

Meta-Regression

Subgroup Analysis

This vignette introduces the regression functionality of the `dtametaTMB` package for meta-analysis of diagnostic test accuracy (DTA) studies.

Validation

The subgroup HSROC and Reitsma implementations reproduce the Cochrane Handbook RF and Anti-CCP examples closely, including the baseline parameters (HSROC: accuracy, threshold, shape; Reitsma: logit sensitivity and specificity), between-study variance estimates, and subgroup effects on accuracy and threshold. The Schuetz CT/MRI analyses yielded results broadly consistent with the published subgroup parameter estimates and variance components. Likelihood-ratio tests led to identical substantive conclusions, although the LR statistics differed somewhat from the published values.

Dummy-coding vs. cell-means parameterization

For subgroup analyses, we distinguish between two equivalent parameterizations of the same model. In the reference-group parameterization, one subgroup serves as the baseline and the remaining subgroup effects are expressed as deviations from this reference. In the group-specific (cell-means) parameterization, each subgroup has its own parameter directly. The former is convenient for testing subgroup differences, whereas the latter is convenient for reporting subgroup-specific estimates and plotting subgroup-specific HSROC curves. The Reitsma subgroup model is fitted twice with both parameterizations. The subgroup HSROC model is estimated using a reference-group parameterization, and subgroup-specific parameters are recovered by evaluating the fitted linear predictors at subgroup-specific design points via Z_{pred} .

How do we include covariates in the Reitsma model?

For sensitivity in study i , we have the number of diseased individuals testing positive: $y_{Ai} \sim \mathcal{B}(n_{Ai}, \pi_{Ai})$.

Similarly for specificity, we have the number of non-diseased individuals testing negative: $y_{Bi} \sim \mathcal{B}(n_{Bi}, \pi_{Bi})$.

Now we introduce a p -dimensional design-vector z_i including study level covariates. Consequently, at the study level, we have

$$\begin{pmatrix} z_i^\top \mu_{Ai} \\ z_i^\top \mu_{Bi} \end{pmatrix} \sim \mathcal{N} \left(\begin{pmatrix} z_i^\top \mu_A \\ z_i^\top \mu_B \end{pmatrix}, \Sigma \right), \quad \text{with} \quad \Sigma = \begin{pmatrix} \sigma_A^2 & \sigma_{AB} \\ \sigma_{AB} & \sigma_B^2 \end{pmatrix}.$$

Let's assume that z_i includes a single binary covariate representing two subgroups. Then using dummy coding with subgroup 1 being the reference, we have

$$z_i^\top \mu_{Ai} = \begin{cases} \mu_{Ai} & \text{for subgroup 1,} \\ \mu_{Ai} + \nu_{A2} & \text{for subgroup 2,} \end{cases} \quad z_i^\top \mu_{Bi} = \begin{cases} \mu_{Bi} & \text{for subgroup 1,} \\ \mu_{Bi} + \nu_{B2} & \text{for subgroup 2,} \end{cases}$$

$$z_i^\top \mu_A = \begin{cases} \mu_A & \text{for subgroup 1,} \\ \mu_A + \nu_{A2} & \text{for subgroup 2,} \end{cases} \quad z_i^\top \mu_B = \begin{cases} \mu_B & \text{for subgroup 1,} \\ \mu_B + \nu_{B2} & \text{for subgroup 2.} \end{cases}$$

```
data("anticcp")
reitsmasub <- fitReitsmaSubgroup(data=anticcp,
                                TP=TP,
                                FP=FP,
                                FN=FN,
                                TN=TN,
                                study=study,
                                subgroup=generation)

reitsmasub
#>
#> Reitsma Subgroup Model
#> -----
#>
#> Number of studies      : 37
#> Number of subgroups   : 2
#> Optimizer              : Converged
#> Hessian                : Positive definite
#> Max |grad|             : 0.0004858134
#> -2 log likelihood      : 533.37 ( df = 7 )
```

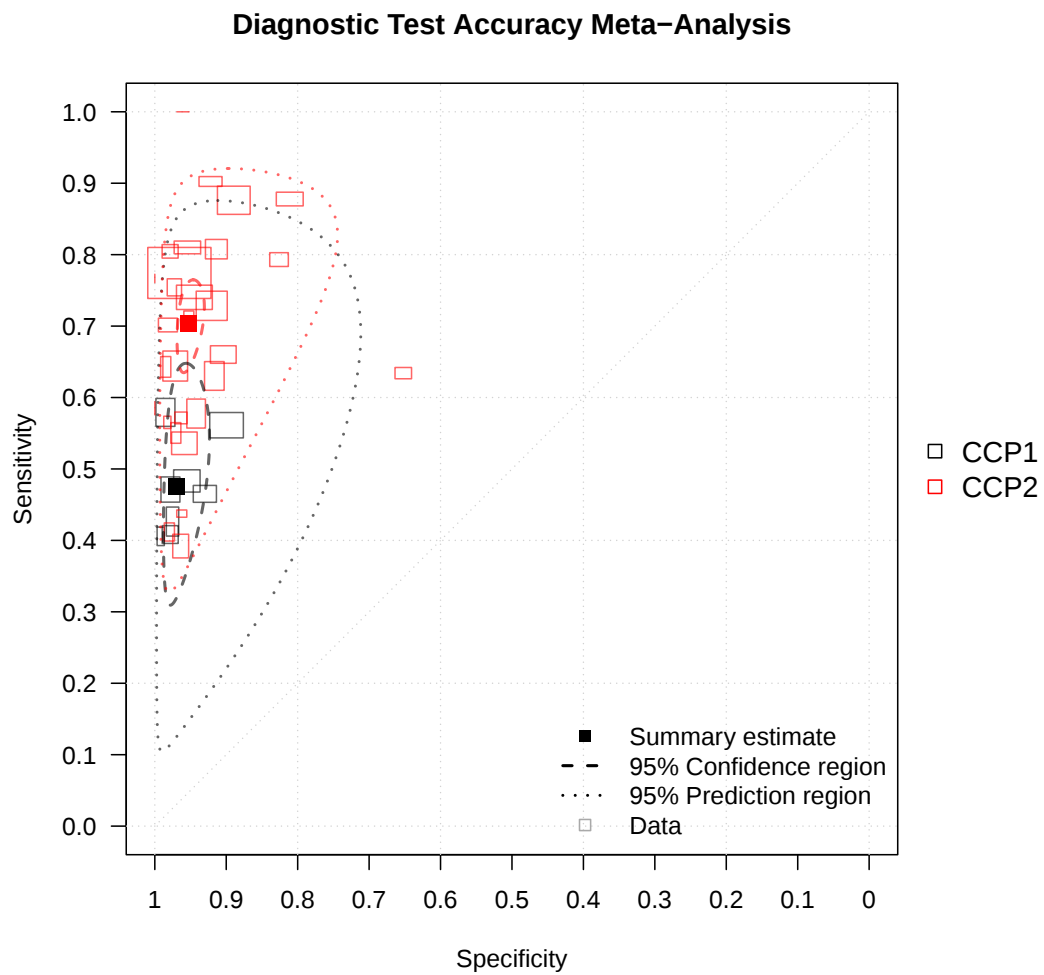
```

#> AIC : 547.37
#> BIC : 558.646
#>
#> Use summary() for parameter estimates.
summary(reitsmasub)
#>
#> Summary: Reitsma Subgroup Model
#> -----
#>
#> Parameter estimates
#> -----
#>
#> Estimate Std_Error
#> mu_A.CCP1 -0.096538827 0.220320580
#> mu_B.CCP1 3.446719200 0.298243680
#> mu_A.CCP2 0.866038550 0.120876810
#> mu_B.CCP2 3.016490700 0.162237530
#> sigma2_A.sens 0.359828660 0.102176490
#> sigma2_B.spec 0.539909270 0.180157210
#> sigma_AB -0.196844970 0.098353409
#> nu_A.CCP2 0.962571510 0.251342000
#> nu_B.CCP2 -0.430209820 0.337711850
#>
#> Sensitivity / specificity
#> -----
#>
#> type Estimate conflevel CI_Lower CI_Upper
#> mu_A.CCP1 sens 0.47588402 0.95 0.37089965 0.58304391
#> mu_B.CCP1 spec 0.96913315 0.95 0.94594449 0.98255779
#> mu_A.CCP2 sens 0.70392073 0.95 0.65229088 0.75081295
#> mu_B.CCP2 spec 0.95331358 0.95 0.93693872 0.96559260
#>
#> Recovered HSROC parameters (Rutter-Gatsonis)
#> -----
#>
#> Lambda Theta beta sigma2_alpha sigma2_theta
#> CCP1 3.0073766 -1.61053430 0.20288656 0.48784243 0.31880558
#> CCP2 3.6840001 -0.88349706 0.20288656 0.48784243 0.31880558
#>
#> Subgroups
#> -----
#> [1] "CCP1" "CCP2"

```

How do I get a summary plot of the Reitsma model?

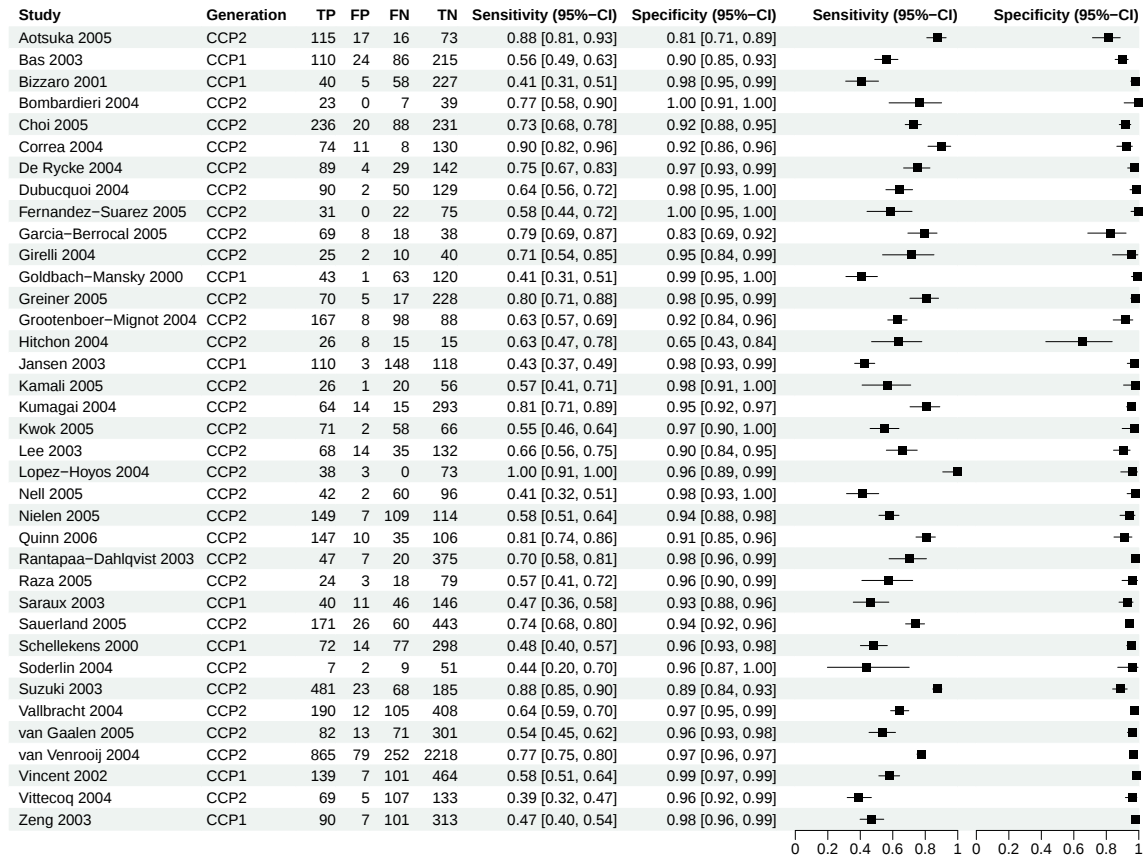
```
plot(reitsmasub,  
     scale=0.01,  
     nudge_legend=-0.2,  
     size="se",  
     col=c("black","red"))
```



How do I get a coupled forest plot?

Note: Rendering forest plots may take longer in an interactive R session due to the underlying grid graphics. Performance is typically faster when knitting the vignette (e.g. via RMarkdown or Quarto).

```
forest(reitsmasub, subgroup_label="Generation")
```



How do I constrain parameters in the Reitsma model?

In sparse data one may wish to fix parameters of the random effects (variance-covariance matrix) at zero. This can be done via the `constrain` argument.

For example,

```
constrainA <- fitReitsmaSubgroup(data=anticcp,
                                TP=TP,
                                FP=FP,
                                FN=FN,
                                TN=TN,
                                study=study,
                                subgroup=generation,
                                constrain="sigma2_A")

summary(constrainA)
#>
#> Summary: Reitsma Subgroup Model
#> -----
#>
#> Parameter estimates
#> -----
#>
#>           Estimate   Std_Error
#> mu_A.CCP1    -5.4394045e-02 0.054985303
#> mu_B.CCP1     3.4466250e+00 0.298754820
#> mu_A.CCP2     9.1797945e-01 0.031391722
#> mu_B.CCP2     2.9968587e+00 0.162162330
#> sigma2_A.sens  4.9303807e-32 0.000000000
#> sigma2_B.spec  5.3963306e-01 0.179976400
#> sigma_AB      0.0000000e+00 0.000000000
#> nu_A.CCP2     9.7237352e-01 0.063315273
#> nu_B.CCP2    -4.4976610e-01 0.337953820
#>
#> Sensitivity / specificity
#> -----
#>
#>           type   Estimate conflevel   CI_Lower   CI_Upper
#> mu_A.CCP1 sens  0.48640484      0.95 0.45954779 0.51334062
#> mu_B.CCP1 spec  0.96913033      0.95 0.94588841 0.98257334
#> mu_A.CCP2 sens  0.71463023      0.95 0.70191901 0.72701020
#> mu_B.CCP2 spec  0.95243201      0.95 0.93577764 0.96492938
#>
#> Recovered HSR0C parameters (Rutter-Gatsonis)
#> -----
#>
#>           Lambda   Theta   beta sigma2_alpha  sigma2_theta
#> CCP1 -3128642.1 -1564321 35.73522 3.262267e-16 8.1556674e-17
#> CCP2 52800433.0 26400216 35.73522 3.262267e-16 8.1556674e-17
#>
#> Subgroups
```

```
#> -----
#> [1] "CCP1" "CCP2"
```

fixes the logit sensitivity variance to zero. This also implies that the random effects covariance is zero. Note that you can also set `constrain` to `"sigma_AB"`, `"sigma2_B"`, or `"all"`.

Constraining fixed effects is controlled by the `sensspec_constrain` argument. For example,

```
constrainsens <- fitReitsmaSubgroup(data=anticcp,
                                   TP=TP,
                                   FP=FP,
                                   FN=FN,
                                   TN=TN,
                                   study=study,
                                   subgroup=generation,
                                   sensspec_constrain="sens")

summary(constrainsens)
#>
#> Summary: Reitsma Subgroup Model
#> -----
#>
#> Parameter estimates
#> -----
#>
#>           Estimate Std_Error
#> mu_A.CCP1      0.653305200 0.12740763
#> mu_B.CCP1      3.081220300 0.32561776
#> mu_A.CCP2      0.653305200 0.00000000
#> mu_B.CCP2      3.118006300 0.17451281
#> sigma2_A.sens   0.541929260 0.14627354
#> sigma2_B.spec   0.577652670 0.19968378
#> sigma_AB       -0.277158320 0.13981426
#> nu_A.CCP2       0.000000000 0.00000000
#> nu_B.CCP2       0.036789072 0.38577329
#>
#> Sensitivity / specificity
#> -----
#>
#>           type  Estimate conflevel  CI_Lower  CI_Upper
#> mu_A.CCP1 sens  0.65775489         0.95 0.59955009 0.71156963
#> mu_B.CCP1 spec  0.95611142         0.95 0.92004961 0.97632601
#> mu_A.CCP2 sens  0.65775489         0.95 0.59955009 0.71156963
#> mu_B.CCP2 spec  0.95762941         0.95 0.94136325 0.96952930
#>
```

```
#> Recovered HSROC parameters (Rutter-Gatsonis)
#> -----
#>           Lambda      Theta      beta sigma2_alpha sigma2_theta
#> CCP1 3.6962516 -1.1843106 0.031918652 0.56469521 0.41833212
#> CCP2 3.7324551 -1.2024124 0.031918652 0.56469521 0.41833212
#>
#> Subgroups
#> -----
#> [1] "CCP1" "CCP2"
```

assumes equal (logit) sensitivities in all subgroups. Note that you can also set `subgroup_constrain` to "spec" or `c("sens","spec")`.

How can I compare constrainsens with the full model?

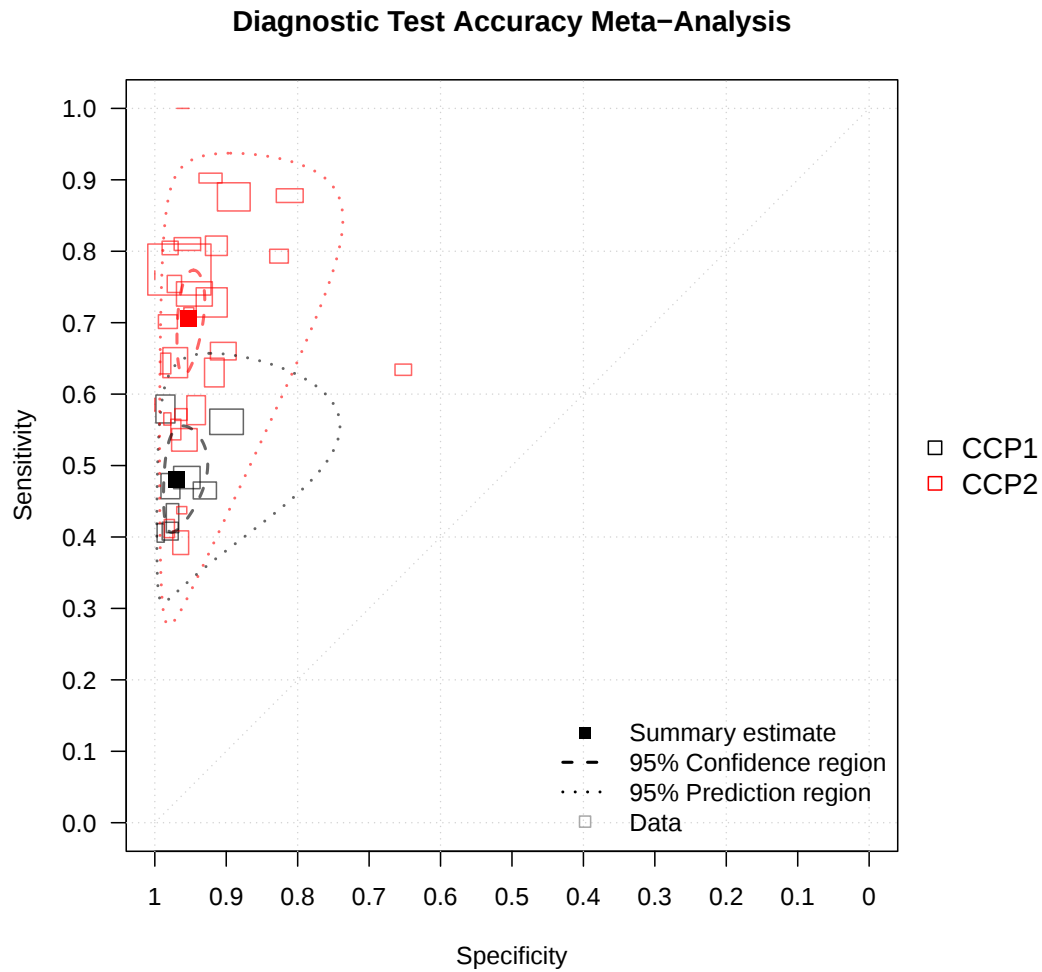
We can perform a likelihood ratio test via `anova`, essentially testing whether there exist subgroup differences in sensitivity.

```
anova(constrainsens,reitsmasub)
#>           Df logLik Df.diff  Chisq Pr(>Chisq)
#> Model 1  6 -272.77
#> Model 2  7 -266.69      1 12.181 0.0004827 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

How do I allow for different subgroup-specific random-effects (co-)variances in the Reitsma model?

```
heteroskedastic <- fitReitsmaSubgroup(data=anticcp,
                                     TP=TP,
                                     FP=FP,
                                     FN=FN,
                                     TN=TN,
                                     study=study,
                                     subgroup=generation,
                                     variances="unequal")
plot(heteroskedastic,
     scale=0.01,
     nudge_legend=-0.2,
```

```
size="se",
col=c("black","red"))
```



```
anova(reitsmasub,heteroskedastic)
#>           Df logLik Df.diff  Chisq Pr(>Chisq)
#> Model 1   7 -266.69
#> Model 2  10 -262.42      3 8.5306   0.03623 *
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

How do we include covariates in the Rutter and Gatsonis model?

The number of diseased individuals from study i who test positive is denoted by $y_{i1} \sim \mathcal{B}(n_{i1}, \pi_{i1})$.

Similarly, the number of non-diseased individuals who test positive is $y_{i2} \sim \mathcal{B}(n_{i2}, \pi_{i2})$.

Now we introduce a p -dimensional design-vector z_i including study level covariates. Consequently, at the study level, we have

$$\text{logit}(\pi_{ij}) = (z_i^\top \theta_i + z_i^\top \alpha_i x_{ij}) \exp(-z_i^\top \beta x_{ij}),$$

$$z_i^\top \alpha_i \sim \mathcal{N}(z_i^\top \Lambda, \sigma_\alpha^2), \quad z_i^\top \theta_i \sim \mathcal{N}(z_i^\top \Theta, \sigma_\theta^2),$$

where

$$x_{ij} = \begin{cases} -0.5 & \text{for non-diseased individuals,} \\ 0.5 & \text{for diseased individuals.} \end{cases}$$

Let's assume that z_i includes a single categorical covariate representing three subgroups. Then using dummy coding with subgroup 1 being the reference, we have

$$\begin{aligned} z_i^\top \theta_i &= \begin{cases} \theta_i & \text{for subgroup 1,} \\ \theta_i + \gamma_2 & \text{for subgroup 2,} \\ \theta_i + \gamma_3 & \text{for subgroup 3,} \end{cases} & z_i^\top \alpha_i &= \begin{cases} \alpha_i & \text{for subgroup 1,} \\ \alpha_i + \xi_2 & \text{for subgroup 2,} \\ \alpha_i + \xi_3 & \text{for subgroup 3,} \end{cases} \\ z_i^\top \beta &= \begin{cases} \beta & \text{for subgroup 1,} \\ \beta + \delta_2 & \text{for subgroup 2,} \\ \beta + \delta_3 & \text{for subgroup 3,} \end{cases} \\ z_i^\top \Lambda &= \begin{cases} \Lambda & \text{for subgroup 1,} \\ \Lambda + \xi_2 & \text{for subgroup 2,} \\ \Lambda + \xi_3 & \text{for subgroup 3,} \end{cases} & z_i^\top \Theta_i &= \begin{cases} \Theta & \text{for subgroup 1,} \\ \Theta + \gamma_2 & \text{for subgroup 2,} \\ \Theta + \gamma_3 & \text{for subgroup 3.} \end{cases} \end{aligned}$$

Note: For prediction `fitRutterGatsonisSubgroup()` uses the prediction matrix

$$Z_{\text{pred}} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

for the case above and therefore recovers the threshold, accuracy, and shape parameters for each subgroup as if it were the reference group.

How do I perform a Rutter and Gatsonis meta-regression with a single categorical study-level covariate?

```
data("RF")
RF2      <- RF[RF$method %in% c("LA","ELISA","Nephelometry"),]
RF2$method <- factor(RF2$method,levels=c("LA","ELISA","Nephelometry"))
ruttergatsonissub <- fitRutterGatsonisSubgroup(data=RF2,
                                              TP=TP,
                                              FP=FP,
                                              FN=FN,
                                              TN=TN,
                                              study=study,
                                              subgroup=method,
                                              constrain="shape") # assumes equal
                                                                # shapes in subgroups
```

```
ruttergatsonissub
#>
#> Rutter & Gatsonis Subgroup Model
#> -----
#>
#> Number of studies      : 47
#> Number of subgroups   : 3
#> Optimizer              : Converged
#> Hessian                : Positive definite
#> Max |grad|             : 0.0001105815
#> -2 log likelihood      : 753.722 ( df = 9 )
#> AIC                   : 771.722
#> BIC                   : 788.374
#>
#> Use summary() for parameter estimates.
summary(ruttergatsonissub)
#>
#> Summary: Rutter & Gatsonis Subgroup Model
#> -----
#>
#> Parameter estimates
#> -----
#>
#>               Estimate Std. Error
#> Lambda_LA      2.456304800 0.323576880
#> xi_ELISA       0.248533150 0.439541070
```

```

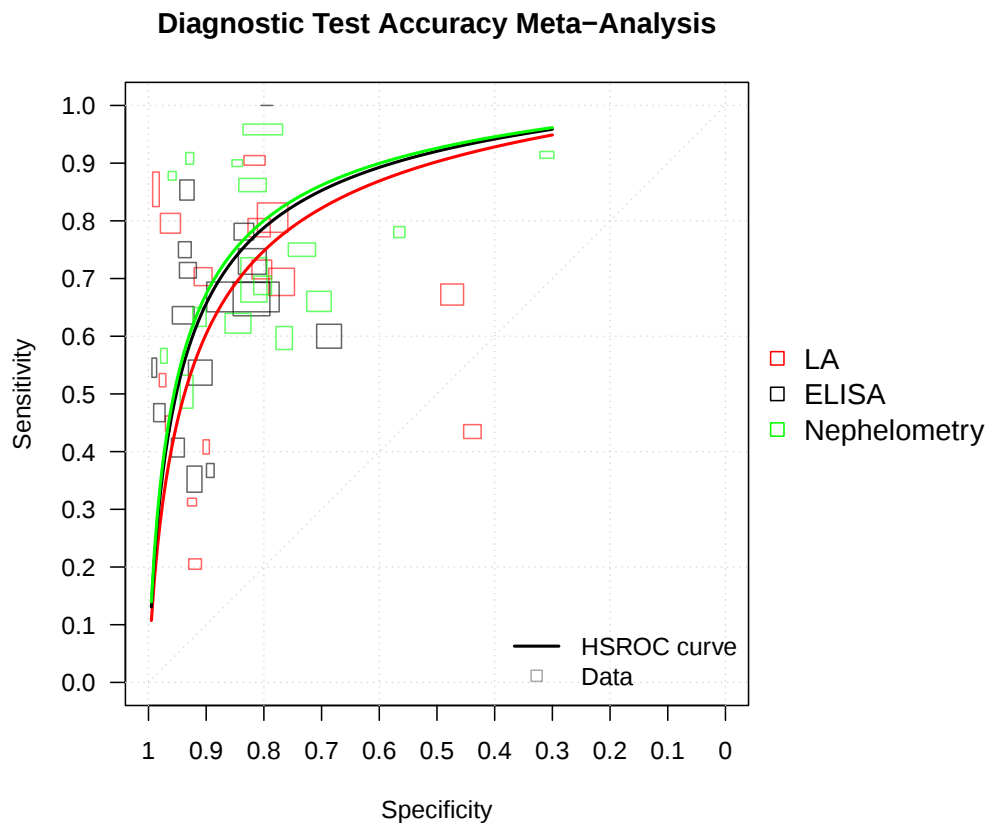
#> xi_Nephelometry      0.331427880 0.442600560
#> Theta_LA             -0.550052130 0.213340280
#> gamma_ELISA          -0.196254860 0.260961940
#> gamma_Nephelometry   0.495859450 0.262230430
#> beta_LA              0.197685730 0.169839590
#> delta_ELISA          0.000000000 0.000000000
#> delta_Nephelometry   0.000000000 0.000000000
#> sigma2_alpha         1.277914600 0.308524740
#> sigma2_theta         0.476840250 0.113445150
#> Lambda_LA            2.456304800 0.323576880
#> Lambda_ELISA         2.704837900 0.326954100
#> Lambda_Nephelometry  2.787732600 0.305779090
#> Theta_LA             -0.550052130 0.213340280
#> Theta_ELISA          -0.746306990 0.209912990
#> Theta_Nephelometry   -0.054192678 0.212131240
#> beta_LA              0.197685730 0.169839590
#> beta_ELISA           0.197685730 0.169839590
#> beta_Nephelometry     0.197685730 0.169839590
#> logitsens            0.826166230 0.286299160
#> logitsens            1.051308700 0.286068740
#> logitsens            1.126401800 0.276894640
#> sens                 0.695543690 0.060627474
#> sens                 0.741026120 0.054898425
#> sens                 0.755174250 0.051193970
#>
#> Sensitivity / specificity
#> -----
#>      subgroup      spec conflevel logitsens Std_Error CI_Lower CI_Upper
#> 1      LA 0.84615385      0.95 0.82616623 0.28629916 0.26503018 1.3873023
#> 2      ELISA 0.84615385      0.95 1.05130870 0.28606874 0.49062426 1.6119931
#> 3 Nephelometry 0.84615385      0.95 1.12640180 0.27689464 0.58369827 1.6691053
#>      Sens SensCI_Lower SensCI_Upper
#> 1 0.69554369 0.56587242 0.80016122
#> 2 0.74102612 0.62025348 0.83368792
#> 3 0.75517425 0.64191793 0.84145650
#>
#> Recovered bivariate parameters (Reitsma)
#> -----
#>      mu_A.sens mu_B.spec sigma2_A.sens sigma2_B.spec sigma_AB
#> LA      0.61428088 1.9629472 0.65348135 0.97037777 -0.1573616
#> ELISA    0.54906777 2.3167685 0.65348135 0.97037777 -0.1573616
#> Nephelometry 1.21359030 1.5985019 0.65348135 0.97037777 -0.1573616

```

```
#>
#> Subgroups
#> -----
#> [1] "LA"          "ELISA"       "Nephelometry"
```

How do I get a summary plot of the Rutter and Gatsonis subgroup model?

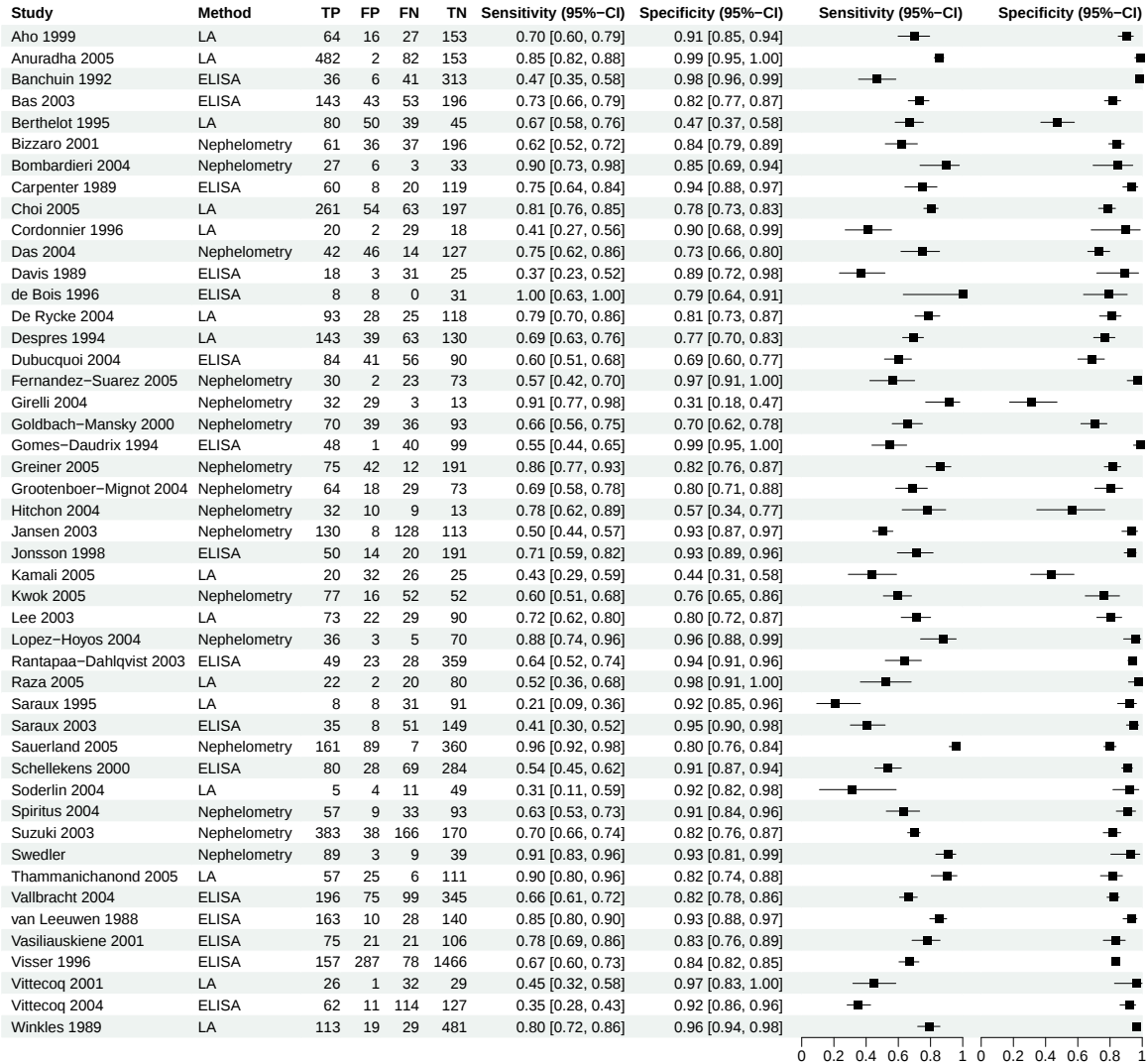
```
plot(ruttergatsonissub,
     specrange=c(0.3,0.995),
     size="se",
     col=c("red","black","green"),
     scale=0.015)
```



How do I get a coupled forest plot?

Note: Rendering forest plots may take longer in an interactive R session due to the underlying grid graphics. Performance is typically faster when knitting the vignette (e.g. via RMarkdown or Quarto).

```
forest(ruttergatsonissub, subgroup_label = "Method")
```



How do I constrain parameters in the Rutter and Gatsonis subgroup model?

In the Rutter and Gatsonis model, all parameter constraints are controlled by `constrain`, i.e.,

- `constrain="sigma2_alpha"` sets σ_α^2 to zero,
- `constrain="sigma2_theta"` sets σ_θ^2 to zero,
- `constrain="accuracy"` assumes equal accuracy parameters across subgroups, i.e. $\xi_2 = \xi_3 = \dots = 0$,
- `constrain="threshold"` assumes equal threshold parameters across subgroups, i.e. $\gamma_2 = \gamma_3 = \dots = 0$,
- `constrain="shape"` assumes equal shape parameters across subgroups, i.e. $\delta_2 = \delta_3 = \dots = 0$,
- `constrain="shape_zero"` fixes all shape parameters at zero.

Constraints can also be combined, for example `constrain=c("shape", "sigma2_theta")`.

How do I use the general Rutter and Gatsonis regression function?

This method is for advanced users who feel comfortable specifying their own design and prediction matrices. For study-level covariates, the design matrix Z needs two identical consecutive rows per study, one for the diseased and one for the non-diseased. Of note, there are neither `plot()` nor `forest()` methods for `fitRutterGatsonisReg()`.

Let's reproduce the subgroup-analysis from before.

```
# Specify design matrix Z
Z <- model.matrix(~method,data=RF2)
# For study level-covariates, we need to two identical consecutive
# rows per study (diseased and non-diseased).
Z2 <- Z[rep(seq_len(nrow(Z)), each = 2), , drop = FALSE]
# Specify prediction matrix Z_pred
Z_pred <- matrix(c(1,0,0,1,1,0,1,0,1),ncol=3,nrow=3,byrow=T)
constrain <- list(shape_coef=factor(c(1, rep(NA, ncol(Z2) - 1))))

ruttergatsonisreg <- fitRutterGatsonisReg(data=RF2,
                                         TP=TP,
                                         FP=FP,
                                         FN=FN,
                                         TN=TN,
                                         study=study,
                                         Z=Z2,
                                         Z_pred=Z_pred,
```

```
map=constrain)
```

```
ruttergatsonisreg
#>
#> Rutter & Gatsonis Regression Model
#> -----
#>
#> Number of studies : 47
#> Optimizer          : Converged
#> Hessian            : Positive definite
#> Max |grad|         : 0.0005885859
#> -2 log likelihood : 753.722 ( df = 9 )
#> AIC                : 771.722
#> BIC                : 788.374
#>
#> Use summary() for parameter estimates.
summary(ruttergatsonisreg)
#>
#> Summary: Rutter & Gatsonis Regression Model
#> -----
#>
#> Parameter estimates
#> -----
#>               Estimate Std. Error
#> accuracy_coef  2.456311600 0.323576730
#> accuracy_coef  0.248522080 0.439540660
#> accuracy_coef  0.331415840 0.442600140
#> threshold_coef -0.550050840 0.213341490
#> threshold_coef -0.196250090 0.260963930
#> threshold_coef  0.495856950 0.262232320
#> shape_coef     0.197681910 0.169839500
#> shape_coef     0.000000000 0.000000000
#> shape_coef     0.000000000 0.000000000
#> sigma2_alpha   1.277912000 0.308524000
#> sigma2_theta   0.476848050 0.113447530
#> Lambda_Pred    2.456311600 0.323576730
#> Lambda_Pred    2.704833600 0.326953910
#> Lambda_Pred    2.787727400 0.305778790
#> Theta_Pred     -0.550050840 0.213341490
#> Theta_Pred     -0.746300930 0.209914150
#> Theta_Pred     -0.054193884 0.212132370
#> beta_Pred      0.197681910 0.169839500
```

```

#> beta_Pred      0.197681910 0.169839500
#> beta_Pred      0.197681910 0.169839500
#> logitsens      0.826171290 0.286299520
#> logitsens      1.051304200 0.286068870
#> logitsens      1.126396500 0.276894900
#> sens           0.695544760 0.060627429
#> sens           0.741025250 0.054898569
#> sens           0.755173280 0.051194156
#>
#> Sensitivity / specificity
#> -----
#>          spec conflevel  logitsens  Std_Error  CI_Lower  CI_Upper      Sens
#> 1 0.84615385      0.95 0.82617129 0.28629952 0.26503454 1.3873080 0.69554476
#> 2 0.84615385      0.95 1.05130420 0.28606887 0.49061948 1.6119888 0.74102525
#> 3 0.84615385      0.95 1.12639650 0.27689490 0.58369248 1.6691005 0.75517328
#>  SensCI_Lower SensCI_Upper
#> 1   0.56587349   0.80016214
#> 2   0.62025236   0.83368733
#> 3   0.64191660   0.84145586

```

How do I compare models?

In the previous section we fitted the RF data set using the Rutter and Gatsonis subgroup model while keeping the shape parameter equal across all subgroups `constrain="shape"`. Now let's fit the full model allowing for different shape parameters across subgroups and check whether the data lend support to this approach. In addition, let's also consider a model that constrains both `shape` and `accuracy`.

```

ruttergatsonissubfull <- fitRutterGatsonisSubgroup(data=RF2,
                                                    TP=TP,
                                                    FP=FP,
                                                    FN=FN,
                                                    TN=TN,
                                                    study=study,
                                                    subgroup=method,
                                                    constrain=NULL)

ruttergatsonissubfull
#>
#> Rutter & Gatsonis Subgroup Model
#> -----
#>

```

```

#> Number of studies      : 47
#> Number of subgroups   : 3
#> Optimizer              : Converged
#> Hessian                : Positive definite
#> Max |grad|             : 0.000337596
#> -2 log likelihood     : 753.553 ( df = 11 )
#> AIC                   : 775.553
#> BIC                   : 795.905
#>
#> Use summary() for parameter estimates.
summary(ruttergatsonissubfull)
#>
#> Summary: Rutter & Gatsonis Subgroup Model
#> -----
#>
#> Parameter estimates
#> -----
#>
#>               Estimate Std. Error
#> Lambda_LA      2.42436690 0.330492250
#> xi_ELISA       0.29745830 0.513153920
#> xi_Nephelometry 0.37160069 0.450866930
#> Theta_LA      -0.50294867 0.244516620
#> gamma_ELISA    -0.25782423 0.370246760
#> gamma_Nephelometry 0.39706560 0.359442990
#> beta_LA        0.27774914 0.266423430
#> delta_ELISA    -0.10288872 0.426612210
#> delta_Nephelometry -0.16038309 0.397537880
#> sigma2_alpha   1.27306770 0.307989470
#> sigma2_theta   0.47623960 0.113435690
#> Lambda_LA      2.42436690 0.330492250
#> Lambda_ELISA   2.72182520 0.392995970
#> Lambda_Nephelometry 2.79596760 0.307081750
#> Theta_LA      -0.50294867 0.244516620
#> Theta_ELISA    -0.76077289 0.278171850
#> Theta_Nephelometry -0.10588306 0.263228430
#> beta_LA        0.27774914 0.266423430
#> beta_ELISA     0.17486043 0.333215410
#> beta_Nephelometry 0.11736605 0.295003220
#> logitsens      0.81869246 0.275191540
#> logitsens      1.06269910 0.324058690
#> logitsens      1.12064950 0.288349850
#> sens           0.69395872 0.058445183

```

```

#> sens          0.74320602 0.061846871
#> sens          0.75410918 0.053468288
#>
#> Sensitivity / specificity
#> -----
#>      subgroup      spec conflevel  logitsens Std_Error  CI_Lower  CI_Upper
#> 1      LA 0.84615385      0.95 0.81869246 0.27519154 0.27932696 1.3580580
#> 2      ELISA 0.84615385      0.95 1.06269910 0.32405869 0.42755578 1.6978425
#> 3 Nephelometry 0.84615385      0.95 1.12064950 0.28834985 0.55549422 1.6858049
#>      Sens SensCI_Lower SensCI_Upper
#> 1 0.69395872 0.56938121 0.79544389
#> 2 0.74320602 0.60528986 0.84525275
#> 3 0.75410918 0.63540935 0.84367166
#>
#> Recovered bivariate parameters (Reitsma)
#> -----
#>      mu_A.sens mu_B.spec sigma2_A.sens sigma2_B.spec  sigma_AB
#> LA      0.61727341 1.9706525 0.60182822 1.04887170 -0.15797268
#> ELISA    0.54989770 2.3155356 0.66704717 0.94632080 -0.15797268
#> Nephelometry 1.21845830 1.5947592 0.70652257 0.89344721 -0.15797268
#>
#> Subgroups
#> -----
#> [1] "LA"          "ELISA"          "Nephelometry"
ruttergatsonissubacc <- fitRutterGatsonisSubgroup(data=RF2,
                                                    TP=TP,
                                                    FP=FP,
                                                    FN=FN,
                                                    TN=TN,
                                                    study=study,
                                                    subgroup=method,
                                                    constrain=c("shape","accuracy"))

ruttergatsonissubacc
#>
#> Rutter & Gatsonis Subgroup Model
#> -----
#>
#> Number of studies : 47
#> Number of subgroups : 3
#> Optimizer : Converged
#> Hessian : Positive definite
#> Max |grad| : 0.00030144

```

```

#> -2 log likelihood      : 754.332 ( df = 7 )
#> AIC                    : 768.332
#> BIC                    : 781.283
#>
#> Use summary() for parameter estimates.
summary(ruttergatsonissubacc)
#>
#> Summary: Rutter & Gatsonis Subgroup Model
#> -----
#>
#> Parameter estimates
#> -----
#>
#>              Estimate Std. Error
#> Lambda_LA      2.658098600 0.191816080
#> xi_ELISA        0.000000000 0.000000000
#> xi_Nephelometry 0.000000000 0.000000000
#> Theta_LA       -0.557459140 0.212932770
#> gamma_ELISA     -0.193852540 0.260936540
#> gamma_Nephelometry 0.497028160 0.262130940
#> beta_LA         0.189496650 0.164729980
#> delta_ELISA     0.000000000 0.000000000
#> delta_Nephelometry 0.000000000 0.000000000
#> sigma2_alpha    1.290646500 0.312339070
#> sigma2_theta    0.476712770 0.113444900
#> Lambda_LA      2.658098600 0.191816080
#> Lambda_ELISA    2.658098600 0.191816080
#> Lambda_Nephelometry 2.658098600 0.191816080
#> Theta_LA       -0.557459140 0.212932770
#> Theta_ELISA     -0.751311680 0.208099540
#> Theta_Nephelometry -0.060430981 0.209583320
#> beta_LA         0.189496650 0.164729980
#> beta_ELISA      0.189496650 0.164729980
#> beta_Nephelometry 0.189496650 0.164729980
#> logitsens       1.007344500 0.165264740
#> logitsens       1.007344500 0.165264740
#> logitsens       1.007344500 0.165264740
#> sens             0.732500150 0.032382581
#> sens             0.732500150 0.032382581
#> sens             0.732500150 0.032382581
#>
#> Sensitivity / specificity
#> -----

```

```

#>      subgroup      spec conflevel logitsens Std_Error CI_Lower CI_Upper
#> 1          LA 0.84615385      0.95 1.0073445 0.16526474 0.6834316 1.3312575
#> 2          ELISA 0.84615385      0.95 1.0073445 0.16526474 0.6834316 1.3312575
#> 3 Nephelometry 0.84615385      0.95 1.0073445 0.16526474 0.6834316 1.3312575
#>      Sens SensCI_Lower SensCI_Upper
#> 1 0.73250015 0.66450416 0.79104856
#> 2 0.73250015 0.66450416 0.79104856
#> 3 0.73250015 0.66450416 0.79104856
#>
#> Recovered bivariate parameters (Reitsma)
#> -----
#>      mu_A.sens mu_B.spec sigma2_A.sens sigma2_B.spec      sigma_AB
#> LA          0.70183981 2.0739937      0.66138278      0.96615672 -0.15405114
#> ELISA        0.52551118 2.2871117      0.66138278      0.96615672 -0.15405114
#> Nephelometry 1.15393750 1.5275698      0.66138278      0.96615672 -0.15405114
#>
#> Subgroups
#> -----
#> [1] "LA"          "ELISA"          "Nephelometry"

```

How do I get the log likelihood, the AIC, and the BIC of a model?

```

logLik(ruttergatsonissubfull)
#> 'log Lik.' -376.7765 (df=11)
AIC(ruttergatsonissubfull)
#> [1] 775.5531
BIC(ruttergatsonissubfull)
#> [1] 795.9047

logLik(ruttergatsonissub)
#> 'log Lik.' -376.8612 (df=9)
AIC(ruttergatsonissub)
#> [1] 771.7223
BIC(ruttergatsonissub)
#> [1] 788.3736

logLik(ruttergatsonissubacc)
#> 'log Lik.' -377.1659 (df=7)
AIC(ruttergatsonissubacc)
#> [1] 768.3318

```

```
BIC(ruttergatsonissubacc)
#> [1] 781.2829
```

How do I perform a likelihood ratio test of the two models?

The tests below formally investigate whether the HSROC curve shapes and/or accuracy parameters are equal in all subgroups.

```
anova(ruttergatsonissub,
      ruttergatsonissubfull)
#>           Df  logLik Df.diff  Chisq Pr(>Chisq)
#> Model 1   9 -376.86
#> Model 2  11 -376.78      2 0.1692    0.9189
anova(ruttergatsonissubacc,
      ruttergatsonissub)
#>           Df  logLik Df.diff  Chisq Pr(>Chisq)
#> Model 1   7 -377.17
#> Model 2   9 -376.86      2 0.6095    0.7373
anova(ruttergatsonissubacc,
      ruttergatsonissubfull)
#>           Df  logLik Df.diff  Chisq Pr(>Chisq)
#> Model 1   7 -377.17
#> Model 2  11 -376.78      4 0.7787    0.9413
```

How do I replicate results from the Cochrane Handbook with respect to the schuetz data set?

What are the results for the full data set?

```
data(schuetz)
head(schuetz)
#>   test      study TP FP FN TN indirect
#> 1  CT Achenbach 2005 25 4  0 19         1
#> 2  CT Alkadhi 2008 57 12 2 79         1
#> 3  CT Andreini 2007 17  0 0 44         1
#> 4  CT Bayrak 2008 64  4 0 32         1
#> 5 MRI Bedaux 2002  7  1 0  1         1
#> 6 MRI Bogaert 2003 12  3 3  1         1
schuetz$test <- factor(schuetz$test, levels=c("MRI", "CT"))
```

```

schuetzreitsma <- fitReitsmaSubgroup(data=schuetz,
                                     TP=TP,FP=FP,FN=FN,TN=TN,
                                     study=study,
                                     subgroup=test,
                                     variances="common")

schuetzreitsma
#>
#> Reitsma Subgroup Model
#> -----
#>
#> Number of studies      : 108
#> Number of subgroups   : 2
#> Optimizer              : Converged
#> Hessian                 : Positive definite
#> Max |grad|             : 0.0006484896
#> -2 log likelihood      : 951.843 ( df = 7 )
#> AIC                    : 965.843
#> BIC                    : 984.618
#>
#> Use summary() for parameter estimates.
summary(schuetzreitsma)
#>
#> Summary: Reitsma Subgroup Model
#> -----
#>
#> Parameter estimates
#> -----
#>
#>           Estimate Std_Error
#> mu_A.MRI      2.09345670 0.26392122
#> mu_B.MRI      0.86046947 0.25722635
#> mu_A.CT       3.48272800 0.15698588
#> mu_B.CT       1.92906880 0.12066635
#> sigma2_A.sens 0.85003921 0.21954582
#> sigma2_B.spec 0.85268255 0.16833781
#> sigma_AB      0.18574633 0.13420667
#> nu_A.CT       1.38930060 0.30152198
#> nu_B.CT       1.06860260 0.28340287
#>
#> Sensitivity / specificity
#> -----
#>
#>           type   Estimate conflevel   CI_Lower   CI_Upper
#> mu_A.MRI sens 0.89026558         0.95 0.82866292 0.93154914

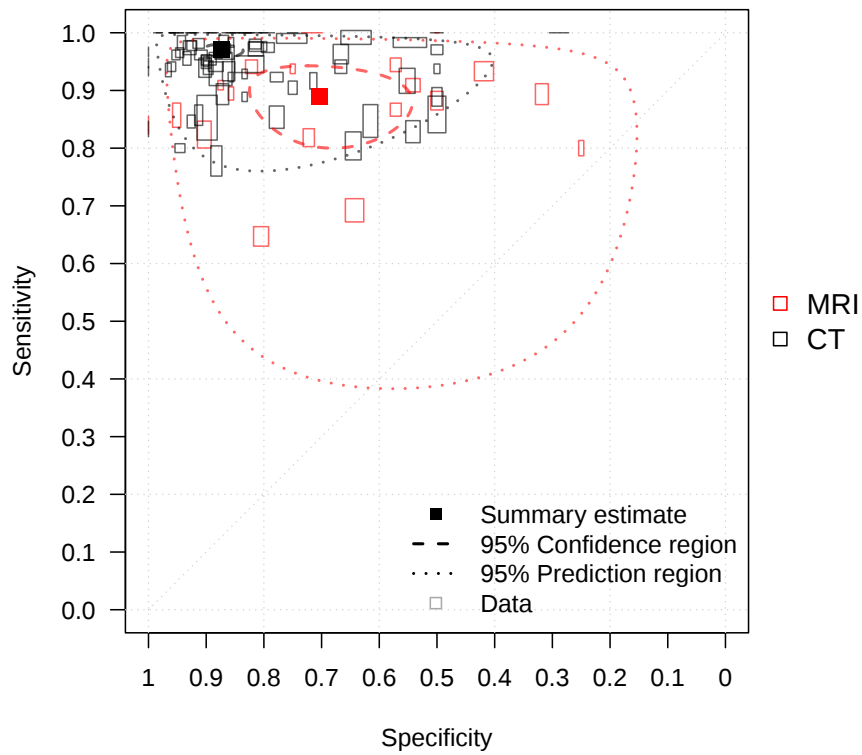
```

```

#> mu_B.MRI spec 0.70275873      0.95 0.58814813 0.79651017
#> mu_A.CT  sens 0.97019231      0.95 0.95988416 0.97791264
#> mu_B.CT  spec 0.87314631      0.95 0.84456145 0.89711484
#>
#> Recovered HSROC parameters (Rutter-Gatsonis)
#> -----
#>      Lambda      Theta      beta sigma2_alpha sigma2_theta
#> MRI 2.9548841 0.61764023 0.0015524244      2.0742124      0.33280676
#> CT  5.4130044 0.77893021 0.0015524244      2.0742124      0.33280676
#>
#> Subgroups
#> -----
#> [1] "MRI" "CT"
plot(schuetzreitsma,
     nudge_legend=-0.2,
     size="se",
     col=c("red","black"))

```

Diagnostic Test Accuracy Meta-Analysis



```
schuetzreitsma1 <- fitReitsmaSubgroup(data=schuetz,
                                     TP=TP,FP=FP,FN=FN,TN=TN,
                                     study=study,
                                     subgroup=test,
                                     variances="unequal")

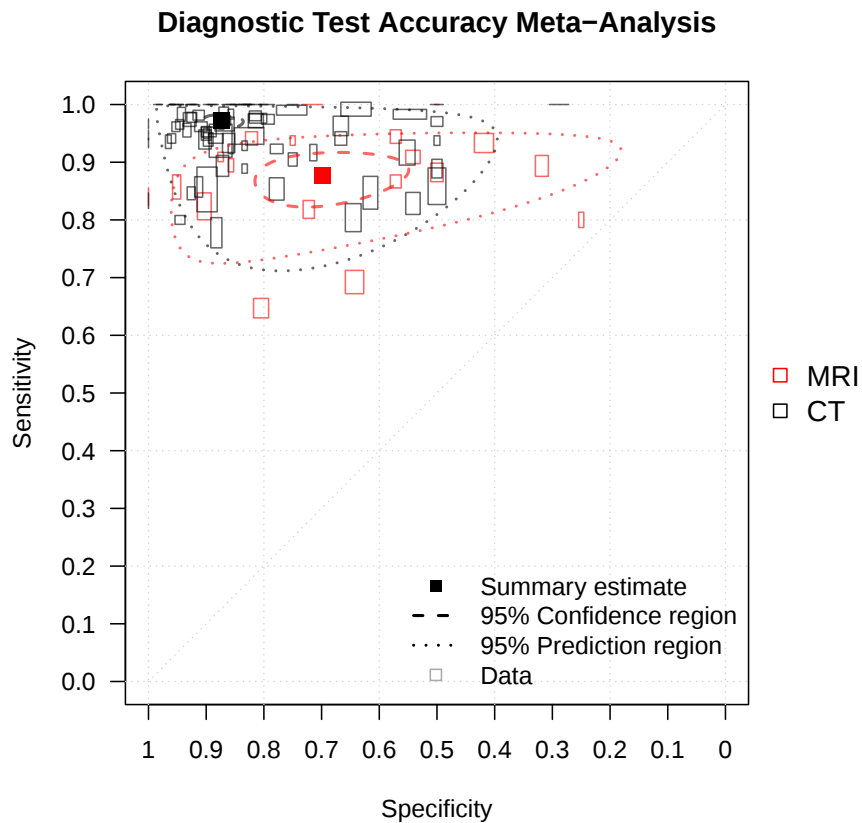
schuetzreitsma1
#>
#> Reitsma Subgroup Model
#> -----
#>
#> Number of studies      : 108
#> Number of subgroups   : 2
#> Optimizer              : Converged
#> Hessian                : Positive definite
#> Max |grad|            : 0.0002796148
#> -2 log likelihood     : 943.305 ( df = 10 )
```

```

#> AIC : 963.305
#> BIC : 990.127
#>
#> Use summary() for parameter estimates.
summary(schuetzreitsma1)
#>
#> Summary: Reitsma Subgroup Model
#> -----
#>
#> Parameter estimates
#> -----
#>
#> Estimate Std_Error
#> mu_A.MRI 1.96703630 0.16243311
#> mu_B.MRI 0.84101935 0.24063720
#> sigma2_A.MRI 0.11264053 0.16001255
#> sigma2_B.MRI 0.71212921 0.34247533
#> sigma_AB.MRI -0.14620447 0.13705725
#> mu_A.CT 3.55942950 0.17395085
#> mu_B.CT 1.93164500 0.12246536
#> sigma2_A.CT 1.10693220 0.30999211
#> sigma2_B.CT 0.88004119 0.19038771
#> sigma_AB.CT 0.30488900 0.17465369
#> nu_A.CT 1.59239120 0.23799912
#> nu_B.CT 1.09062400 0.27000679
#>
#> Sensitivity / specificity
#> -----
#>
#> type Estimate conflevel CI_Lower CI_Upper
#> mu_A.MRI sens 0.87729242 0.95 0.83871166 0.90766058
#> mu_B.MRI spec 0.69867986 0.95 0.59130894 0.78795786
#> mu_A.CT sens 0.97233223 0.95 0.96152427 0.98016683
#> mu_B.CT spec 0.87343139 0.95 0.84443674 0.89767671
#>
#> Recovered HSR0C parameters (Rutter-Gatsonis)
#> -----
#>
#> Lambda Theta beta sigma2_alpha sigma2_theta
#> MRI 3.6494770 1.29435500 0.92202888 0.27403476 0.21471316
#> CT 5.4067094 0.65770238 -0.11468947 2.58375460 0.34104965
#>
#> Subgroups
#> -----
#> [1] "MRI" "CT"

```

```
plot(schuetzreitsma1,
     nudge_legend=-0.2,
     size="se",
     col=c("red","black"))
```



```
schuetzreitsma2 <- fitReitsmaSubgroup(data=schuetz,
                                       TP=TP,FP=FP,FN=FN,TN=TN,
                                       study=study,
                                       subgroup=test,
                                       sensspec_constrain="sens")
schuetzreitsma3 <- fitReitsmaSubgroup(data=schuetz,
                                       TP=TP,FP=FP,FN=FN,TN=TN,
                                       study=study,
                                       subgroup=test,
                                       sensspec_constrain="spec")
anova(schuetzreitsma,schuetzreitsma1)
```

```

#>      Df logLik Df.diff Chisq Pr(>Chisq)
#> Model 1  7 -475.92
#> Model 2 10 -471.65      3 8.5381    0.03611 *
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
anova(schuetzreitsma2,schuetzreitsma)
#>      Df logLik Df.diff Chisq Pr(>Chisq)
#> Model 1  6 -485.79
#> Model 2  7 -475.92      1 19.745  8.848e-06 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
anova(schuetzreitsma3,schuetzreitsma)
#>      Df logLik Df.diff Chisq Pr(>Chisq)
#> Model 1  6 -482.64
#> Model 2  7 -475.92      1 13.432  0.0002474 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

What are the results for the direct comparisons?

```

schuetz2 <- subset(schuetz,indirect==0)
schuetzreitsma4 <- fitReitsmaSubgroup(data=schuetz2,
                                     TP=TP,FP=FP,FN=FN,TN=TN,
                                     study=study,
                                     subgroup=test,
                                     constrain="sigma2_A")

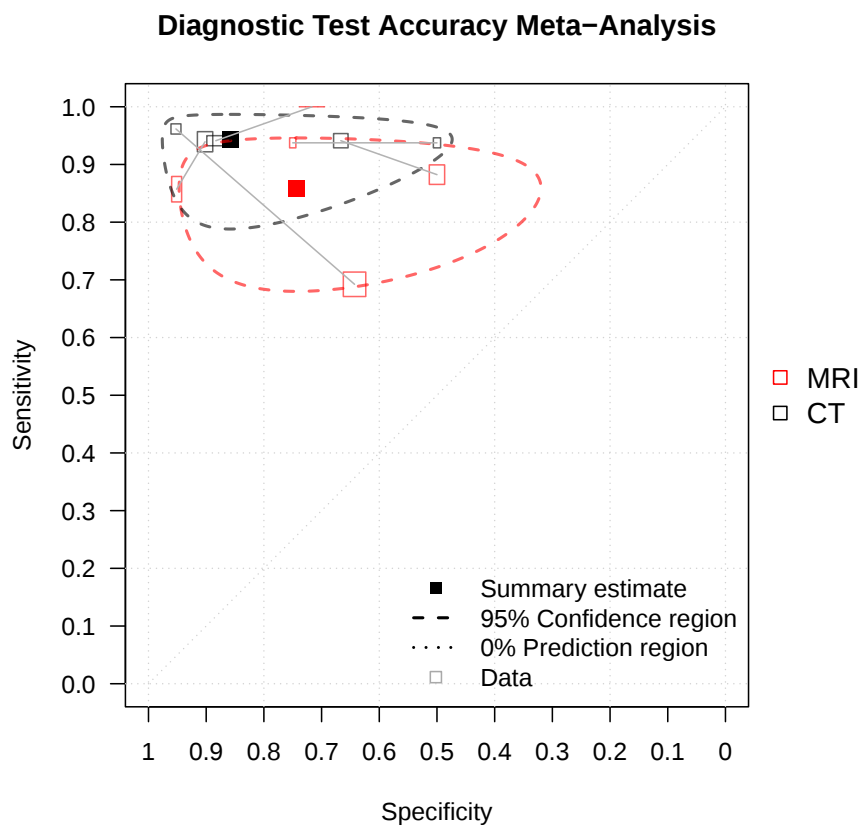
schuetzreitsma4
#>
#> Reitsma Subgroup Model
#> -----
#>
#> Number of studies      : 10
#> Number of subgroups   : 2
#> Optimizer              : Converged
#> Hessian                : Positive definite
#> Max |grad|            : 1.86672e-05
#> -2 log likelihood     : 88.734 ( df = 5 )
#> AIC                   : 98.734
#> BIC                   : 100.247
#>

```

```

#> Use summary() for parameter estimates.
summary(schuetzreitsma4)
#>
#> Summary: Reitsma Subgroup Model
#> -----
#>
#> Parameter estimates
#> -----
#>
#>               Estimate Std_Error
#> mu_A.MRI      1.8082893e+00 0.24124001
#> mu_B.MRI      1.0631770e+00 0.41604406
#> mu_A.CT       2.8134122e+00 0.34318787
#> mu_B.CT       1.8013974e+00 0.43628527
#> sigma2_A.sens 4.9303807e-32 0.00000000
#> sigma2_B.spec 5.8153371e-01 0.43286772
#> sigma_AB      0.0000000e+00 0.00000000
#> nu_A.CT       1.0051219e+00 0.41949312
#> nu_B.CT       7.3821679e-01 0.60559889
#>
#> Sensitivity / specificity
#> -----
#>
#>      type   Estimate conflevel   CI_Lower   CI_Upper
#> mu_A.MRI sens 0.85915500       0.95 0.79174360 0.90730054
#> mu_B.MRI spec 0.74329721       0.95 0.56162156 0.86745121
#> mu_A.CT  sens 0.94339630       0.95 0.89480371 0.97028818
#> mu_B.CT  spec 0.85831895       0.95 0.72036925 0.93441053
#>
#> Recovered HSROC parameters (Rutter-Gatsonis)
#> -----
#>
#>      Lambda   Theta   beta sigma2_alpha sigma2_theta
#> MRI 105972100 52986049 35.77261 3.3865514e-16 8.4663784e-17
#> CT  164875820 82437911 35.77261 3.3865514e-16 8.4663784e-17
#>
#> Subgroups
#> -----
#> [1] "MRI" "CT"
plot(schuetzreitsma4, predlevel=0.000001,
     nudge_legend=-0.2,
     size="se", scale=0.0025,
     connectstudies = TRUE,
     col=c("red", "black"))

```



References

- Reitsma, J. B., et al. (2005). Bivariate analysis of sensitivity and specificity produces informative summary measures in diagnostic reviews. *Journal of Clinical Epidemiology*, 58(10), 982–990.
- Rutter, C. M., & Gatsonis, C. A. (2001). A hierarchical regression approach to meta-analysis of diagnostic test accuracy evaluations. *Statistics in Medicine*, 20(19), 2865–2884.
- Harbord, R. M., Deeks, J. J., Egger, M., Whiting, P., & Sterne, J. A. C. (2007). A unification of models for meta-analysis of diagnostic accuracy studies. *Biostatistics*, 8(2), 239–251.
- Riley, R. D., Ensor, J., Jackson, D., & Burke, D. L. (2018). Deriving percentage study weights in multi-parameter meta-analysis models. *Statistical Methods in Medical Research*, 27(10), 2885–2905.
- Hoyer, A., Hirt, S., Kuss, O. (2018). Meta-analysis of full ROC curves using bivariate time-to-event models for interval-censored data. *Research Synthesis Methods*, 9(1), 62–72.

Deeks, J. J., Bossuyt, P. M., Leeflang, M. M., & Takwoingi, Y. (editors) (2023). Cochrane Handbook for Systematic Reviews of Diagnostic Test Accuracy. Version 2.0 (updated July 2023). *Cochrane*.