-2k-1}(\frac{m-k}{2})}{\Gamma^{T-k}(m-k)}] \end{aligned} \tag{9}$$

We prove that (9) equals 1 using mathematical induction. Start with $T=2$, then for any $m>1$, $f(2,m)$ reduces to

$$2^{m-2}\pi^{-\frac{1}{2}}\frac{\Gamma(\frac{m}{2})\Gamma(\frac{m-1}{2})}{\Gamma(m-1)} \equiv 1 \tag{10}$$

which is known as a special case ($k=1$) of the identity $\Gamma(m-k) \equiv 2^{m-k-1}\pi^{-1/2}\Gamma(\frac{m-k}{2})\Gamma(\frac{m-k+1}{2})$ (Abramowitz and StegunAbramowitz and Stegun (1965), p. 483) . Assume $f(T,m)=1$ holds for $T=t$. It must then be shown that $f(t+1,m)=1$, $\forall m > T-1$, where

$$\begin{aligned} f(t+1,m) &= \prod_{k=1}^{t}[2^{(m-k-1)(t+1-k)} \times \frac{\Gamma(\frac{m}{2})}{\pi^{(t+1)/4}}\frac{\Gamma^{2t-2k+1}(\frac{m-k}{2})}{\Gamma^{t+1-k}(m-k)}] \\ &= [2^{m-t-1}\frac{\Gamma(\frac{m}{2})}{\pi^{(t+1)/4}}\frac{\Gamma(\frac{m-t}{2})}{\Gamma(m-t)}] \times \prod_{k=1}^{t-1}[2^{(m-k-1)(t+1-k)} \times \frac{\Gamma(\frac{m}{2})}{\pi^{(t+1)/4}}\frac{\Gamma^{2t-2k+1}(\frac{m-k}{2})}{\Gamma^{t+1-k}(m-k)}] \\ &= [2^{m-t-1}\frac{\Gamma(\frac{m}{2})}{\pi^{(t+1)/4}}\frac{\Gamma(\frac{m-t}{2})}{\Gamma(m-t)}] \times f(t,m) \times \prod_{k=1}^{t-1}[2^{m-k-1}\frac{1}{\pi^{1/4}}\frac{\Gamma^2(\frac{m-k}{2})}{\Gamma(m-k)}] \end{aligned} \tag{11}$$

Using the same identity $\Gamma(m-k) \equiv 2^{m-k-1}\pi^{-1/2}\Gamma(\frac{m-k}{2})\Gamma(\frac{m-k+1}{2})$ multiple times, $f(t+1,m)$

is

$$\begin{aligned} f(t+1,m) &= [\frac{1}{\pi^{(t-1)/4}}\frac{\Gamma(\frac{m}{2})}{\Gamma(\frac{m-t+1}{2})}] \times 1 \times \prod_{k=1}^{t-1}[\frac{1}{\pi^{-1/4}}\frac{\Gamma(\frac{m-k}{2})}{\Gamma(\frac{m-k+1}{2})}] \\ &= \prod_{k=1}^{t}[\frac{\Gamma(\frac{m-k+1}{2})}{\Gamma(\frac{m-k+1}{2})}] \\ &= 1 \end{aligned} \tag{12}$$

(c). According to EatonEaton (1983) (p. 256), if $\mathbf{\Sigma} \sim \mathrm{W}_T(m, \mathbf{\Psi})$, then any $T_1 \times T_1$ principal sub-matrix $\mathbf{\Sigma}_1$ of $\mathbf{\Sigma}$ has following distribution

$$\mathbf{\Sigma}_1 \sim W_{T_1}(m, \mathbf{\Psi}_1) \tag{13}$$

with $\mathbf{\Psi}_1$ being the $T_1 \times T_1$ principal sub-matrix of $\mathbf{\Psi}$. By applying it to the (2), any $T_1 \times T_1$ principal sub-matrix $\mathbf{R}_1$ of $\mathbf{R}$ has density function $b_{T_1}(m)|\mathbf{R}_1|^{\frac{m-T_1-1}{2}}$. Simply let T_1=2, we could obtain the marginal distribution of ρ_{ij}

$$f(\rho_{ij}) = \frac{\Gamma^2(\frac{m}{2})}{\Gamma_2(\frac{m}{2})}(1-\rho_{ij}^2)^{\frac{m-3}{2}}, \quad -1 \leq \rho_{ij} \leq 1 \tag{14}$$

□

B Stan code for AB-NMA models

B.1 AB-IW model

```
data{
  int<lower=0> len; //total number of data size
  int<lower=0> ntrt; //total number of treatments
  int<lower=0> nstudy; //total number of studies
  int<lower=0> t[len]; //indices of treatment
```

```
  int<lower=0> s[len]; //indices of study
  int<lower=0> r[len]; //number of events
  int<lower=0> totaln[len]; //number of participants
  int<lower=0,upper=1> higher_better; //to indicate whether higher absolute risk is better or not
  vector<lower=0>[ntrt] zeros; //mean effects in MVN
  matrix[ntrt,ntrt] Lambda; //parameter lambda in Inverse-Wishart, generally indentity matrix
  real<lower=1> nu; //parameter nu in Inverse-Wishart, generally ntrt+1
}
parameters {
  matrix[nstudy,ntrt] vi; //random effects
  vector[ntrt] mu; //fixed effect for treatment
  cov_matrix[ntrt] Sigma; //covariance matrix
}
transformed parameters {
  vector<lower=0,upper=1>[len] p; //Study_specified Absolute Risk
  for(i in 1:len){
    p[i]=inv_logit(mu[t[i]]+vi[s[i],t[i]]);
  }
}
model{
  for(i in 1:len){
    r[i]~binomial(totaln[i],p[i]);
  }
  for(j in 1:nstudy){
    vi[j,]~multi_normal(zeros,Sigma);
  }
  for(j in 1:ntrt){
    mu[j]~normal(0,100);
  }
```

```
  Sigma~inv_wishart(nu,Lambda);
}
generated quantities {
  real AR[ntrt]; //Populaiton averaged absolute risk
  real MU[ntrt];
  real SD[ntrt]; //standard deviation of covariance matrix
  real OR[ntrt,ntrt]; //Odds Ratio
  real LOR[ntrt,ntrt]; //Log odds Ratio
  real cLOR[ntrt,ntrt]; //cLog odds Ratio
  real CORR[ntrt,ntrt]; //correlation matrix
  int rk[ntrt]; // rank of AR
  int rank_prob[ntrt,ntrt]; //rank probability
  real DEV=0; // Total Residual Deviance
  for(j in 1:ntrt){
    AR[j]=inv_logit(mu[j]/sqrt(1+Sigma[j,j]*256/75/pi()/pi()));
    SD[j]=sqrt(Sigma[j,j]);
    MU[j]=mu[j];
  }
  for(j in 1:ntrt){
    for(k in 1:ntrt){
      OR[j,k]=AR[j]/(1-AR[j])/AR[k]*(1-AR[k]);
      LOR[j,k]=log(OR[j,k]);
      cLOR[j,k]=MU[j]-MU[k];
      CORR[j,k]=Sigma[j,k]/SD[j]/SD[k];
    }
  }
  for(j in 1:ntrt){
    rk[j]=higher_better?ntrt-rank(AR,j):rank(AR,j)+1;
  }
  for(j in 1:ntrt){
    for(k in 1:ntrt){
```

```
      rank_prob[k,j]=(rk[k]==j);
    }
  }
  for(i in 1:len){
    DEV=DEV+2*(r[i]*(-log(p[i]*totaln[i]))+(totaln[i]-r[i])*(-log(totaln[i]-p[i]*totaln[i])));
  }
}
```

B.2 AB-RW model

```
data{
  int<lower=0> len; //total number of data size
  int<lower=0> ntrt; //total number of treatments
  int<lower=0> nstudy; //total number of studies
  int<lower=0> t[len]; //indices of treatment
  int<lower=0> s[len]; //indices of study
  int<lower=0> r[len]; //number of events
  int<lower=0> totaln[len]; //number of participants
  real<lower=0> uni_upper; //uniform distribution upper bound
  int<lower=0,upper=1> higher_better; //to indicate whether higher absolute risk is better or not
  vector<lower=0>[ntrt] zeros; //mean effects in MVN
  real<lower=0.1> eta; //parameter eta in LKJ, generally 1
}
parameters {
  matrix[nstudy,ntrt] vi; //random effects
  vector[ntrt] mu; //fixed effect for treatment
  cholesky_factor_corr[ntrt] LOmega; //LKJ prior
  vector[ntrt] ssd; //seperate standard deviation
}
```

```
transformed parameters {
  vector<lower=0,upper=1>[len] p; //Study_specified Absolute Risk
  corr_matrix[ntrt] CORR; //correlation matrix
  for(i in 1:len){
    p[i]=inv_logit(mu[t[i]]+vi[s[i],t[i]]);
  }
  CORR <- multiply_lower_tri_self_transpose(LOmega);
}
model{
  for(i in 1:len){
    r[i]~binomial(totaln[i],p[i]);
  }
  for(j in 1:nstudy){
    vi[j,]~multi_normal(zeros, quad_form_diag(CORR,ssd));
  }
  for(j in 1:ntrt){
    mu[j]~normal(0,100);
    ssd[j]~uniform(0,uni_upper);
  }
  LOmega~lkj_corr_cholesky(eta);
}
generated quantities {
  real AR[ntrt]; //Populaiton averaged absolute risk
  real MU[ntrt];
  real SD[ntrt]; //standard deviation
  real OR[ntrt,ntrt]; //Odds Ratio
  real LOR[ntrt,ntrt]; //Log odds Ratio
  real cLOR[ntrt,ntrt]; //cLog odds Ratio
  int rk[ntrt]; // rank of AR
  int rank_prob[ntrt,ntrt]; //rank probability
  real DEV=0; // Total Residual Deviance
```

```
  for(j in 1:ntrt){
    AR[j]=inv_logit(mu[j]/sqrt(1+ssd[j]*ssd[j]*256/75/pi()/pi()));
    SD[j]=ssd[j];
    MU[j]=mu[j];
  }
  for(j in 1:ntrt){
    for(k in 1:ntrt){
      OR[j,k]=AR[j]/(1-AR[j])/AR[k]*(1-AR[k]);
      LOR[j,k]=log(OR[j,k]);
      cLOR[j,k]=MU[j]-MU[k];
    }
  }
  for(j in 1:ntrt){
    rk[j]=higher_better?ntrt-rank(AR,j):rank(AR,j)+1;
  }
  for(j in 1:ntrt){
    for(k in 1:ntrt){
      rank_prob[k,j]=(rk[k]==j);
    }
  }
  for(i in 1:len){
    DEV=DEV+2*(r[i]*(-log(p[i]*totaln[i]))+(totaln[i]-r[i])*(-log(totaln[i]-p[i]*totaln[i]))));
  }
}
```

B.3 AB-RIW model

```
data{
  int<lower=0> len; //total number of data size
  int<lower=0> ntrt; //total number of treatments
```

```
  int<lower=0> nstudy; //total number of studies
  int<lower=0> t[len]; //indices of treatment
  int<lower=0> s[len]; //indices of study
  int<lower=0> r[len]; //number of events
  int<lower=0> totaln[len]; //number of participants
  real<lower=0> uni_upper; //uniform distribution upper bound
  int<lower=0,upper=1> higher_better; //to indicate whether higher
   ↪ absolute risk is better or not
  vector<lower=0>[ntrt] zeros; //mean effects in MVN
  matrix[ntrt,ntrt] Lambda; //parameter lambda in Inverse-Wishart,
   ↪ generally indentity matrix
  real<lower=1> nu; //parameter nu in Inverse-Wishart, generally ntrt+1
}
parameters {
  matrix[nstudy,ntrt] vi; //random effects
  vector[ntrt] mu; //fixed effect for treatment
  cov_matrix[ntrt] Sigma; //covariance matrix
  vector[ntrt] ssd; //seperate standard deviation
}
transformed parameters {
  vector<lower=0,upper=1>[len] p; //Study_specified Absolute Risk
  corr_matrix[ntrt] CORR; //correlation matrix
  for(i in 1:len){
    p[i]=inv_logit(mu[t[i]]+vi[s[i],t[i]]);
  }
  for(j in 1:ntrt){
    for(k in 1:ntrt){
      CORR[j,k]=Sigma[j,k]/sqrt(Sigma[j,j])/sqrt(Sigma[k,k]);
    }
  }
}
```

```
model{
  for(i in 1:len){
    r[i]~binomial(totaln[i],p[i]);
  }
  for(j in 1:nstudy){
    vi[j,]~multi_normal(zeros, quad_form_diag(CORR,ssd));
  }
  for(j in 1:ntrt){
    mu[j]~normal(0,100);
    ssd[j]~uniform(0,uni_upper);
  }
  Sigma~inv_wishart(nu,Lambda);
}
generated quantities {
  real AR[ntrt]; //Populaiton averaged absolute risk
  real MU[ntrt];
  real SD[ntrt]; //standard deviation
  real OR[ntrt,ntrt]; //Odds Ratio
  real LOR[ntrt,ntrt]; //Log odds Ratio
  real cLOR[ntrt,ntrt]; //cLog odds Ratio
  int rk[ntrt]; // rank of AR
  int rank_prob[ntrt,ntrt]; //rank probability
  real DEV=0; // Total Residual Deviance
  for(j in 1:ntrt){
    AR[j]=inv_logit(mu[j]/sqrt(1+ssd[j]*ssd[j]*256/75/pi()/pi()));
    SD[j]=ssd[j];
    MU[j]=mu[j];
  }
  for(j in 1:ntrt){
    for(k in 1:ntrt){
      OR[j,k]=AR[j]/(1-AR[j])/AR[k]*(1-AR[k]);
```

```
      cLOR[j,k]=MU[j]-MU[k];
      LOR[j,k]=log(OR[j,k]);
    }
  }
  for(j in 1:ntrt){
    rk[j]=higher_better?ntrt-rank(AR,j):rank(AR,j)+1;
  }
  for(j in 1:ntrt){
    for(k in 1:ntrt){
      rank_prob[k,j]=(rk[k]==j);
    }
  }
  for(i in 1:len){
    DEV=DEV+2*(r[i]*(-log(p[i]*totaln[i]))+(totaln[i]-r[i])*(-log(totaln[i]-p[i]*totaln[i])));
  }
}
```

B.4 AB-EQ model

```
data{
  int<lower=0> len; //total number of data size
  int<lower=0> ntrt; //total number of treatments
  int<lower=0> nstudy; //total number of studies
  int<lower=0> t[len]; //indices of treatment
  int<lower=0> s[len]; //indices of study
  int<lower=0> r[len]; //number of events
  int<lower=0> totaln[len]; //number of participants
  real<lower=0> uni_upper; //uniform distribution upper bound
  int<lower=0,upper=1> higher_better; //to indicate whether higher absolute risk is better or not
```

```
  vector<lower=0>[ntrt] zeros; //mean effects in MVN
}
parameters {
  matrix[nstudy,ntrt] vi; //random effects
  vector[ntrt] mu; //fixed effect for treatment
  vector[ntrt] ssd; //seperate standard deviation
  real CORR; //correlation
}
transformed parameters {
  vector<lower=0,upper=1>[len] p; //Study_specified Absolute Risk
  matrix[ntrt,ntrt] Omega; //precision matrix
  real diag;
  real offdiag;
  diag=(1+(ntrt-2)*CORR)/(1+(ntrt-2)*CORR-(ntrt-1)*CORR^2);
  offdiag=(-CORR/(1+(ntrt-2)*CORR-(ntrt-1)*CORR^2));
  for(i in 1:len){
    p[i]=inv_logit(mu[t[i]]+vi[s[i],t[i]]);
  }
  for(j in 1:ntrt){
    for(k in 1:ntrt){
      Omega[j,k]=(j==k?diag:offdiag)/ssd[j]/ssd[k];
    }
  }
}
model{
  for(i in 1:len){
    r[i]~binomial(totaln[i],p[i]);
  }
  for(j in 1:nstudy){
    vi[j,]~multi_normal_prec(zeros, Omega);
  }
```

```
  for(j in 1:ntrt){
    mu[j]~normal(0,100);
    ssd[j]~uniform(0,uni_upper);
  }
  CORR~uniform((-1.000)/(ntrt-1)+0.0001,0.9999);
}
generated quantities {
  real AR[ntrt]; //Populaiton averaged absolute risk
  real MU[ntrt];
  real SD[ntrt]; //standard deviation
  real OR[ntrt,ntrt]; //Odds Ratio
  real LOR[ntrt,ntrt]; //Log odds Ratio
  real cLOR[ntrt,ntrt]; //cLog odds Ratio
  int rk[ntrt]; // rank of AR
  int rank_prob[ntrt,ntrt]; //rank probability
  real DEV=0; // Total Residual Deviance
  for(j in 1:ntrt){
    AR[j]=inv_logit(mu[j]/sqrt(1+ssd[j]*ssd[j]*256/75/pi()/pi()));
    SD[j]=ssd[j];
    MU[j]=mu[j];
  }
  for(j in 1:ntrt){
    for(k in 1:ntrt){
      OR[j,k]=AR[j]/(1-AR[j])/AR[k]*(1-AR[k]);
      cLOR[j,k]=MU[j]-MU[k];
      LOR[j,k]=log(OR[j,k]);
    }
  }
  for(j in 1:ntrt){
    rk[j]=higher_better?ntrt-rank(AR,j):rank(AR,j)+1;
  }
```

```
  for(j in 1:ntrt){
    for(k in 1:ntrt){
      rank_prob[k,j]=(rk[k]==j);
    }
  }
  for(i in 1:len){
    DEV=DEV+2*(r[i]*(-log(p[i]*totaln[i]))+(totaln[i]-r[i])*(-log(totaln[i
   ↪ ]-p[i]*totaln[i])));
  }
}
```

B.5 AB-EQ-EV model

```
data{
  int<lower=0> len; //total number of data size
  int<lower=0> ntrt; //total number of treatments
  int<lower=0> nstudy; //total number of studies
  int<lower=0> t[len]; //indices of treatment
  int<lower=0> s[len]; //indices of study
  int<lower=0> r[len]; //number of events
  int<lower=0> totaln[len]; //number of participants
  real<lower=0> uni_upper; //uniform distribution upper bound
  int<lower=0,upper=1> higher_better; //to indicate whether higher
   ↪ absolute risk is better or not
  vector<lower=0>[ntrt] zeros; //mean effects in MVN
}
parameters {
  matrix[nstudy,ntrt] vi; //random effects
  vector[ntrt] mu; //fixed effect for treatment
  real ssdd; //seperate standard deviation
  real CORR; //correlation
```

```
}
transformed parameters {
  vector<lower=0,upper=1>[len] p; //Study_specified Absolute Risk
  matrix[ntrt,ntrt] Omega; //precision matrix
  real diag;
  real offdiag;
  diag=(1+(ntrt-2)*CORR)/(1+(ntrt-2)*CORR-(ntrt-1)*CORR^2);
  offdiag=(-CORR/(1+(ntrt-2)*CORR-(ntrt-1)*CORR^2));
  for(i in 1:len){
    p[i]=inv_logit(mu[t[i]]+vi[s[i],t[i]]);
  }
  for(j in 1:ntrt){
    for(k in 1:ntrt){
      Omega[j,k]=(j==k?diag:offdiag)/ssdd/ssdd;
    }
  }
}
model{
  for(i in 1:len){
    r[i]~binomial(totaln[i],p[i]);
  }
  for(j in 1:nstudy){
    vi[j,]~multi_normal_prec(zeros, Omega);
  }
  for(j in 1:ntrt){
    mu[j]~normal(0,100);
  }
  ssdd~uniform(0,uni_upper);
  CORR~uniform((-1.000)/(ntrt-1)+0.0001,0.9999);
}
generated quantities {
```

```
real AR[ntrt]; //Populaiton averaged absolute risk
real MU[ntrt];
real SD; //standard deviation
real OR[ntrt,ntrt]; //Odds Ratio
real LOR[ntrt,ntrt]; //Log odds Ratio
real cLOR[ntrt,ntrt]; //cLog odds Ratio
int rk[ntrt]; // rank of AR
int rank_prob[ntrt,ntrt]; //rank probability
real DEV=0; // Total Residual Deviance
for(j in 1:ntrt){
  AR[j]=inv_logit(mu[j]/sqrt(1+ssdd*ssdd*256/75/pi()/pi()));
  MU[j]=mu[j];
}
SD=ssdd;
for(j in 1:ntrt){
  for(k in 1:ntrt){
    OR[j,k]=AR[j]/(1-AR[j])/AR[k]*(1-AR[k]);
    LOR[j,k]=log(OR[j,k]);
    cLOR[j,k]=MU[j]-MU[k];
  }
}
for(j in 1:ntrt){
  rk[j]=higher_better?ntrt-rank(AR,j):rank(AR,j)+1;
}
for(j in 1:ntrt){
  for(k in 1:ntrt){
    rank_prob[k,j]=(rk[k]==j);
  }
}
for(i in 1:len){
  DEV=DEV+2*(r[i]*(-log(p[i]*totaln[i]))+(totaln[i]-r[i])*(-log(totaln[i
```

```
   ↪ ]-p[i]*totaln[i])));
  }
}
```

C R Simulation codes for Table 1

```
logit <- function(x){
  return(log(x/(1-x)))
}
#########
library(rstan)
library(bayesplot)
library(coda)
library(mvtnorm)
library(randomizr)
######
job = 1
#####
d = 3
nstudy = 18
pop_size <- 500
set.seed(seed)
sd = c(0.7,0.4,0.1)
cor <- matrix(c(1,2/3,4/9,2/3,1,2/3,4/9,2/3,1),nrow=3,ncol=3)
sigma <- matrix(NA,nrow=3,ncol=3)
for(i in 1:d){
  for(j in 1:d){
    sigma[i,j] <- cor[i,j]*sd[i]*sd[j]
  }
}
v <- rmvnorm(nstudy, rep(0, length = 3), sigma)
```

```
rep.row<-function(x,n){
  matrix(rep(x,each=n),nrow=n)
}
mu <- rep.row(c(-1,-1.5,-2),nstudy)
pi1 <- exp(mu+v)/(1+exp(mu+v))
event <- matrix(NA,nrow=nstudy,ncol=3)
for(i in 1:d){
  event[,i] <- rbinom(nstudy,pop_size,pi1[,i])
}

missing_pattern_determine <- function(ind=1, event){
d = 3
nstudy = 18

# 1. No missing
if(ind==1){
  missing_ind <- rep(0,nstudy)
}

# 2. MAR
if(ind==2){
  missing_ind <- rep(0,nstudy)
  for(i in 1:nstudy){
    if(event[i,2]>=median(event[,2])){missing_ind[i] <- 3}
    else{missing_ind[i] <- 1}
  }
}

# 3. MNAR
beta_mar_0 <- 4.5
beta_mar_1 <- 1
```

```
if(ind==3){
  missing_ind <- rep(0,nstudy)
  eventm <- event
  for(i in 1:nstudy){
    if(event[i,2]==0){eventm[i,2]=0.5}
    if(event[i,2]==500){eventm[i,2]=499.5}
    if(event[i,1]==0){eventm[i,1]=0.5}
    if(event[i,1]==500){eventm[i,1]=499.5}
    if(event[i,3]==0){eventm[i,3]=0.5}
    if(event[i,3]==500){eventm[i,3]=499.5}
    tempmisind <- exp(beta_mar_0+beta_mar_1*logit(eventm[i,2]/500)+beta_
   ↪ mar_1*logit(eventm[i,1]/500)+beta_mar_1*logit(eventm[i,3]/500))/(1+
   ↪ exp(beta_mar_0+beta_mar_1*logit(eventm[i,2]/500)+beta_mar_1*logit(
   ↪ eventm[i,1]/500)+beta_mar_1*logit(eventm[i,3]/500)))
    if(rbinom(1,1,tempmisind)==1){missing_ind[i] <- 1}
    else{missing_ind[i] <- 3}
  }
}

### Construct the non-missing pattern
non_missing_pattern <- matrix(1,nrow=nstudy,ncol=d)
for(i in 1:nstudy){
  if(missing_ind[i]!=0){
    non_missing_pattern[i,missing_ind[i]] <- 0
  }
}
return(non_missing_pattern)
}

###### statistical analyze part
for(missing_ii in 1:3){
```

```
#missing_ii <- 3
set.seed(seed*missing_ii)
# missing pattern
non_missing_pattern <- missing_pattern_determine(ind=missing_ii,event)
# set up network
len <- sum(sum(non_missing_pattern))
s.id <- rep(0,len)
t.id <- rep(0,len)
event.n <- rep(0,len)
total.n <- rep(pop_size,len)
ind <- 1
for(i in 1:nstudy){
  for(j in 1:d){
    if(non_missing_pattern[i,j]!=0){
      s.id[ind] <- i
      t.id[ind] <- j
      event.n[ind] <- event[i,j]
      ind <- ind+1
    }
  }
}
ntrt <- d
r <- event.n
n <- total.n


#### AB-IW
data.stan <- list(len=len,ntrt=ntrt,nstudy=nstudy,t=t.id,s=s.id,r=event.
 ↪ n,totaln=total.n, zeros=rep(0,ntrt),Lambda=diag(ntrt),nu=ntrt+1,
 ↪ higher_better=1,uni_upper=5)
init.stan<- function() { list(mu=rep(0, ntrt),vi=matrix(0,nstudy,ntrt),
```

```
 ↪ Sigma=diag(ntrt))}
fit <- sampling(IW,pars=c("MU","LOR","cLOR","AR","SD","CORR","rank_prob"
 ↪ ,"DEV","p"),data=data.stan,init=init.stan,chains=4,iter=2000,warmup
 ↪ =1000,verbose=TRUE,seed=(1234+job),thin=1,control = list(max_
 ↪ treedepth = 15))
sum_fit <- summary(fit)$summary
saveRDS(sum_fit,paste("IW_",missing_ii,"_",job,sep=""))

#### AB-RW
data.stan <- list(len=len,ntrt=ntrt,nstudy=nstudy,t=t.id,s=s.id,r=event.
 ↪ n,totaln=total.n, zeros=rep(0,ntrt),eta=1,higher_better=1,uni_upper
 ↪ =5)
init.stan<- function() { list(mu=rep(0, ntrt),vi=matrix(0,nstudy,ntrt),
 ↪ LOmega=diag(ntrt),ssd=runif(ntrt,0,1)) }
fit <- sampling(RW,pars=c("MU","LOR","cLOR","AR","SD","CORR","rank_prob"
 ↪ ,"DEV","p"),data=data.stan,init=init.stan, chains=4,iter=2000,warmup
 ↪ =1000,verbose=TRUE,seed=(1234+job),thin=1,control = list(adapt_delta
 ↪  = 0.99,max_treedepth = 15))
sum_fit <- summary(fit)$summary
saveRDS(sum_fit,paste("RW_",missing_ii,"_",job,sep=""))



#### AB-RIW
data.stan <- list(len=len,ntrt=ntrt,nstudy=nstudy,t=t.id,s=s.id,r=event.
 ↪ n,totaln=total.n, zeros=rep(0,ntrt),Lambda=diag(ntrt),nu=ntrt+1,
 ↪ higher_better=1,uni_upper=5)
init.stan<- function() { list(mu=rep(0, ntrt),vi=matrix(0,nstudy,ntrt),
 ↪ Sigma=diag(ntrt),ssd=runif(ntrt,0,1)) }
fit <- sampling(RIW,pars=c("MU","LOR","cLOR","AR","SD","CORR","rank_prob
 ↪ ","DEV","p"),data=data.stan,init=init.stan, chains=4,iter=2000,
 ↪ warmup=1000,verbose=TRUE,seed=(1234+job),thin=1,control = list(adapt
```

```
_delta = 0.99,max_treedepth = 15))
  sum_fit <- summary(fit)$summary
  saveRDS(sum_fit,paste("RIW_",missing_ii,"_",job,sep=""))


  #### AB-EQ
  data.stan <- list(len=len,ntrt=ntrt,nstudy=nstudy,t=t.id,s=s.id,r=event.n,totaln=total.n, zeros=rep(0,ntrt),higher_better=1,uni_upper=5)
  init.stan<- function() { list(mu=rep(0, ntrt),vi=matrix(0,nstudy,ntrt), CORR=0,ssd=runif(ntrt,0,1)) }
  fit <- sampling(EQ,pars=c("MU","LOR","cLOR","AR","SD","CORR","rank_prob","DEV","p"),data=data.stan,init=init.stan, chains=4,iter=2000,warmup=1000,verbose=TRUE,seed=(1234+job),thin=1,control = list(adapt_delta = 0.99,max_treedepth = 15))
  sum_fit <- summary(fit)$summary
  saveRDS(sum_fit,paste("EQ_",missing_ii,"_",job,sep=""))
}
```

References

Abramowitz, M. and I. A. Stegun (Eds.) (1965). *Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables (Dover Books on Mathematics)*. Dover Publications.

Eaton, M. L. (1983). *Multivariate Statistics: A Vector Space Approach (Wiley Series in Probability and Statistics)*. John Wiley & Sons Inc.

James, A. T. (1964). Distributions of matrix variates and latent roots derived from normal samples. *The Annals of Mathematical Statistics 35*(2), 475–501.

Lewandowski, D., D. Kurowicka, and H. Joe (2009). Generating random correlation matrices based on vines and extended onion method. *Journal of Multivariate Analysis 100*(9), 1989–2001.