Automating network meta-analysis

Gert van Valkenhoef

Department of Epidemiology, University Medical Center Groningen (NL),
Faculty of Economics and Business, University of Groningen (NL)

18 Apr. 2011
Bristol, United Kingdom

Outline

1. Introduction
2. MTC models
3. Problem definition
4. Solution & evaluation
5. Discussion

Mini-CV

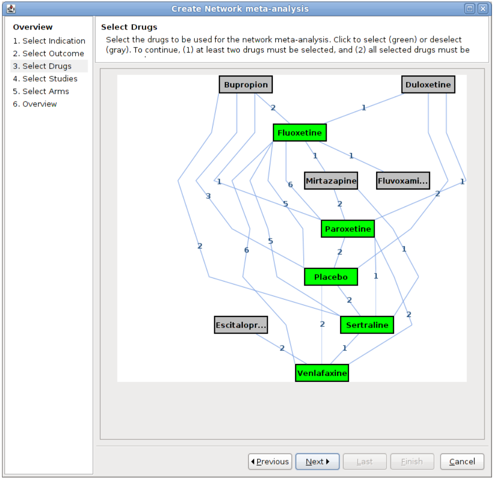
- Born July 1985, The Netherlands
- MSc in Artificial Intelligence (focus on formal aspects)
- Several years experience in software development
- Current: PhD student (medical informatics/statistics)
  - Interests: the intersection between statistics and information technology applied to evidence-based medical decision making.

PhD project

- Part of TI Pharma project Escher
  - "science-driven drug regulation"
- ADDIS decision support system
  - evidence-based decision support
  - for benefit-risk decision making
  - based on a database of clinical trials
  - using evidence synthesis
- **Network meta-analysis** highly relevant
  - but model specification difficult manual step

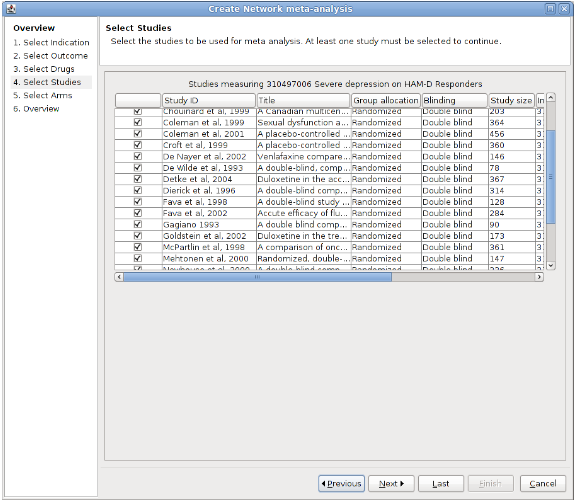
Network meta-analysis in ADDIS

Please select the drugs to be included



Network meta-analysis in ADDIS

Please select the studies to be included



Network meta-analysis in ADDIS

Now please write down the model for us...

test.inco.model

```
	}

	# Random effects in study Alves et al, 1999
	re[1, 1] ~ dnorm(d.Fluoxetine.Paroxetine + d.Paroxetine.Sertraline + d.Sertraline.Venlafaxine + -
w.Fluoxetine.Paroxetine.Sertraline.Venlafaxine, tau.d)
	delta[1, 1, 1] <- 0
	delta[1, 1, 5] <- re[1, 1]
	# Random effects in study Ballus et al, 2000
	re[2, 1] ~ dnorm(d.Paroxetine.Sertraline + d.Sertraline.Venlafaxine + -w.Paroxetine.Sertraline.Venlafaxine, tau.d)
	delta[2, 2, 2] <- 0
	delta[2, 2, 5] <- re[2, 1]
	# Random effects in study Bennie et al, 1995
	re[3, 1] ~ dnorm(d.Fluoxetine.Paroxetine + d.Paroxetine.Sertraline + -w.Fluoxetine.Paroxetine.Sertraline, tau.d)
	delta[3, 1, 1] <- 0
	delta[3, 1, 4] <- re[3, 1]
	# Random effects in study Boyer et al, 1998
	re[4, 1] ~ dnorm(d.Fluoxetine.Paroxetine + d.Paroxetine.Sertraline + -w.Fluoxetine.Paroxetine.Sertraline, tau.d)
	delta[4, 1, 1] <- 0
	delta[4, 1, 4] <- re[4, 1]
	# Random effects in study Chouinard et al, 1999
	re[5, 1] ~ dnorm(d.Fluoxetine.Paroxetine, tau.d)
	delta[5, 1, 1] <- 0
	delta[5, 1, 2] <- re[5, 1]
	# Random effects in study Coleman et al, 1999
	re[6, 1] ~ dnorm(-d.Sertraline.Venlafaxine + -d.Venlafaxine.Placebo + w.Placebo.Sertraline.Venlafaxine, tau.d)
	delta[6, 3, 3] <- 0
	delta[6, 3, 4] <- re[6, 1]
	# Random effects in study Coleman et al, 2001
	re[7, 1] ~ dnorm(d.Venlafaxine.Placebo + d.Sertraline.Venlafaxine + -
w.Fluoxetine.Paroxetine.Sertraline.Venlafaxine.Placebo + d.Paroxetine.Sertraline + d.Fluoxetine.Paroxetine, tau.d)
	delta[7, 1, 1] <- 0
	delta[7, 1, 3] <- re[7, 1]
	# Random effects in study Croft et al, 1999
	re[8, 1] ~ dnorm(-d.Sertraline.Venlafaxine + -d.Venlafaxine.Placebo + w.Placebo.Sertraline.Venlafaxine, tau.d)
	delta[8, 3, 3] <- 0
	delta[8, 3, 4] <- re[8, 1]
```

Network meta-analysis in ADDIS

This would not be feasible for most users!

- So we set out to automate this step

Automating network meta-analysis

- The model has to be manually specified for each analysis
- Difficult step
  - Especially inconsistency models (Lu & Ades, 2006)
  - Problems have no bearing on interpreting the results
- An algorithm to generate the model would enable
  - network meta-analysis for a broader audience
  - network meta-analysis for (semi-)automated decision support

Model generation: open problems

1. How to choose the 'basic parameters'
   - Lu & Ades (2006) identified this as 'determining the ICDF'
2. How to choose study baselines
   - (the model is in terms of relative effects, so a study is parametrized relative to some baseline)
3. How to specify priors
4. How to draw starting values for the markov chains

We solved (1-4), but I will discuss (1)

Agenda

- Goal: determine ICDF in inconsistency models
- Approach:
  1. Understand & formalize the problem
  2. Apply existing algorithms $\rightarrow$ solution
  3. Evaluate running time on published evidence networks
- Conclusions: the problems are subtle, and the solution is (theoretically) inefficient, but very fast on published networks

MTC models

Mixed Treatment Comparison (MTC) – Lu & Ades, 2004; 2006

- is an extension of (Bayesian) pair-wise meta-analysis
- combines direct and indirect evidence
- to summarize evidence from a network of trials
- using a Bayesian Hierarchical Model (BHM)
- estimated using Markov Chain Monte Carlo (MCMC)

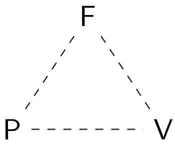
Main assumption:

- *consistency* between direct and indirect evidence
- can be tested using *inconsistency* models

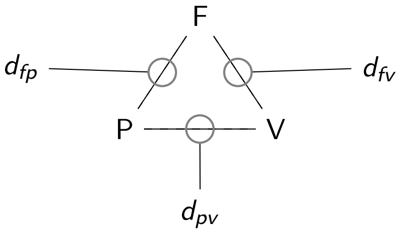
Example MTC (1): data

Study	Fluox	Parox	Venla
Chouinard et al, 1999	67/101	67/102	
De Wilde et al, 1993	25/41	24/37	
Fava et al, 1998	31/54	32/55	
Fava et al, 2002	57/92	64/96	
Gagiano, 1993	27/45	30/45	
Schone and Ludwig, 1993	9/52	20/54	
Alves et al, 1999	30/47		25/40
De Nayer et al, 2002	27/73		37/73
Dierick et al, 1996	95/161		107/153
Rudolph and Feiger, 1999	52/103		57/100
Silverstone and Ravindran, 1999	77/121		84/128
Tylee et al, 1997	58/170		67/171
Ballus et al, 2000		23/43	25/41
McPartlin et al, 1998		128/178	137/183

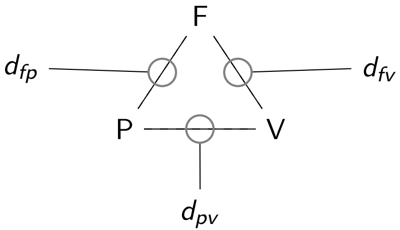
Example MTC (2): consistency



Example MTC (2): consistency

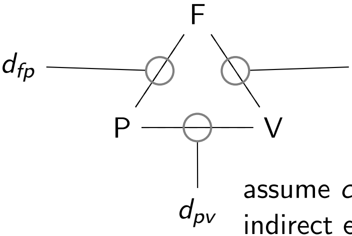


Example MTC (2): consistency



	pair-wise OR	**network** OR
$\exp(d_{fp})$	1.24 (0.92, 1.67)	
$\exp(d_{fv})$	1.30 (1.03, 1.65)	
$\exp(d_{pv})$	1.20 (0.80, 1.82)	

Example MTC (2): consistency

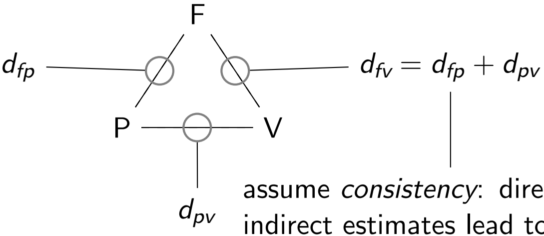


$d_{fv} = d_{fp} + d_{pv}$

assume *consistency*: direct and indirect estimates lead to the same conclusions.

	pair-wise OR	**network** OR
$\exp(d_{fp})$	1.24 (0.92, 1.67)	
$\exp(d_{fv})$	1.30 (1.03, 1.65)	
$\exp(d_{pv})$	1.20 (0.80, 1.82)	

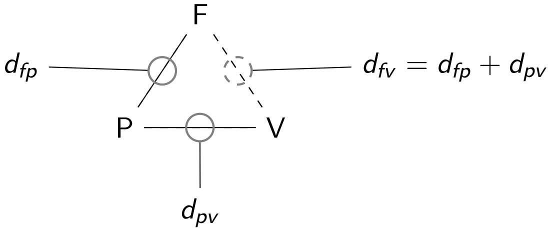
Example MTC (2): consistency



assume *consistency*: direct and indirect estimates lead to the same conclusions.

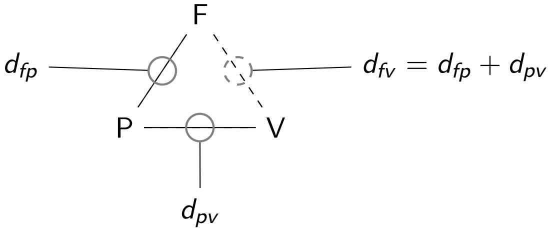
	pair-wise OR	**network** OR
$\exp(d_{fp})$	1.24 (0.92, 1.67)	1.22 (0.92, 1.61)
$\exp(d_{fv})$	1.30 (1.03, 1.65)	1.34 (1.08, 1.67)
$\exp(d_{pv})$	1.20 (0.80, 1.82)	1.11 (0.82, 1.50)

Example MTC (3): functional parameter



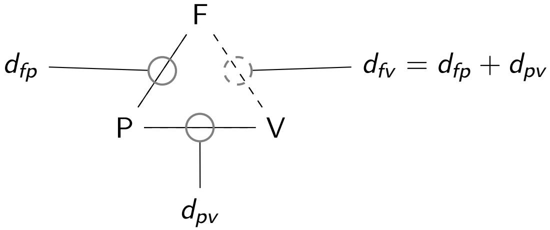
- d_{fv} is fully determined by $d_{fp} + d_{pv}$

Example MTC (3): functional parameter



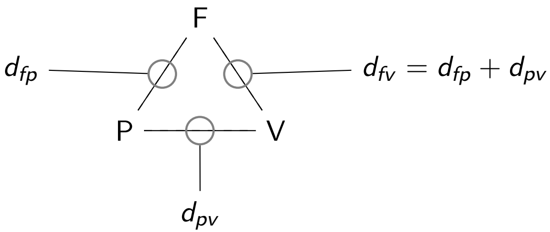
- d_{fv} is fully determined by $d_{fp} + d_{pv}$
- Call d_{fv} a **functional** parameter (can be eliminated)

Example MTC (3): functional parameter

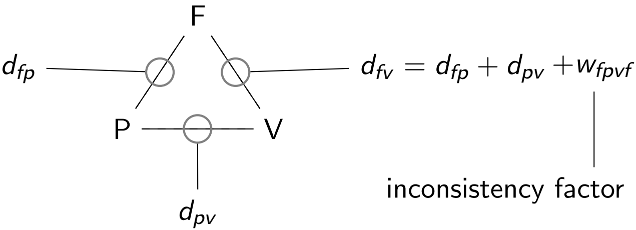


- d_{fv} is fully determined by $d_{fp} + d_{pv}$
- Call d_{fv} a **functional** parameter (can be eliminated)
- And d_{fp}, d_{pv} **basic** parameters ($\sim \mathcal{N}(\cdot, \cdot)$)

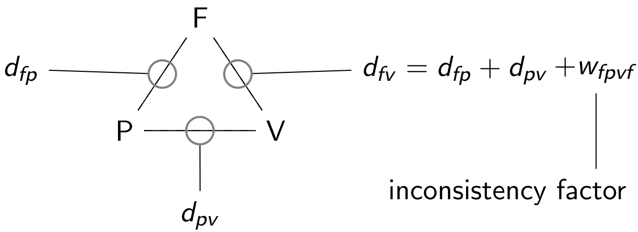
Example MTC (4): inconsistency



Example MTC (4): inconsistency



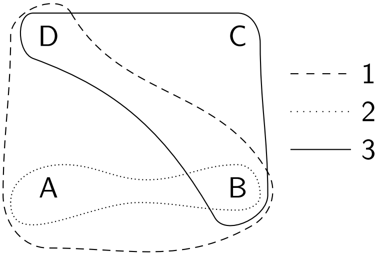
Example MTC (4): inconsistency



Now we have an *inconsistency model*:

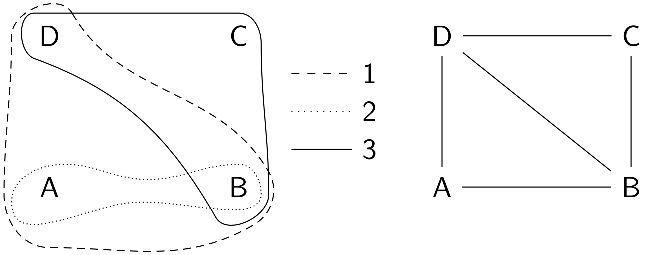
- The estimate of w_{fpvf} is an indicator for (in)consistency
- Basically it 'restores' the dimensionality of the model

Evidence structure



- A set of studies $S = \{S_1, \ldots, S_n\}$
- Each study has a set of treatments $T(S_i)$
- Fully connected comparison graph $G(S_i) = (T(S_i), E(S_i))$
- The evidence structure is the collection of the $G(S_i)$

Evidence graph



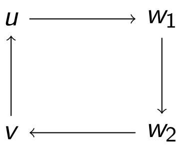
- Evidence graph $G(S) = (T(S), E(S))$
- The graph of all comparisons made in at least one trial
- We assume that $G(S)$ is connected
  - otherwise MTC is not possible

Effect parameters

$$\mathrm{A} \;\underset{d((B,A)) = -d_{AB}}{\overset{d((A,B)) = d_{AB}}{\rightleftarrows}}\; \mathrm{B}$$

Consistency (general formulation)

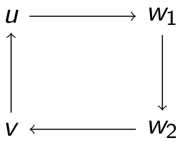
Given a (simple, directed) cycle $\mathbf{C} = ((u, w_1), \ldots, (w_n, v), (v, u))$.



Consistency (general formulation)

Given a (simple, directed) cycle $\mathbf{C} = ((u, w_1), \ldots, (w_n, v), (v, u))$.
Then consistency dictates:

$$w_{\mathbf{C}} = \sum_{e \in \mathbf{C}} d(e) = 0$$



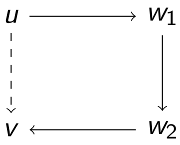
Consistency (general formulation)

Given a (simple, directed) cycle $\mathbf{C} = ((u, w_1), \ldots, (w_n, v), (v, u))$.
Then consistency dictates:

$$w_{\mathbf{C}} = \sum_{e \in \mathbf{C}} d(e) = 0$$

or, if $\mathbf{p} = ((u, w_1), \ldots, (w_n, v))$:

$$d((u, v)) = \sum_{e \in \mathbf{p}} d(e)$$



Basic and functional parameters

Consistency:

$$d((u,v)) = \sum_{e \in \mathbf{p}} d(e)$$

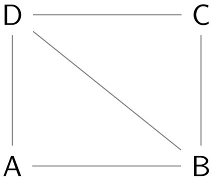
- The functional parameter $d((u,v))$
  - is defined in terms of the basic parameters (right hand side)
  - so we can eliminate it from the BHM
- Each cycle has at least one functional parameter
  - Less, and we don't assume full consistency
- Each functional parameter is the only one in at least one cycle
  - Otherwise the definition becomes circular

Basic parameters: spanning tree

Proposition: if we partition the edges $E(S)$ into basic and functional ones, such that $E_b \cup E_f = E(S)$, then $G_b = (T_b, E_b)$ is a spanning tree of $G(S)$.

Basic parameters: spanning tree

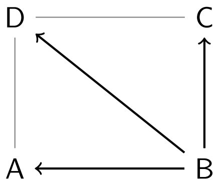
Proposition: if we partition the edges $E(S)$ into basic and functional ones, such that $E_b \cup E_f = E(S)$, then $G_b = (T_b, E_b)$ is a spanning tree of $G(S)$.



So choosing a spanning tree determines basic/functional parameters

Basic parameters: spanning tree

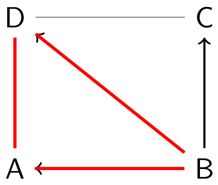
Proposition: if we partition the edges $E(S)$ into basic and functional ones, such that $E_b \cup E_f = E(S)$, then $G_b = (T_b, E_b)$ is a spanning tree of $G(S)$.



So choosing a spanning tree determines basic/functional parameters

Basic parameters: spanning tree

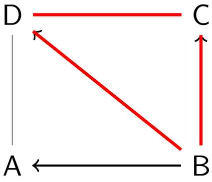
Proposition: if we partition the edges $E(S)$ into basic and functional ones, such that $E_b \cup E_f = E(S)$, then $G_b = (T_b, E_b)$ is a spanning tree of $G(S)$.



$d((A, D)) = d((A, B)) + d((B, D))$

Basic parameters: spanning tree

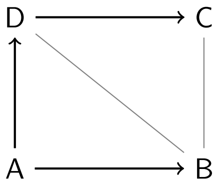
Proposition: if we partition the edges $E(S)$ into basic and functional ones, such that $E_b \cup E_f = E(S)$, then $G_b = (T_b, E_b)$ is a spanning tree of $G(S)$.



$d((C,D)) = d((C,B)) + d((B,D))$

Basic parameters: spanning tree

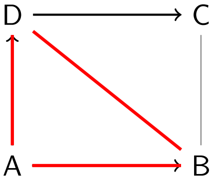
Proposition: if we partition the edges $E(S)$ into basic and functional ones, such that $E_b \cup E_f = E(S)$, then $G_b = (T_b, E_b)$ is a spanning tree of $G(S)$.



So choosing a spanning tree determines basic/functional parameters

Basic parameters: spanning tree

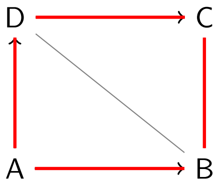
Proposition: if we partition the edges $E(S)$ into basic and functional ones, such that $E_b \cup E_f = E(S)$, then $G_b = (T_b, E_b)$ is a spanning tree of $G(S)$.



$d((B, D)) = d((B, A)) + d((A, D))$

Basic parameters: spanning tree

Proposition: if we partition the edges $E(S)$ into basic and functional ones, such that $E_b \cup E_f = E(S)$, then $G_b = (T_b, E_b)$ is a spanning tree of $G(S)$.



$d((B, C)) = d((B, A)) + d((A, D)) + d((D, C))$

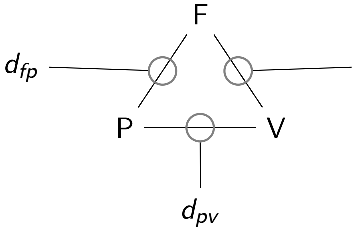
Summary

- Consistency is a natural assumption in MTC
- Basic and functional parameters emerge from consistency
- It turns out that the basic parameters are a spanning tree

Problem definition

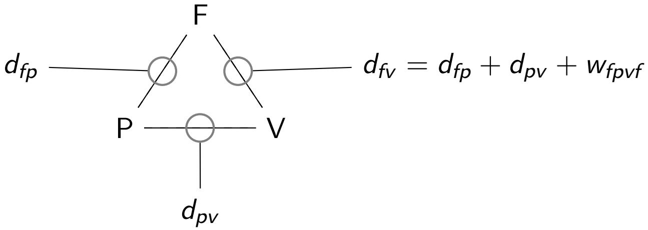
- How should the spanning tree G_b be chosen?
- Examples will show that ‘any will do’ is not the answer

Cycles that cannot be inconsistent (1)

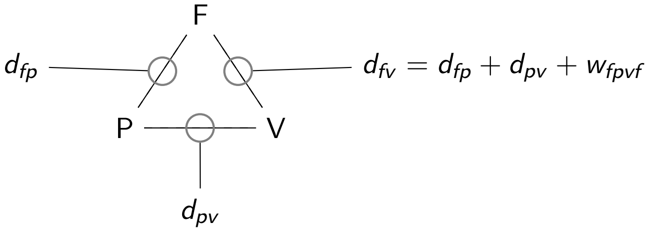


$d_{fv} = d_{fp} + d_{pv} + w_{fpvf}$

Cycles that cannot be inconsistent (1)



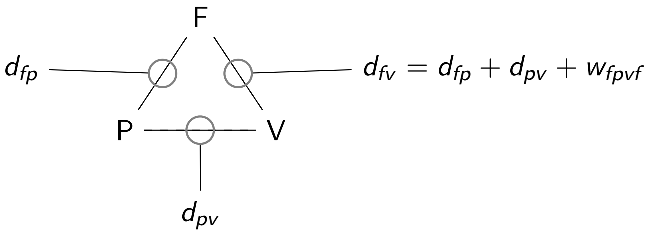
Cycles that cannot be inconsistent (1)



Now imagine that there are only 3-arm trials in this structure.

- Clearly $w_{fpvf} = 0 \rightarrow$ the model is under constrained

Cycles that cannot be inconsistent (1)



Now imagine that there are only 3-arm trials in this structure.

- Clearly $w_{fpvf} = 0 \rightarrow$ the model is under constrained
- So we should exclude w_{fpvf}, i.e. ICDF $= 0$.
  - ICDF = InConsistency Degrees of Freedom

Cycles that cannot be inconsistent (2)

Definition: let $f((u, v), S')$ be the distribution we estimate for $d((u, v))$ based on the studies $S' \subset S$.

Cycles that cannot be inconsistent (2)

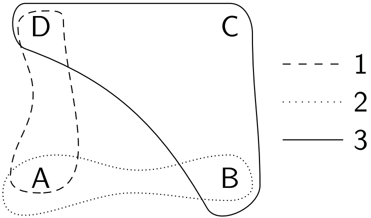
Definition: let $f((u, v), S')$ be the distribution we estimate for $d((u, v))$ based on the studies $S' \subset S$.

Conjecture (internal consistency): the estimate of $d((u, v))$ will be the same regardless of the path we choose:

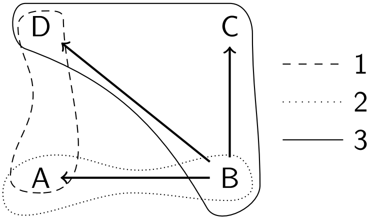
$$f((u, v), S') = f((u, w), S') + f((w, v), S')$$

Supposing that $u, w, v \in T(S_i)$ for all $S_i \in S'$.

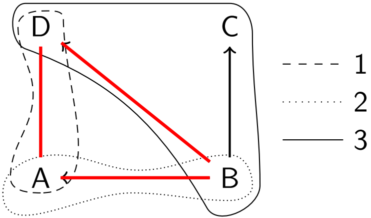
Cycles that cannot be inconsistent (3)



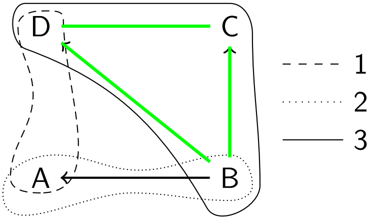
Cycles that cannot be inconsistent (3)



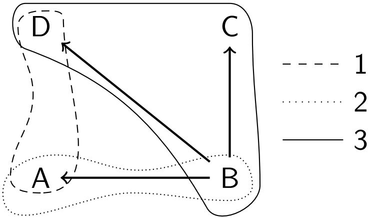
Cycles that cannot be inconsistent (3)



Cycles that cannot be inconsistent (3)

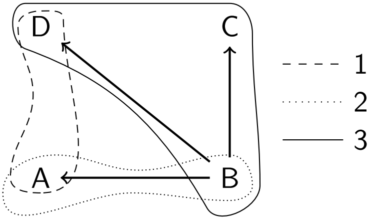


Cycles that cannot be inconsistent (3)



$$\text{ICD} = 1 \ (|E_f| = 2)$$

Cycles that cannot be inconsistent (3)



$$\text{ICD} = 1\ (|E_f| = 2)$$

- ICD = number of w's to be added for a specific G_b
- ICDF = evidence structure 'degrees of freedom' (inherent)

Cycles that cannot be inconsistent (4)

The conjecture gives an idea when to expect the same results from distinct paths.
The next slides have definitions and a theorem to generalize this to any cycle:

- Partition: subdivision into edges with supporting studies
- Reduction: dependent adjacent edges can be reduced to one
- Cycle with ≥ 3 independent adjacent edges can be inconsistent

Elementary partition of a cycle

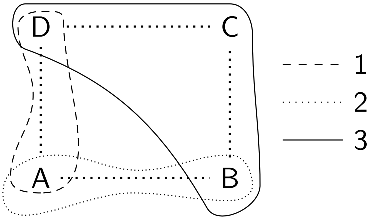
Definition: Let C be a (simple) cycle in $G(S)$, and fix a direction. The *elementary partition* of C is (P, r), where:

- P the set of edges of C (in the chosen direction)
- $r(e) = \{S_i \in S : e \in E(S_i)\}$

Elementary partition of a cycle

Definition: Let C be a (simple) cycle in $G(S)$, and fix a direction. The *elementary partition* of C is (P, r), where:

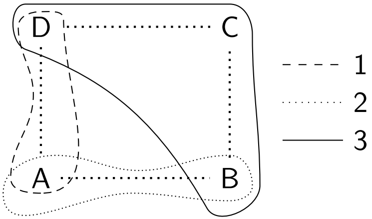
- P the set of edges of C (in the chosen direction)
- $r(e) = \{S_i \in S : e \in E(S_i)\}$



Elementary partition of a cycle

Definition: Let C be a (simple) cycle in $G(S)$, and fix a direction. The *elementary partition* of C is (P, r), where:

- P the set of edges of C (in the chosen direction)
- $r(e) = \{S_i \in S \,:\, e \in E(S_i)\}$

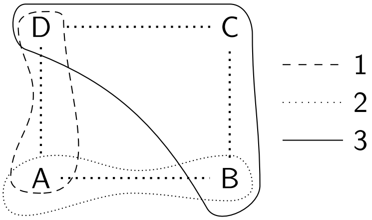


$$P = \{(A, B), (B, C), (C, D), (D, A)\}$$

Elementary partition of a cycle

Definition: Let C be a (simple) cycle in $G(S)$, and fix a direction. The *elementary partition* of C is (P, r), where:

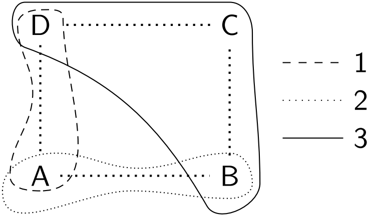
- P the set of edges of C (in the chosen direction)
- $r(e) = \{S_i \in S \ : \ e \in E(S_i)\}$



$$P = \{(A, B), (B, C), (C, D), (D, A)\}$$

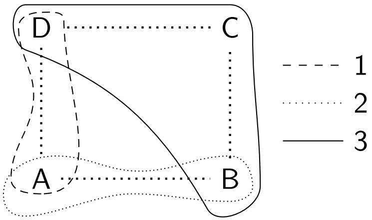
$$r((A, B)) = \{2\}$$
$$r((B, C)) = \{3\}$$
$$r((C, D)) = \{3\}$$
$$r((D, A)) = \{1\}$$

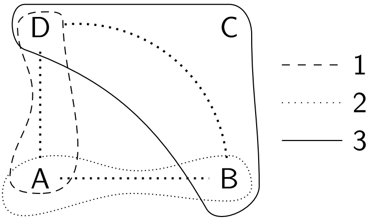
Reduction of a partition



$$P = \{(A,B),(B,C),(C,D),(D,A)\}$$
$$r((A,B)) = \{2\}$$
$$r((B,C)) = \{3\}$$
$$r((C,D)) = \{3\}$$
$$r((D,A)) = \{1\}$$

Reduction of a partition





$P = \{(A,B),(B,C),(C,D),(D,A)\}$
$r((A,B)) = \{2\}$
$r((B,C)) = \{3\}$
$r((C,D)) = \{3\}$
$r((D,A)) = \{1\}$

$P' = \{(A,B),(B,D),(D,A)\}$
$r'((A,B)) = \{2\}$
$r'((B,D)) = \{3\}$
$r'((D,A)) = \{1\}$

Reduction preserves estimates

Let (P, r) be the elementary partition of some cycle C. Then the (in)consistency equation can be written as:

$$F(P, r) = \sum_{e \in P} f(e, r(e)) = w_C$$

Lemma: if (P', r') is obtained from (P, r) in a single reduction step, then $F(P', r') = F(P, r)$.
Proof: by induction using internal consistency.

Inconsistency cycle

Theorem: let C be a cycle in $G(S)$. Suppose the elementary partition of C has m independent pairs of adjacent edges. Then, C is *potentially inconsistent* iff $m \geq 3$.

Inconsistency cycle

Theorem: let C be a cycle in $G(S)$. Suppose the elementary partition of C has m independent pairs of adjacent edges. Then, C is *potentially inconsistent* iff $m \geq 3$.

Proof ($m < 3$):

Inconsistency cycle

Theorem: let C be a cycle in $G(S)$. Suppose the elementary partition of C has m independent pairs of adjacent edges. Then, C is *potentially inconsistent* iff $m \geq 3$.

Proof ($m < 3$):

- if $m = 0$, $F(P, r) = 0$ due to internal consistency

Inconsistency cycle

Theorem: let C be a cycle in $G(S)$. Suppose the elementary partition of C has m independent pairs of adjacent edges. Then, C is *potentially inconsistent* iff $m \geq 3$.

Proof ($m < 3$):

- if $m = 0$, $F(P, r) = 0$ due to internal consistency
- $m = 1$ would imply that C is not a cycle

Inconsistency cycle

Theorem: let C be a cycle in $G(S)$. Suppose the elementary partition of C has m independent pairs of adjacent edges. Then, C is *potentially inconsistent* iff $m \geq 3$.

Proof ($m < 3$):

- if $m = 0$, $F(P, r) = 0$ due to internal consistency
- $m = 1$ would imply that C is not a cycle
- $m = 2$: using reduction, $F(P, r) \neq 0$ implies that

$$f((u,v),R_1) + f((v,u),R_2) \neq 0$$
$$f((u,v),R_1) \neq f((u,v),R_2)$$

Which is just heterogeneity on (u, v).

Inconsistency cycle

Theorem: let C be a cycle in $G(S)$. Suppose the elementary partition of C has m independent pairs of adjacent edges. Then, C is *potentially inconsistent* iff $m \geq 3$.

Proof ($m \geq 3$): (Intuition) we can always choose the data such that $w_C \neq 0$.

Inconsistency degree (not there yet)

- Now we know when a cycle is *potentially* inconsistent
  - *actually* inconsistent depends on data

Inconsistency degree (not there yet)

- Now we know when a cycle is *potentially* inconsistent
  - *actually* inconsistent depends on data
- Seems like we can now define the Inconsistency Degree (ICD)
  - just count the potentially inconsistent cycles
  - (for a spanning tree)

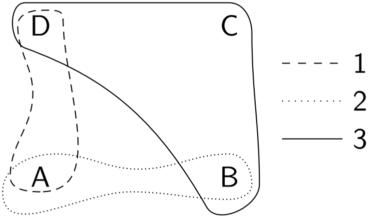
Inconsistency degree (not there yet)

- Now we know when a cycle is *potentially* inconsistent
  - *actually* inconsistent depends on data
- Seems like we can now define the Inconsistency Degree (ICD)
  - just count the potentially inconsistent cycles
  - (for a spanning tree)
- But...

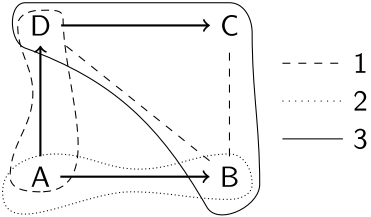
Inconsistency degree (not there yet)

- Now we know when a cycle is *potentially* inconsistent
  - *actually* inconsistent depends on data
- Seems like we can now define the Inconsistency Degree (ICD)
  - just count the potentially inconsistent cycles
  - (for a spanning tree)
- But... there is one more problem to solve

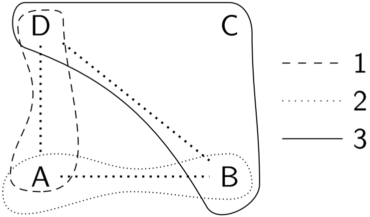
Equivalent reductions



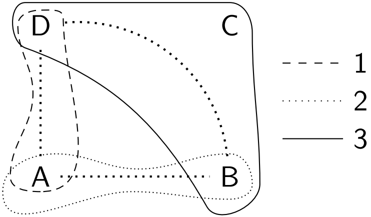
(a) Evidence structure



(b) Spanning tree



(c) Cycle 1



(d) Cycle 2

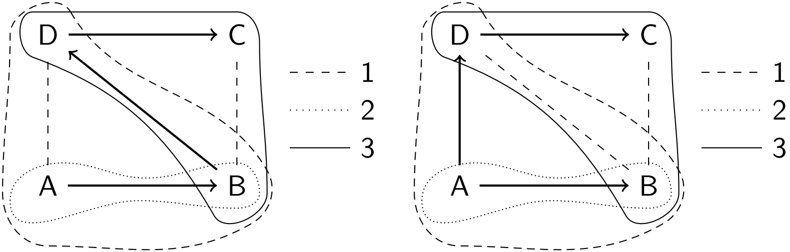
Inconsistency degree

Definition: the cycles C_1 and C_2 are equivalent in S ($C_1 \sim_S C_2$) if their maximally reduced elementary partitions in S are equivalent.

Definition: let G_b be a spanning tree of $G(S)$, and $\mathbf{C}$ the set of cycles induced by G_b. Then $\mathbf{C}/\sim_S$ are the equivalence classes under $\sim_S$ in $\mathbf{C}$.

- If one cycle in a class is potentially inconsistent, so are the others (since they have the 'same' reduction)
- The inconsistency degree $\mathrm{icd}(S, G_b)$ is the number of classes in $\mathbf{C}/\sim_S$ that are potentially inconsistent

Spanning trees with different ICD



(a) ICD = 1 (b) ICD = 2

Figure:

Spanning tree selection problem

The spanning tree selection problem is, given an evidence structure S, to choose the spanning tree G_b that maximizes $\mathrm{icd}(S, G_b)$.

$$\mathrm{icdf}(S) = \max\{\mathrm{icd}(S, G_b) : G_b \,\mathrm{spanning\,tree\,of}\, G(S)\}$$

- One set if ICFs $\mathbf{w}$ can be linearly transformed to another set $\mathbf{w}'$ (Lu & Ades, 2006), given equal ICD.
- So it matters only that we maximize the ICD.

Algorithm

The algorithm is very easy:

1. Generate spanning trees (Gabow & Myers, 1978)
2. Until we find one that maximizes $\mathrm{icd}(S, G_b)$
   - Which we can now calculate easily

Algorithm

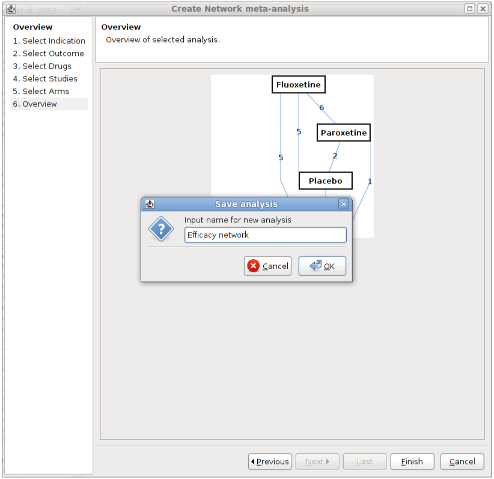
The algorithm is very easy:

1. Generate spanning trees (Gabow & Myers, 1978)
2. Until we find one that maximizes $\mathrm{icd}(S, G_b)$
   - Which we can now calculate easily
3. (there are some additional subtleties related to baseline selection)

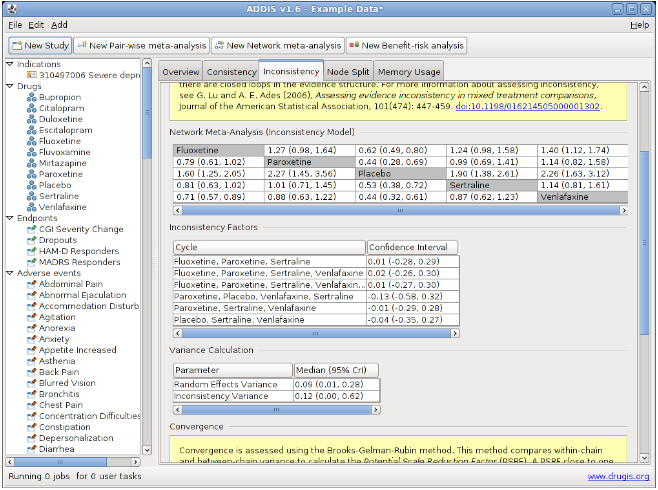
Evaluation

- We extracted the evidence structures from a review of 18 networks (Salanti et al., 2008).
- The 15 non-trivial ones were analyzed using the algorithm.
- All problems were solved in < 4 seconds.
- (In some cases choosing the study baselines is problematic)

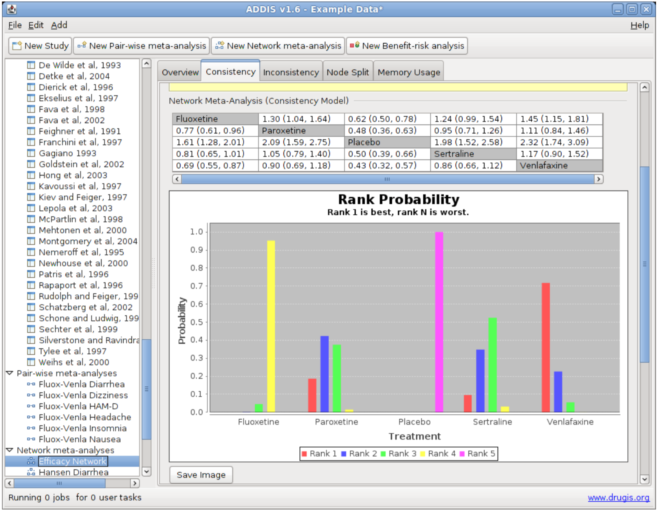
Network meta-analysis in ADDIS



Network meta-analysis in ADDIS



Network meta-analysis in ADDIS



Discussion

- MTC models can be **generated** (automatically)
  - instead of **specified** (manually)
- In some cases, the algorithm could be very inefficient
  - But we have not encountered this in practice

Additional remarks

- To complete automation
  - Assign baselines using search
  - Guess sensible priors and starting values, using heuristics
- This has been implemented in working software
  1. Tool to generate BUGS/JAGS models
  2. R package for MTC analysis
  3. ADDIS software for storing and analyzing clinical trials
- Download @ `http://drugis.org`

Future work

- Relax the baseline selection problem
- Solve the parametrization problem for node-splitting models
- Find a more efficient algorithm

Future work

- Relax the baseline selection problem
- Solve the parametrization problem for node-splitting models
- Find a more efficient algorithm

All of these require a deeper understanding of the statistics

Future work

- Relax the baseline selection problem
- Solve the parametrization problem for node-splitting models
- Find a more efficient algorithm

All of these require a deeper understanding of the statistics

- Which is why I'm here!

Acknowledgements

Co-authors

- Tommi Tervonen (Erasmus University Rotterdam)
- Bert de Brock (University of Groningen)
- Hans Hillege (University Medical Center Groningen)

Funding

- Project Escher (TI Pharma, Netherlands)