

日本製薬工業協会 データサイエンス部会シンポジウム
「ネットワークメタアナリシスを基礎から学ぶ:概説・事例から留意事項まで」

NMAのデモンストレーション



2019年11月28日

渥美 淳

東レ株式会社

日本製薬工業協会 医薬品評価委員会データサイエンス部会 継続課題TF5

1. NMAを実施するソフトウェア等の紹介
2. ベイズ流NMAのデモンストレーション
 - 先の発表「HTAにおけるNMA及び異質性, 不一致性の評価」で用いられた例の実施
 - パーキンソン病の補助療法の例
 - ✓ 固定効果モデル・変量効果モデル
 - ✓ Deviance plot, Ranking probabilities, SUCRA,...
 - 急性心筋梗塞に対する血栓溶解薬の例
 - ✓ 大規模試験が存在しなかった場合の結果

はじめに

- 本発表で扱うネットワークメタアナリシス(NMA)は、要約データ(Aggregated data)を用いたベイズ流NMAを中心に取り扱いっており、被験者レベルのデータ(Individual Participant/Patient Data : IPD)を用いたNMA(IPD-NMA)は範囲外となっております。

1. NMAを実施するソフトウェア等の紹介

NMAを実施するソフトの分類

- MCMC専用ソフト
 - 自由にモデリング可能
 - 論文等でBUGSコードが公開されている場合が多い
 - 汎用の統計解析ソフトに比べてモデルを記述しやすい
 - データハンドリングや図示のために他と組み合わせて使う場合が多い
- 汎用の統計解析ソフト
 - 汎用機能により自分でモデルを作成する
 - NMA用の機能/アドインを用いる
 - NMA用の図示機能
 - 組み込まれているNMAモデルのみ利用可能
- NMA専用のソフト
 - GUI対応, 図示機能
 - 組み込まれているNMAモデルのみ利用可能
 - 実際の解析はMCMC専用ソフトを利用する場合が多い

➤ WinBUGS; Bayesian inference Using Gibbs Sampling

- マルコフ連鎖モンテカルロ (Markov chain Monte Carlo; MCMC) 法による解析を行うソフトウェア

- <https://www.mrc-bsu.cam.ac.uk/software/bugs/the-bugs-project-winbugs/>

- 参考: 日本製薬工業協会 医薬品評価委員会 DS部会. WinBUGS の使い方. 2014

- **v1.4.3で開発終了 (Win10でも動いてはいるように見えるものの…)**

- Windows10ではインストーラー版(exeファイル)ではなくzipファイル版を使うこと!

- NICE DSU TSD等にNMAの豊富な事例

- 方法論の研究論文やシステマティックレビューにコードがついている場合も

- 代替ソフトウェア(BUGS言語): OpenBUGS, JAGS

➤ Stan

- NUTS(No-U-turn sampler)アルゴリズムに基づいたハミルトニアン・モンテカルロ法(HMC)の採用
- WinBUGSに比べてデバッグしやすい

➤ R

- 汎用MCMCパッケージ: MCMCpack
- MCMCに外部ソフト利用: Stan(はRstan, WinBUGS(JAGS, OpenBUGS)はR2WinBUGS(rjags/R2jags/runjags, BRugs)
- NMA用パッケージ
 - netmeta: Network Meta-Analysis using Frequentist Methods
 - Pcneta: Performs arm-based network meta-analysis for datasets with binary, continuous, and count outcomes using the Bayesian methods
 - gemtc: Network meta-analyses (mixed treatment comparisons) in the Bayesian framework using JAGS.
 - nmaINLA: Network Meta-Analysis using Integrated Nested Laplace Approximations
 - BUGSnet: an R package to facilitate the conduct and reporting of Bayesian network Meta-analyses

汎用の統計解析ソフト:代表例とNMAのための機能

➤ SAS

■ 汎用MCMC: MCMC procedure

□ SAS/STAT 14.1からはサンプリング法にHMC, NUTSが追加

□ 外部ソフトも利用可能.

✓ WinBUGS利用は製薬協のWinBUGS の使い方で提供されている

■ 一般化線形混合モデル (glimmix procedure) (Madden, L. V., 2016)

➤ Stata

■ NMA用機能: mvmeta (Frequentist Methods)

□ <http://www.mrc-bsu.cam.ac.uk/software/stata-software>

➤ Python

■ 汎用MCMC: PyMC, PyStan (Stan利用), PyJAGS (JAGS利用)

過去のSASユーザー総会でのNMA関連発表

- 福井 伸行, 乙黒 俊也, 磯崎 充宏. ネットワークメタアナリシスによる無作為化比較試験の統合. SASユーザー総会2014
- 舟尾暢男, 黒田晋吾. SAS でのNetwork Meta-Analysisの実施例 ~頻度論に基づくアプローチ~. SASユーザー総会2016
- 奥山ことば, 町田 光陽, 石渡 量太, 渥美 淳, 奥村 広之. NICE TSD Evidence Synthesisシリーズの紹介: ネットワークメタアナリシスを中心として. SASユーザー総会2019

NMA専用のソフト: GeMTC・ADDIS

NMA実行ソフトおよびRパッケージ

➤ Rパッケージ(開発停止)

- NICE TSD, ISPORレポートで検討されたモデルを利用(BUGS利用)

- <http://www.bristol.ac.uk/population-health-sciences/centres/cresyda/mpes/mtc/>

- NICE TSD 著者らのNMAの本でも紹介されている

- V0.7からは

- JAGS のみ対応(RJGAS利用)、WinBUGS とOpenBUGSは非対応

- residual deviance statistics追加とともにDIC算出方法はNICE TSD流へ変更

➤ GUI 版(開発停止)

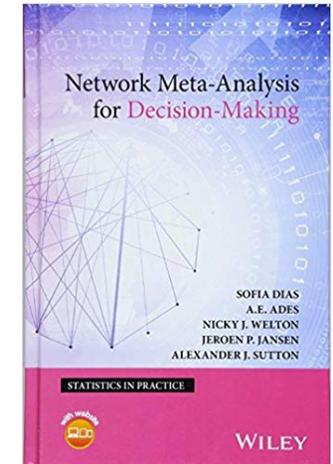
- 非Rパッケージの独立したソフトウェア

- <https://drugis.org/software/addis1/gemtc>

➤ 開発はADDIS 2へ移行

- web上で実施する形式

- <https://drugis.org/software/addis2/index>



NMA専用のソフト: TIDI

- 費用効果分析にNMAを組み込んだ統合ソフト (Bujkiewicz, S., 2011)
 - Transparent Interactive Decision Interrogator
- NICE TSD6で紹介されている
- ExcelベースでR、WinBUGSを操作する…動作確認できず
 - Windows10やoffice 365に一部未対応?
- Sylwia Bujkiewicz Intro to TIDI(University of LEICESTER)

<https://www2.le.ac.uk/departments/health-sciences/research/biostats/staff-pages/sb309>

NMA専用のソフト: NetMetaXL

- Excelベース(VBA)のGUIで簡単に試せる
 - WinBUGS利用 (WinBUGSスクリプトやlogは出力される)
 - <http://www.NetMetaXL.com>
- CADTHが協力
 - Canadian Agency for Drugs and Technologies in Health
カナダのHTA医療技術評価機関
 - NICE TSD, ISPORレポートで検討されたモデルを利用
- 生成可能な図表: evidence network diagrams, forest plots, league tables of pairwise comparisons, probability plots (rankograms), and inconsistency plots
- Limitation: v1-6-1
 - 解析データセット: 15 treatment options and 50 studies
 - Outcome: 2値
- How to...YouTube Video on the CADTH channel: <https://www.youtube.com/user/CADTHACMTS>

➤ 将来的に使えなくなる可能性

- 特にMicrosoft office製品と連携している場合は注意
- オープンソースソフトウェアの利用も視野へ
 - R, OpenBUGS, JAGS, stan

➤ 費用効果分析に用いることに関してNICE TSD6には…

- 確率的感度分析としてモンテカルロシミュレーションベースのアプローチを推奨
 - MCMCでは、事後分布が意思決定の分析に直接用いられる
- エビデンス統合では複数のパラメータを推定し、そのほとんどの場合でパラメータ間に相関があるため、パラメータの複雑な不確実性の構造が決定モデルに伝播するようなソフトウェアを用いるべき

ソフトウェアの参考文献

- NICE TSDs
URL: <http://nicesu.org.uk/technical-support-documents/technical-support-documents/>
 - NICE TSDs 1～7がエビデンスの統合, WinBUGSによる例が豊富
- Dias S, Ades AE, Welton NJ, et. al. (2018) Network Meta-Analysis for Decision-Making. Wiley.
- Madden, L. V., Piepho, H. P., & Paul, P. A. (2016). Statistical models and methods for network meta-analysis. *Phytopathology*, 106(8), 792-806.
- van Valkenhoef, G., Lu, G., de Brock, B., Hillege, H., Ades, A. E., & Welton, N. J. (2012). Automating network meta-analysis. *Research synthesis methods*, 3(4), 285-299.
- R gemtcパッケージ URL: <http://cran.r-project.org/web/packages/gemtc/>
- Bujkiewicz, S., Jones, H., Lai, M., Cooper, N., Hawkins, N., Abrams, K., Spiegelhalter, D., Sutton, A. (2011). Introduction to Transparent Interactive Decision Interrogator (TIDI)..
- Brown, S., Hutton, B., Clifford, T., Coyle, D., Grima, D., Wells, G., & Cameron, C. (2014). A Microsoft-Excel-based tool for running and critically appraising network meta-analyses—an overview and application of NetMetaXL. *Systematic reviews*, 3(1), 110.

ソフトウェア調査報告の文献:

- Neupane, B., Richer, D., Bonner, A. J., Kibret, T., & Beyene, J. (2014). Network meta-analysis using R: a review of currently available automated packages. *PloS one*, 9(12), e115065.
- Xu C., Niu Y., Wu J., Gu H., Zhang C. (2018). Software and package applying for network meta-analysis: A usage-based comparative study. *J Evid Based Med*. Aug;11(3):176-183.

2. ベイズ流NMAのデモンストレーション

➤ PC Spec

- CPU: i3-8130U @ 2.20GHz
- RAM: 8GB

➤ System 日本語環境

- Windows 10 Enterprise (x64)
- R x64 v3.5.3, RStudio Version 1.2.1335
 - ※v3.6.x以降は乱数の仕様変更により, 結果が変わる可能性あり.
 - Rパッケージgemtc 0.8-2
 - ✓ NMAモデル:一般化線形モデルの枠組み
 - ✓ NICE TSDで検討されたモデルが多数組み込まれている(全てではない)
- JAGS v4.3.0, WinBUGS v1.4.3

➤ GeMTC → JAGSへ渡す seedは工夫が必要

- Rの乱数シード `set.seed(XXXX e.g. 4989)`は初期値の自動設定用
- 結果の再現性のためにはJAGS用のコードを追加する必要がある
作者より

The seeds for the JAGS RNGs and the R RNG are independent, so both need to be set:

- Use `set.seed()` prior to running `mtc.model()` to control the generation of random initial values.
- Set `.RNG.name` and `.RNG.seed` in `model$inits` to control the JAGS RNGs.

- Seedによらず同様な結果となることが望ましい。

パーキンソン病の補助療法:ドーパミンアゴニスト (治療法2~5)のNMA

➤ NMAの目的

- パーキンソン病での補助療法としてドーパミンアゴニスト(治療薬2~5)が処方された患者における, Off-time時間(薬剤の効果が認められない時間)の減少を評価する

➤ 試験データ

- 7つの臨床試験について, 各治療の例数, Off-time(薬剤の効果が認められない時間)の減少の平均, その標準偏差が与えられている

試験	治療法	例数	Off-timeの減少の平均	SD	群間差
1	1	54	-1.22	3.7	
	3	95	-1.53	4.28	-0.31
2	1	172	-0.7	3.7	
	2	173	-2.4	3.4	-1.7
3	1	76	-0.3	4.4	
	2	71	-2.6	4.3	-2.3
4	4	81	-1.2	4.3	-0.9
	3	128	-0.24	3	
5	4	72	-0.59	3	-0.35
	3	80	-0.73	3	
6	4	46	-0.18	3	0.55
	4	137	-2.2	2.31	
7	5	131	-2.5	2.18	-0.3
	4	154	-1.8	2.48	
7	5	143	-2.1	2.99	-0.3

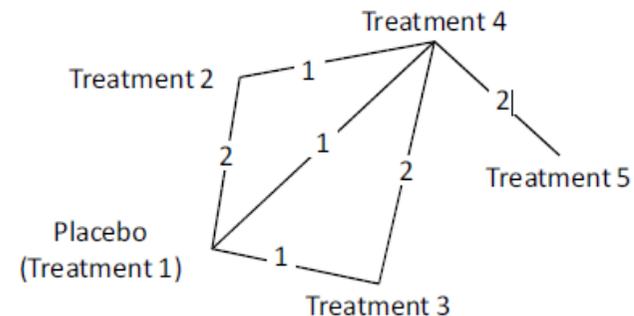


図. パーキンソン治療のネットワーク図
先端は治療(treatment1~5), 連結線は試験内で直接比較された治療のペア. 線上の数値は試験数.

[出典:NICE DSU TSD2]

パーキンソン病の補助療法の例:ネットワーク図, 変量効果モデル実行

```
> library(gemtc)
> Sys.setlocale(locale="C")
> data<-read.csv("parkinson.txt")
> network <- mtc.network(data)
> summary(network)
> plot(network)
> n.chain<-3
> set.seed(1234)
> model <- mtc.model(network,linearModel="random",n.chain = n.chain)
> for (i in 1:n.chain) {
+   model$inits[[i]]$.RNG.name = "base::Mersenne-Twister"
+   model$inits[[i]]$.RNG.seed = 123 + i
+   print(model$inits[i])
+ }
```

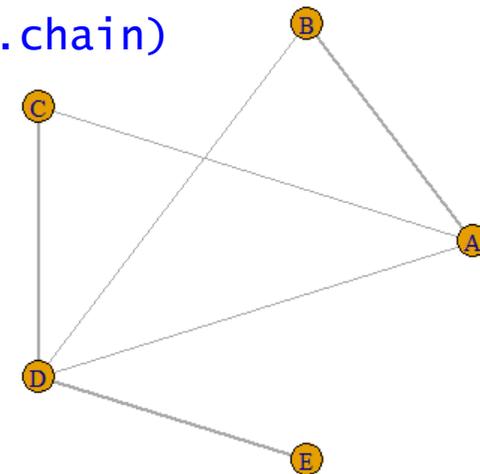
JAGSへ渡すseedを指定するための追加コード例

#変量効果モデルの実行

> #変量効果モデルの実行

```
> result.re <- mtc.run(model, n.adapt=50000, n.iter=100000,thin=1)
```

変数名はパッケージの規定に従うこと. 詳細はhelp参照
parkinson.txt ↓
"", "study", "treatment", "mean", "std.dev", "sampleSize"
"1", "1", "A", -1.22, 3.7, 54
"2", "1", "C", -1.53, 4.28, 95
... ..



変量効果モデルの結果

```
> summary(result.re)
```

```
Results on the Mean Difference scale
```

```
Iterations = 50001:150000  
Thinning interval = 1  
Number of chains = 3  
Sample size per chain = 1e+05
```

```
1. Empirical mean and standard deviation for each variable,  
plus standard error of the mean:
```

	Mean	SD	Naive SE	Time-series SE
d.D.A	0.53270	0.6151	0.0011230	0.005276
d.D.B	-1.31362	0.6855	0.0012515	0.005551
d.D.C	0.02582	0.4654	0.0008496	0.003173
d.D.E	-0.30060	0.4230	0.0007724	0.001544
sd.d	0.37538	0.3598	0.0006570	0.004272

```
2. Quantiles for each variable:
```

	2.5%	25%	50%	75%	97.5%
d.D.A	-0.66598	0.1577	0.52760	0.9075	1.7466
d.D.B	-2.67483	-1.7242	-1.30858	-0.8935	0.0126
d.D.C	-0.88287	-0.2483	0.02464	0.3022	0.9327
d.D.E	-1.14675	-0.5102	-0.30049	-0.0922	0.5482
sd.d	0.01211	0.1236	0.27078	0.5073	1.3959

```
-- Model fit (residual deviance):
```

Dbar	pD	DIC
13.62607	12.43443	26.06050

```
15 data points, ratio 0.9084, I2 = 0%
```

pairwise comparisons

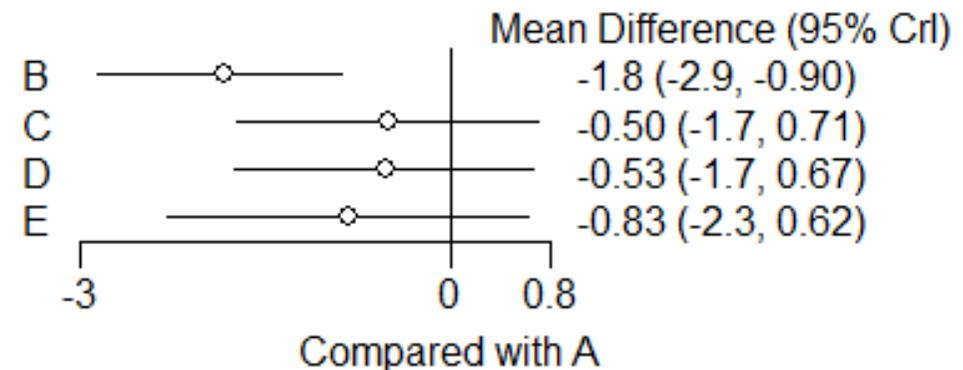
>(tbl=relative.effect.table(result.re))

Mean Difference (95% CrI)

A	-1.84 (-2.861, -0.8957)	-0.5025 (-1.743, 0.7096)	-0.5276 (-1.747, 0.666)	-0.8273 (-2.303, 0.6186)
1.84 (0.8957, 2.861)	B	1.338 (-0.08151, 2.784)	1.309 (-0.0126, 2.675)	1.009 (-0.5513, 2.614)
0.5025 (-0.7096, 1.743)	-1.338 (-2.784, 0.08151)	C	-0.02464 (-0.9327, 0.8829)	-0.3268 (-1.555, 0.9093)
0.5276 (-0.666, 1.747)	-1.309 (-2.675, 0.0126)	0.02464 (-0.8829, 0.9327)	D	-0.3005 (-1.147, 0.5482)
0.8273 (-0.6186, 2.303)	-1.009 (-2.614, 0.5513)	0.3268 (-0.9093, 1.555)	0.3005 (-0.5482, 1.147)	E

#A(プラセボ)を対照とした群間差のフォレストプロット

> forest(tbl, "A")



deviance plot, leverage plot

```
> deviance<-mtc.deviance(result.re)
> pdf("deviancepot.parkinson_re.pdf")
> plot(deviance)
> dev.off()
```

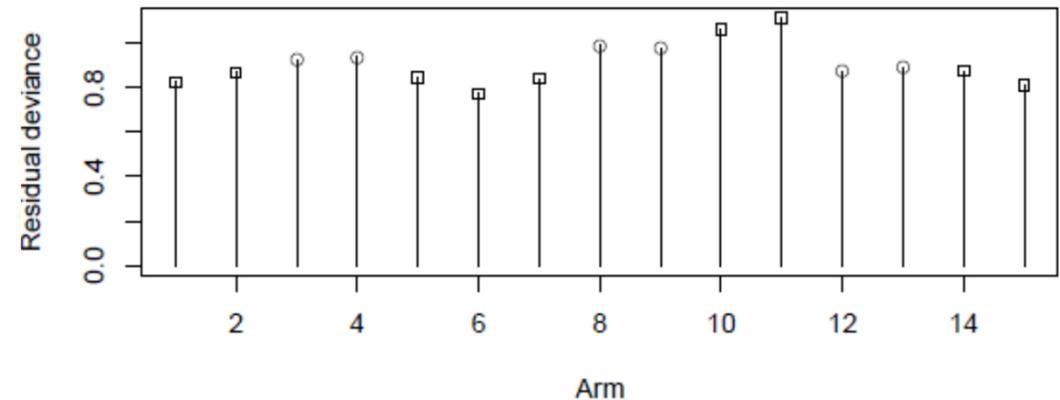
#MCMC診断プロット(スライドからは割愛)

```
> pdf("result.parkinson_re.pdf")
> plot(result.re)
> dev.off()
```

```
> pdf("autocorr.plot.parkinson_re.pdf")
> autocorr.plot(result.re$samples)
> crosscorr(result.re)
> dev.off()
```

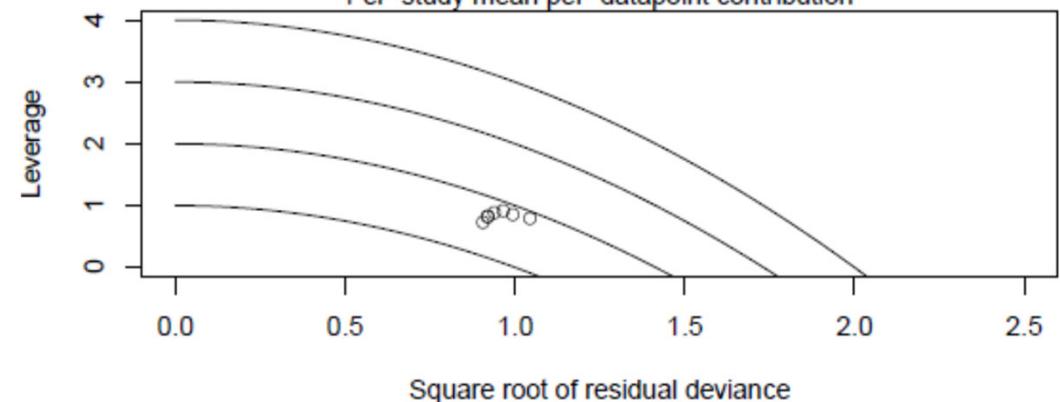
```
> gelman.diag(result.re) #<1.1なら収束と判断
```

Per-arm residual deviance



Leverage versus residual deviance

Per-study mean per-datapoint contribution



Ranking probabilities

#事後分布のサンプリングごとに治療効果を順位付け→順位の確率の推定値

```
> ranks <- rank.probability(result.re)
```

```
> round(ranks,2,preferredDirection=-1) #負がbetterと指定
```

```
Rank probability; preferred direction = -1
```

```
  [,1] [,2] [,3] [,4] [,5]  
A 0.00 0.06 0.08 0.14 0.73  
B 0.90 0.06 0.02 0.01 0.00  
C 0.02 0.21 0.25 0.39 0.13  
D 0.01 0.09 0.45 0.36 0.09  
E 0.07 0.58 0.20 0.10 0.05
```

```
> colnames(ranks)<-1:5
```

```
> plot(ranks,legend = colnames(ranks) ,col = rainbow(5))
```

#SUCRAはパッケージに機能がなため手動計算 (次スライドへ続く)

```
> library(reshape2)
```

```
> library(ggplot2)
```

```
> ranks.ma<-matrix(ranks, nrow=nrow(ranks),  
ncol=ncol(ranks))
```

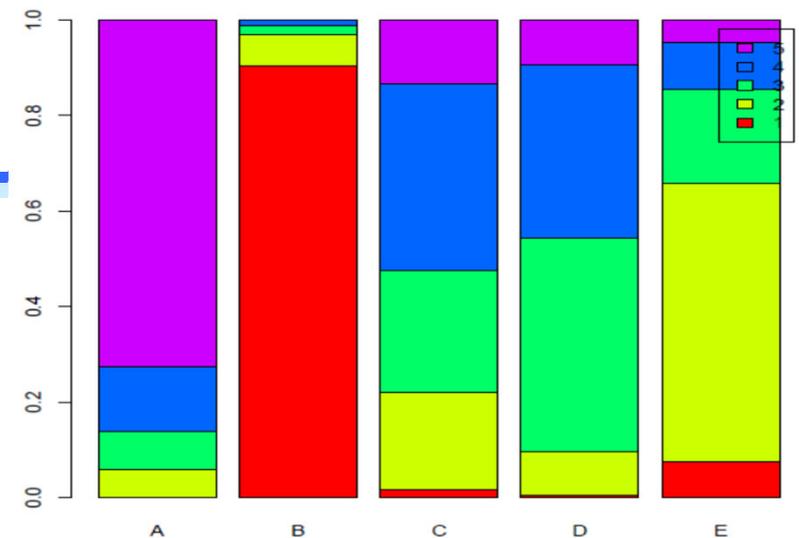
```
> row.names(ranks.ma)<-row.names(ranks)
```

```
> y<-melt(ranks.ma)#図のためのデータ整形
```

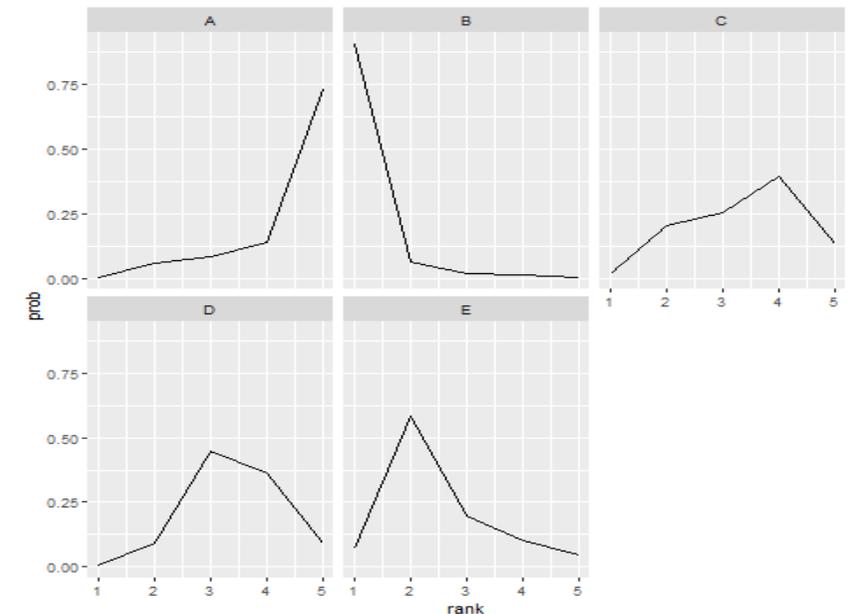
```
> colnames(y)<-c("treat","rank","prob")
```

```
> ggplot(y,aes(x = rank, y = prob))
```

```
+ geom_line()+facet_wrap(~treat)
```



薬協



SUCRA Surface Under the Cumulative RAnking curve

```
> #順位確率の累積和 パッケージに機能がないので手動計算
> y<-y[order(y$treat,y$rank),]
> y<-within(y, cum_prob <- ave(prob, treat, FUN=cumsum))
> y[1:6,]
```

	treat	rank	prob	cum_prob
1	A	1	0.0007366667	0.0007366667
6	A	2	0.0566800000	0.0574166667
11	A	3	0.0810800000	0.1384966667
16	A	4	0.1356800000	0.2741766667
21	A	5	0.7258233333	1.0000000000
2	B	1	0.9043300000	0.9043300000

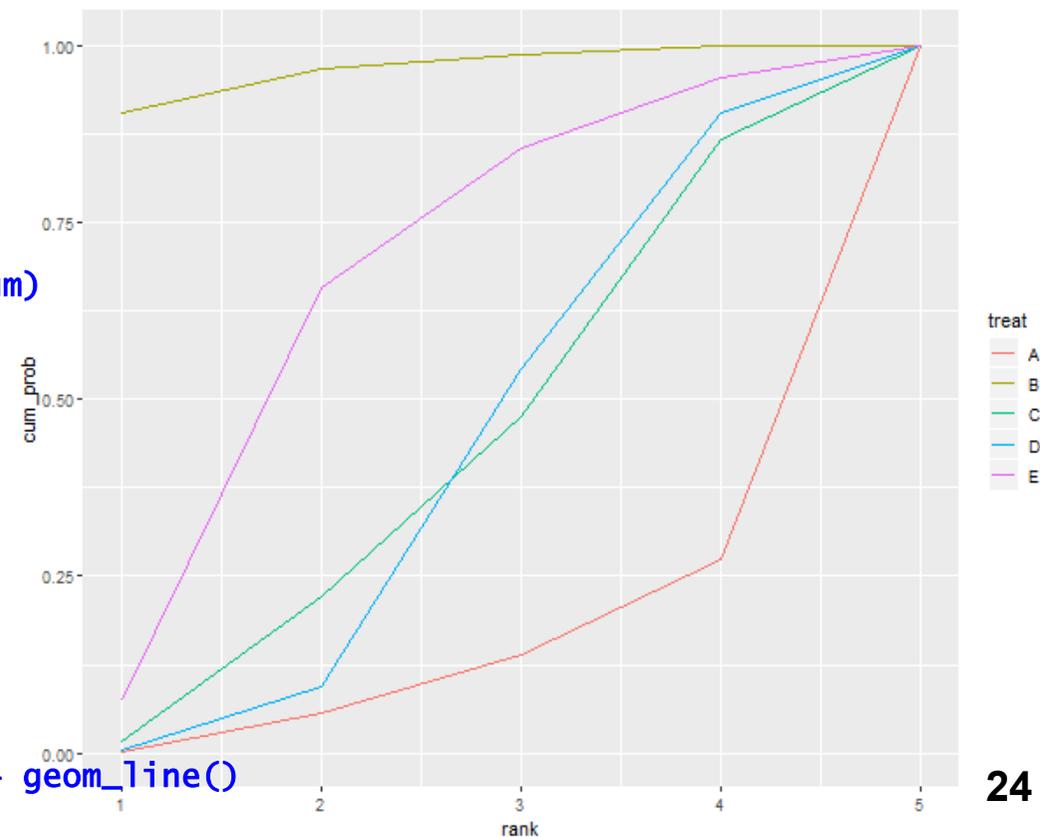
```
> #SUCRA
> sucra<-aggregate(y$cum_prob,by=list(treat=y$treat),sum)
> sucra$sucra<-(sucra$x-1)/(nrow(sucra)-1)
> sucra[c(1,3)]
```

treat	sucra
1 A	0.1177067
2 B	0.9648492
3 C	0.3950633
4 D	0.3873342
5 E	0.6350467

```
> ggplot(y,aes(x = rank, y = cum_prob, color = treat))+ geom_line()
```

$$SUCRA_i = \frac{\sum_{j=1}^{T-1} P_{i,j}}{T-1}$$

$P_{i,j}$: 治療iの順位jの確率の累積和, T: 治療数
順位jの確率の累積和(下図)の曲面下面積



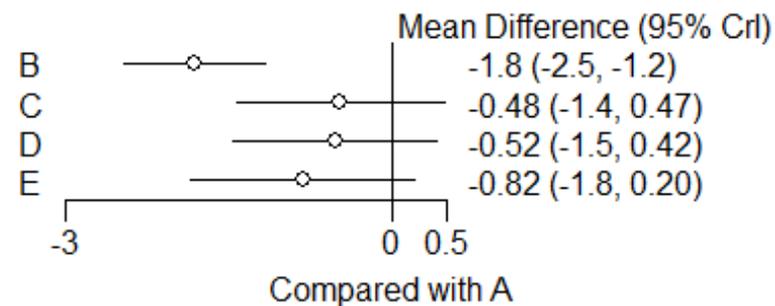
固定効果モデルの場合

```
> set.seed(1234)
> model <- mtc.model(network, linearModel="fixed", n.chain = n.chain)
> for (i in 1:n.chain) {
+   model$inits[[i]]$.RNG.name = "base::Mersenne-Twister"
+   model$inits[[i]]$.RNG.seed = 123+i
+   print(model$inits[i])
+ }
> result.fe <- mtc.run(model, n.adapt=50000, n.iter=100000, thin=1)
```

-- Model fit (residual deviance):

Dbar	pD	DIC
13.28124	10.99330	24.27454

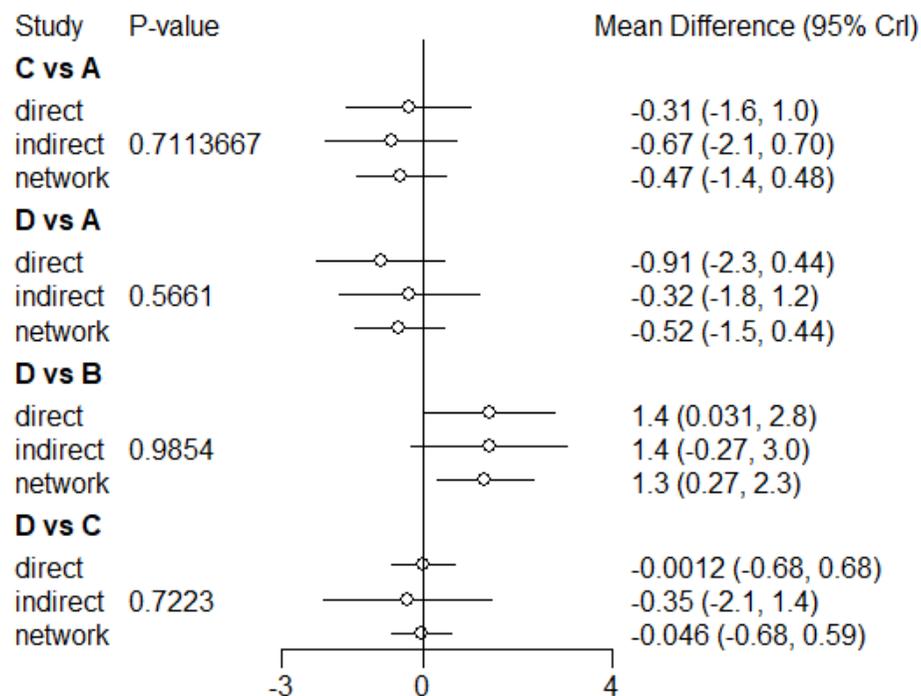
15 data points, ratio 0.8854, $\hat{I}^2 = 0\%$



Node-splitting法による不一致性の評価

➤ 固定効果モデルの結果

- > result.pd.ns <- mtc.nodesplit(network, linearModel='fixed', n.adapt=10000, n.iter=20000, thin=1, n.chain = 3)
- > summary.pd.ns <- summary(result.pd.ns)
- > plot(summary.pd.ns)

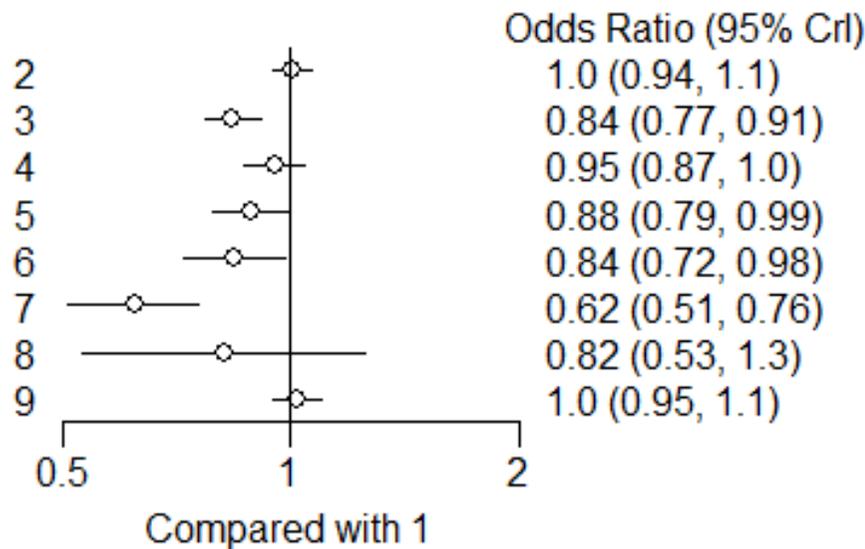


急性心筋梗塞に対する血栓溶解薬の例 大規模試験を除いた場合

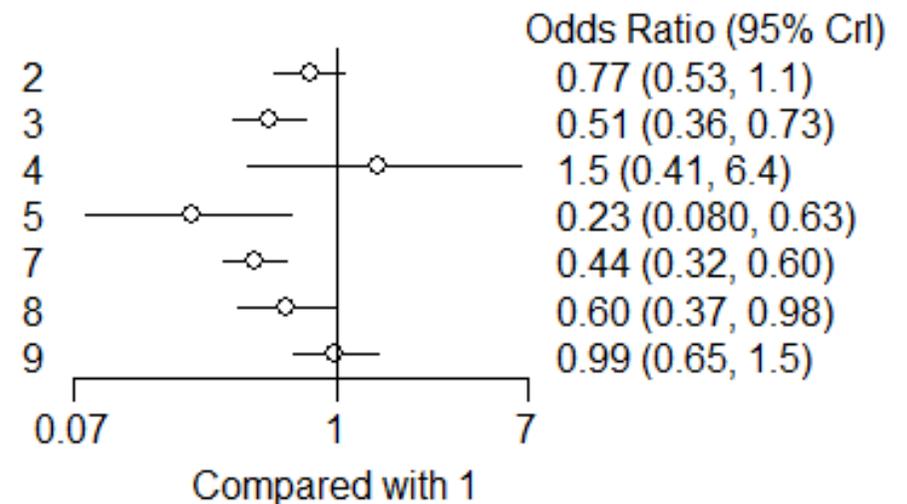
➤ 固定効果モデルの結果

■ 試験サイズによる影響

全試験の場合



1000例/群の試験を除いた場合
(大規模試験が実施されなかった場合)



参考：Certolizumabのメタ回帰の例

➤ 連続値の共変量(平均罹病期間)でのメタ回帰

- 「③すべての治療で1つの交互作用」について
- R gemtcパッケージではセンタリングと標準化が行われる
 - The regressor variable is automatically centered and standardized using the method recommended by Gelman et al. (2008).

➤ ベースラインリスクによるメタ回帰のモデル

- 今回提示したRパッケージ(gemtc以外も)には用意されていない
Rで用いるには直接JAGSなどを利用すること

参考: DIC算出方法のソフトによる違い

例) 2値 ロジット変換

➤ NICE TSDでは

$$DIC = \bar{D}_{res} + p_D$$

$$\blacksquare D_{res} = \sum_i \sum_k dev_{ik} = \sum_i \sum_k 2 \left(r_{ik} \log \left(\frac{r_{ik}}{\hat{r}_{ik}} \right) + (n_{ik} - r_{ik}) \log \left(\frac{n_{ik} - r_{ik}}{n_{ik} - \hat{r}_{ik}} \right) \right)$$

□ dev_{ik} は各iterationで計算される各データポイント(第i群の第k群)についての残差逸脱度, $\hat{r}_{ik} = n_{ik}p_{ik}$

$$\blacksquare p_D = \sum_i \sum_k [\overline{dev}_{ik} - \widetilde{dev}_{ik}]$$

□ \overline{dev}_{ik} は dev_{ik} の事後平均, \widetilde{dev}_{ik} は \hat{r}_{ik} の事後平均 \tilde{r}_{ik} のdeviance

■ ただし, NICE TSDの例示では p_D はWinBUGSの算出機能を用いている場合もある

➤ WinBUGSやSASのDIC算出機能の場合

■ 対数尤度の定数項を含むため値が異なる

□ 値の大きさの評価はしないので問題にはならない

■ \tilde{r}_{ik} ではなく, 対数オッズ比(基本パラメータ)の事後平均から求めた \hat{r}_{ik} が使われる

→ 初期値を指定するパラメーターから算出する仕様

SASのproc MCMCではPriorステートメントに指定するパラメーターのこと

END OF FILE