

Meta-Regression

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
#> Model fit              : Converged
#> -2 log likelihood      : 533.37 ( df = 7 )
#> AIC                    : 547.37
#> BIC                    : 558.646
```

```

#>
#>
#> Use summary() for parameter estimates.
summary(reitsmasub)
#> $estimates
#>
#>      Estimate Std_Error
#> mu_A.CCP1    -0.09653883 0.22032058
#> mu_B.CCP1     3.44671920 0.29824368
#> mu_A.CCP2     0.86603855 0.12087681
#> mu_B.CCP2     3.01649066 0.16223753
#> sigma2_A.sens 0.35982866 0.10217649
#> sigma2_B.spec 0.53990927 0.18015721
#> sigma_AB      -0.19684497 0.09835341
#> nu_A.CCP2     0.96257151 0.25134200
#> nu_B.CCP2     -0.43020982 0.33771185
#>
#> $sensspec
#>
#>      type      Orig conflevel  CI_Lower  CI_Upper
#> mu_A.CCP1 sens 0.4758840      0.95 0.3708997 0.5830439
#> mu_B.CCP1 spec 0.9691331      0.95 0.9459445 0.9825578
#> mu_A.CCP2 sens 0.7039207      0.95 0.6522909 0.7508130
#> mu_B.CCP2 spec 0.9533136      0.95 0.9369387 0.9655926
#>
#> $RutterGatsonis_recovered
#>
#>      Lambda      Theta      beta sigma2_alpha sigma2_theta
#> CCP1 3.007377 -1.6105343 0.2028866    0.4878424    0.3188056
#> CCP2 3.684000 -0.8834971 0.2028866    0.4878424    0.3188056
#>
#> $subgroups
#> [1] "CCP1" "CCP2"

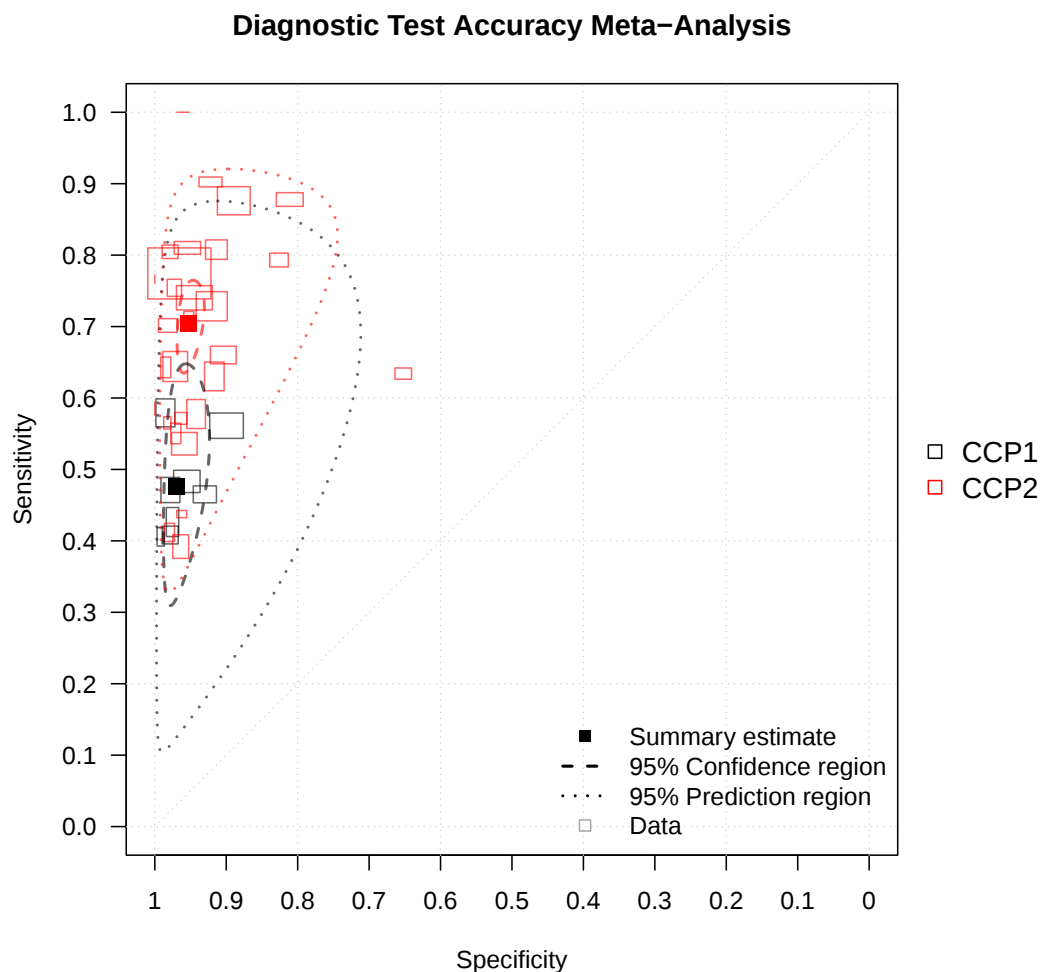
```

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"))

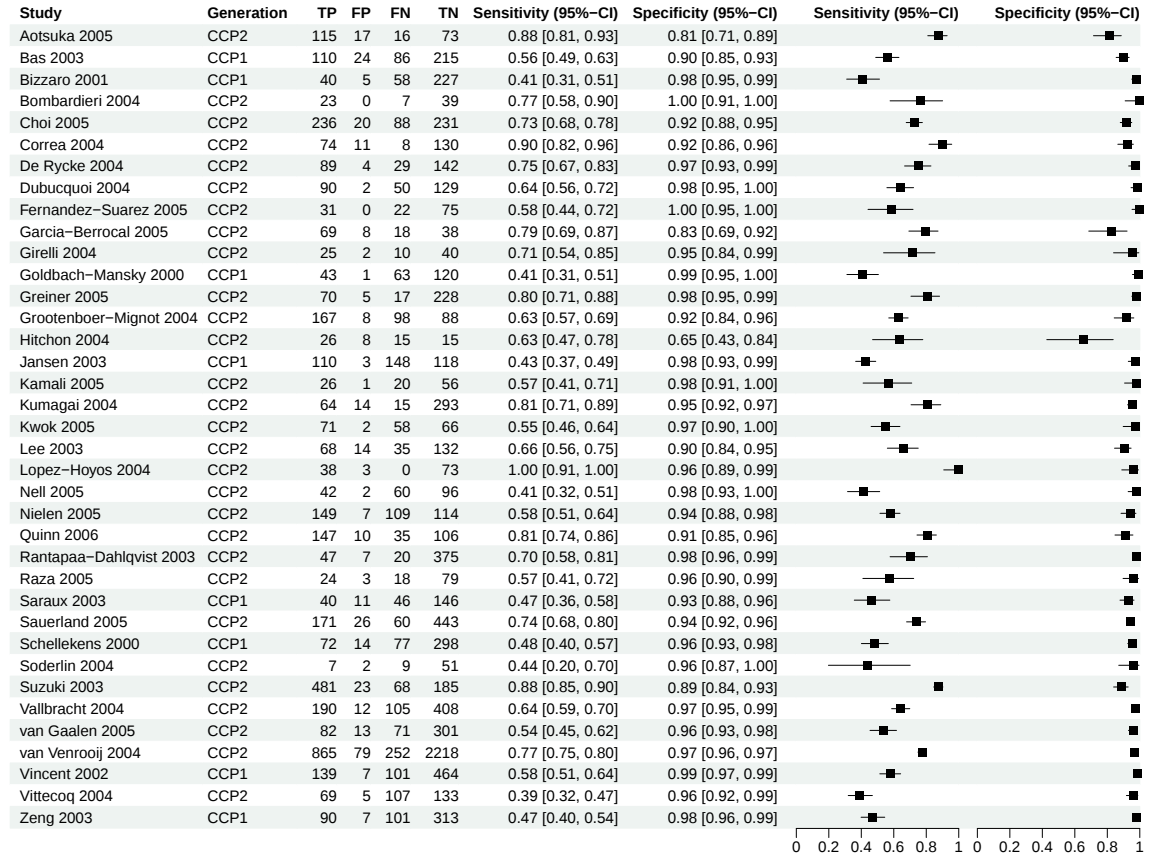
```



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)$estimates
#>               Estimate Std_Error
#> mu_A.CCP1      -5.439405e-02 0.05498530
#> mu_B.CCP1       3.446625e+00 0.29875482
#> mu_A.CCP2       9.179794e-01 0.03139172
#> mu_B.CCP2       2.996859e+00 0.16216233
#> sigma2_A.sens   4.930381e-32 0.00000000
#> sigma2_B.spec   5.396331e-01 0.17997640
#> sigma_AB        0.000000e+00 0.00000000
#> nu_A.CCP2       9.723735e-01 0.06331527
#> nu_B.CCP2      -4.497661e-01 0.33795382

```

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)$estimates
#>               Estimate Std_Error
#> mu_A.CCP1      0.65330520 0.1274076
#> mu_B.CCP1      3.08122030 0.3256178
#> mu_A.CCP2      0.65330520 0.0000000
#> mu_B.CCP2      3.11800628 0.1745128
#> sigma2_A.sens   0.54192926 0.1462735
#> sigma2_B.spec   0.57765267 0.1996838
#> sigma_AB       -0.27715832 0.1398143
#> nu_A.CCP2      0.00000000 0.0000000
#> nu_B.CCP2      0.03678907 0.3857733

```

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.

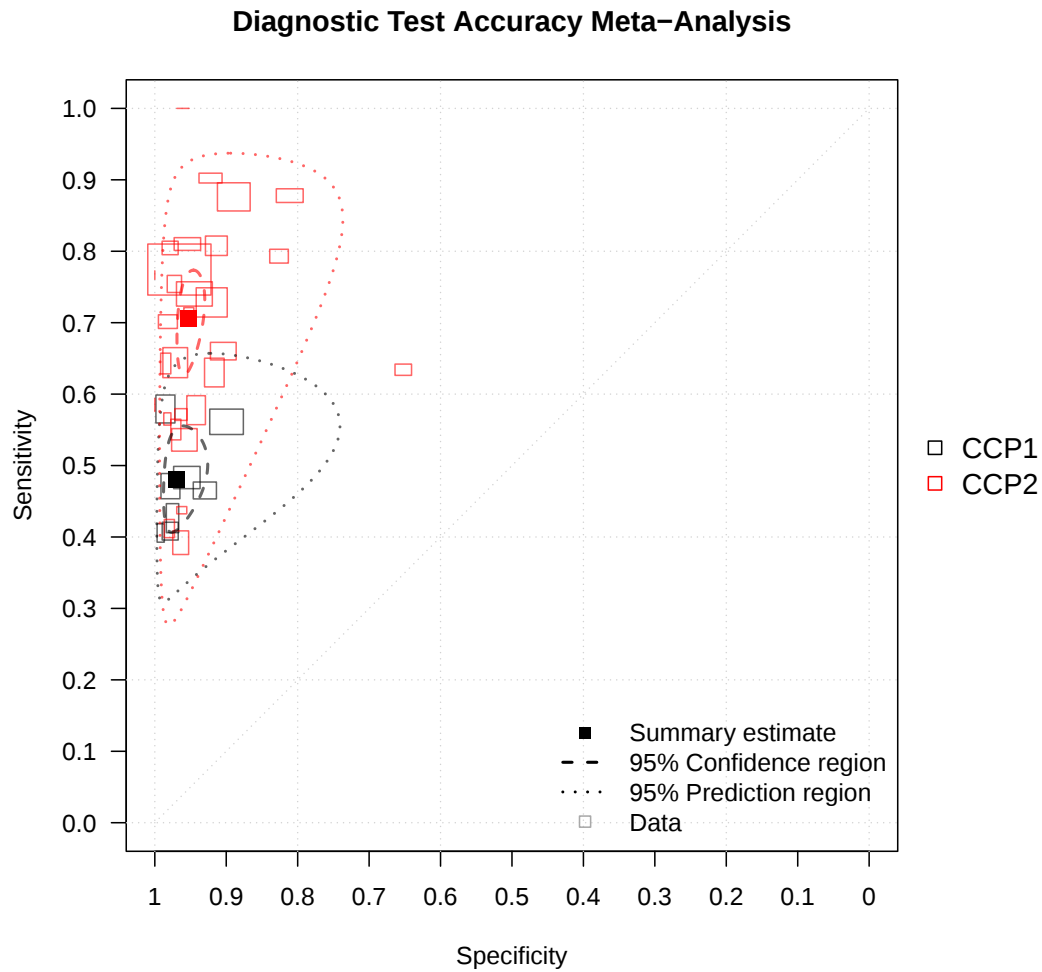
```
anova(constrainsens,reitsmasub)
#>           Df logLik Df.diff  Chisq Pr(>Chisq)
#> Model 1   6 -272.77
#> Model 2   7 -266.69       1 12.181  0.0004827 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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")

heteroskedastic
#>
#> Reitsma Subgroup Model
#> -----
#>
#> Number of studies      : 37
#> Number of subgroups    : 2
#> Model fit              : Converged
#> -2 log likelihood      : 524.839 ( df = 10 )
#> AIC                   : 544.839
#> BIC                   : 560.948
#>
#>
#> Use summary() for parameter estimates.
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
#> Model fit              : Converged
#> -2 log likelihood      : 753.722 ( df = 9 )
#> AIC                   : 771.722
#> BIC                   : 788.374
#>
#>
#> Use summary() for parameter estimates.
summary(ruttergatsonissub)
#> $estimates
#>
#>              Estimate Std. Error
#> Lambda_LA      2.45630480 0.32357688
#> xi_ELISA       0.24853313 0.43954106
#> xi_Nephelometry 0.33142795 0.44260055
#> Theta_LA      -0.55005215 0.21334028
#> gamma_ELISA    -0.19625487 0.26096194
#> gamma_Nephelometry 0.49585941 0.26223043
#> beta_LA        0.19768571 0.16983959
#> delta_ELISA    0.00000000 0.00000000
```

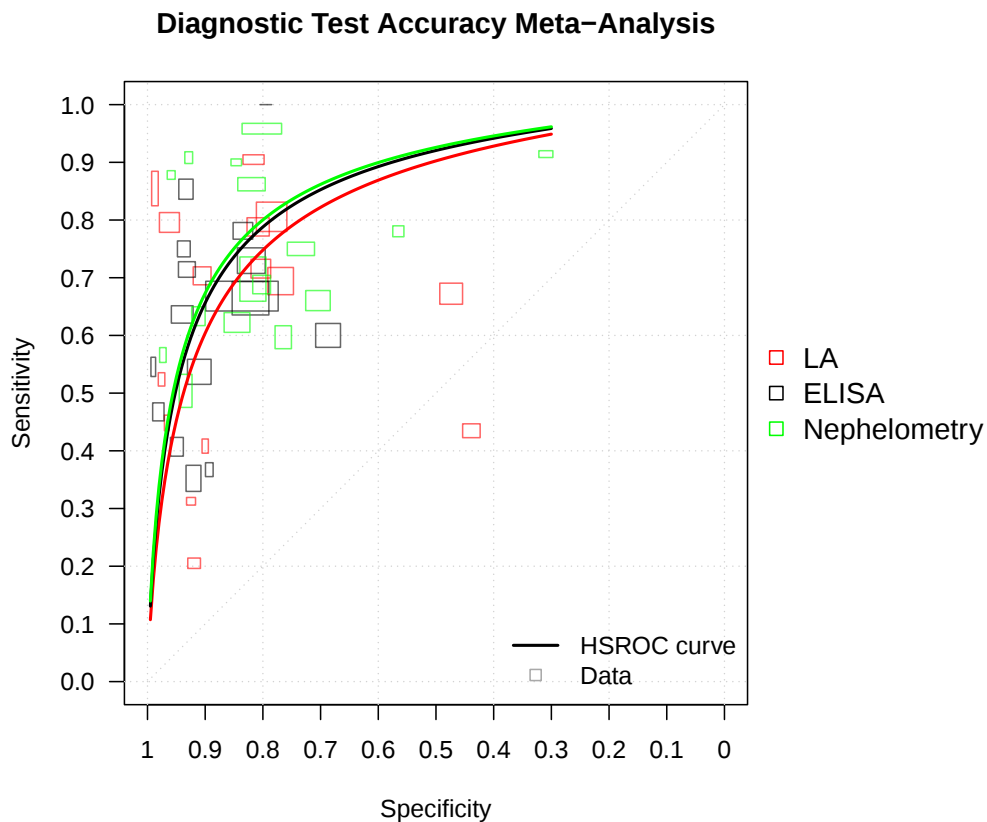
```

#> delta_Nephelometry 0.00000000 0.00000000
#> sigma2_alpha 1.27791453 0.30852473
#> sigma2_theta 0.47684025 0.11344515
#> Lambda_LA 2.45630480 0.32357688
#> Lambda_ELISA 2.70483793 0.32695409
#> Lambda_Nephelometry 2.78773275 0.30577909
#> Theta_LA -0.55005215 0.21334028
#> Theta_ELISA -0.74630703 0.20991299
#> Theta_Nephelometry -0.05419275 0.21213124
#> beta_LA 0.19768571 0.16983959
#> beta_ELISA 0.19768571 0.16983959
#> beta_Nephelometry 0.19768571 0.16983959
#> logitsens 0.82616625 0.28629916
#> logitsens 1.05130869 0.28606874
#> logitsens 1.12640188 0.27689464
#> sens 0.69554369 0.06062747
#> sens 0.74102613 0.05489842
#> sens 0.75517427 0.05119397
#>
#> $sensspec
#>      subgroup      spec conflevel logitsens Std_Error CI_Lower CI_Upper
#> 1      LA 0.8461538      0.95 0.8261663 0.2862992 0.2650302 1.387302
#> 2      ELISA 0.8461538      0.95 1.0513087 0.2860687 0.4906243 1.611993
#> 3 Nephelometry 0.8461538      0.95 1.1264019 0.2768946 0.5836984 1.669105
#>      Sens SensCI_Lower SensCI_Upper
#> 1 0.6955437 0.5658724 0.8001612
#> 2 0.7410261 0.6202535 0.8336879
#> 3 0.7551743 0.6419180 0.8414565
#>
#> $Reitsma_recovered
#>      mu_A.sens mu_B.spec sigma2_A.sens sigma2_B.spec sigma_AB
#> LA 0.6142809 1.962947 0.6534814 0.9703777 -0.1573616
#> ELISA 0.5490677 2.316769 0.6534814 0.9703777 -0.1573616
#> Nephelometry 1.2135903 1.598502 0.6534814 0.9703777 -0.1573616
#>
#> $subgroups
#> [1] "LA" "ELISA" "Nephelometry"

```

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

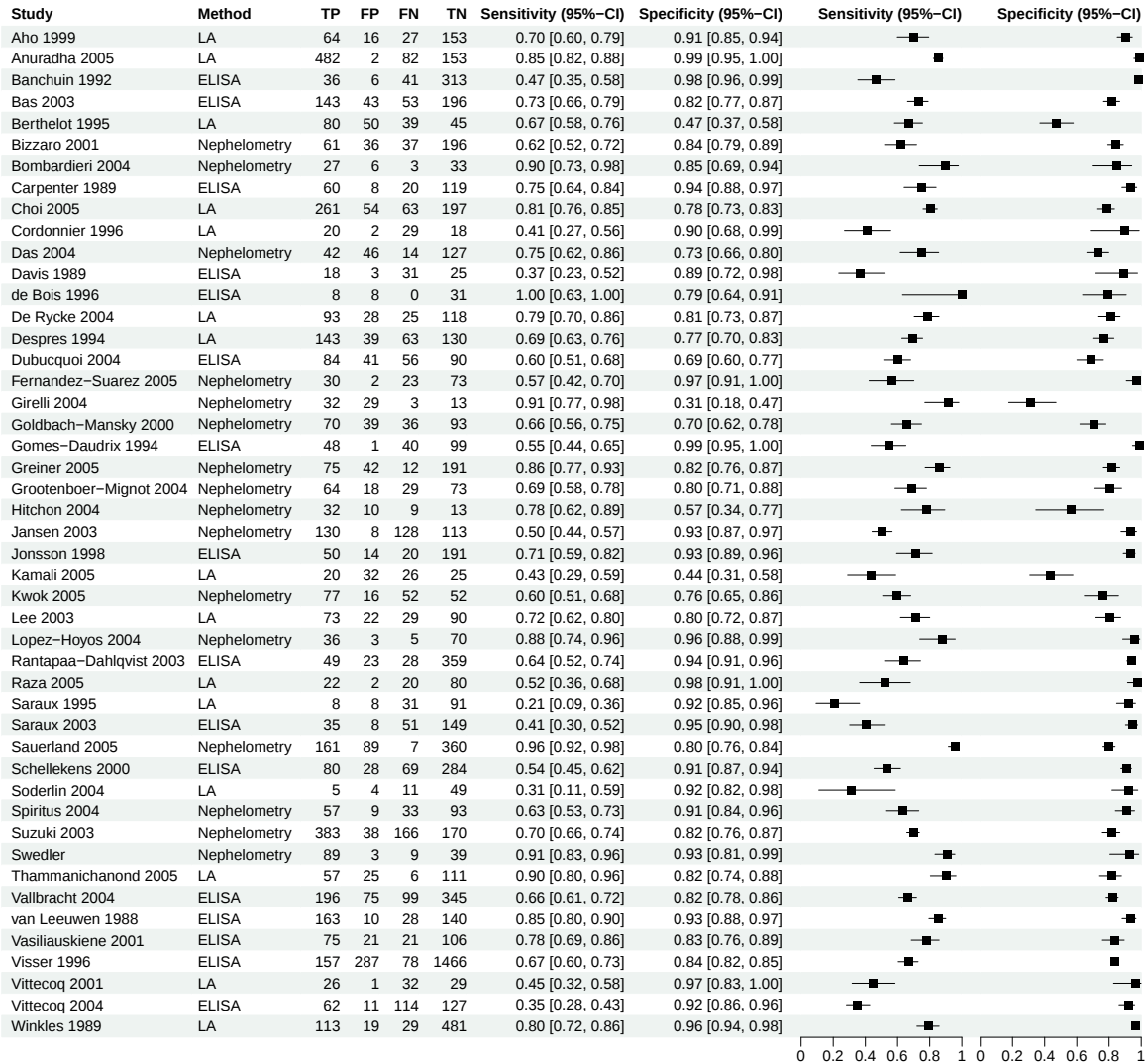
```
plot(ruttergatsonissub,  
     specrange=c(0.3,0.995),  
     size="se",  
     col=c("red","black","green"),  
     scale=0.015)
```



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
#> Model fit          : Converged
#> -2 log likelihood : 753.722 ( df = 9 )
#> AIC                : 771.722
#> BIC                : 788.374
#>
#>
#> Use summary() for parameter estimates.
summary(ruttergatsonisreg)
#> $estimates
#>
#>               Estimate Std. Error
#> accuracy_coef  2.45631156 0.32357673
#> accuracy_coef  0.24852208 0.43954066
#> accuracy_coef  0.33141584 0.44260014
#> threshold_coef -0.55005084 0.21334149
#> threshold_coef -0.19625009 0.26096393
#> threshold_coef  0.49585695 0.26223232
#> shape_coef     0.19768191 0.16983950
#> shape_coef     0.00000000 0.00000000
#> shape_coef     0.00000000 0.00000000
#> sigma2_alpha   1.27791200 0.30852400
#> sigma2_theta   0.47684805 0.11344753
#> Lambda_Pred    2.45631156 0.32357673
#> Lambda_Pred    2.70483365 0.32695391
#> Lambda_Pred    2.78772741 0.30577879
#> Theta_Pred     -0.55005084 0.21334149
#> Theta_Pred     -0.74630093 0.20991415
#> Theta_Pred     -0.05419388 0.21213237
#> beta_Pred      0.19768191 0.16983950
#> beta_Pred      0.19768191 0.16983950
#> beta_Pred      0.19768191 0.16983950
#> logitsens      0.82617129 0.28629952
#> logitsens      1.05130416 0.28606887
#> logitsens      1.12639652 0.27689490
#> sens           0.69554476 0.06062743
#> sens           0.74102525 0.05489857
#> sens           0.75517328 0.05119416
#>
#> $sensspec
#>
#>      spec conflevel logitsens Std_Error  CI_Lower CI_Upper      Sens
#> 1 0.8461538      0.95 0.8261713 0.2862995 0.2650345 1.387308 0.6955448

```

```
#> 2 0.8461538      0.95 1.0513042 0.2860689 0.4906195 1.611989 0.7410253
#> 3 0.8461538      0.95 1.1263965 0.2768949 0.5836925 1.669101 0.7551733
#>   SensCI_Lower SensCI_Upper
#> 1      0.5658735      0.8001621
#> 2      0.6202524      0.8336873
#> 3      0.6419166      0.8414559
```

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.

```
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
#> Model fit              : Converged
#> -2 log likelihood      : 753.553 ( df = 11 )
#> AIC                   : 775.553
#> BIC                   : 795.905
#>
#>
#> Use summary() for parameter estimates.
summary(ruttergatsonissubfull)
#> $estimates
#>               Estimate Std. Error
#> Lambda_LA      2.4243669 0.33049224
#> xi_ELISA       0.2974586 0.51315391
```

```

#> xi_Nephelometry      0.3716007 0.45086693
#> Theta_LA            -0.5029485 0.24451659
#> gamma_ELISA         -0.2578243 0.37024673
#> gamma_Nephelometry   0.3970655 0.35944296
#> beta_LA              0.2777490 0.26642342
#> delta_ELISA          -0.1028886 0.42661220
#> delta_Nephelometry  -0.1603830 0.39753787
#> sigma2_alpha         1.2730676 0.30798946
#> sigma2_theta         0.4762395 0.11343564
#> Lambda_LA           2.4243669 0.33049224
#> Lambda_ELISA         2.7218254 0.39299597
#> Lambda_Nephelometry  2.7959676 0.30708174
#> Theta_LA            -0.5029485 0.24451659
#> Theta_ELISA          -0.7607728 0.27817184
#> Theta_Nephelometry  -0.1058830 0.26322841
#> beta_LA              0.2777490 0.26642342
#> beta_ELISA           0.1748604 0.33321542
#> beta_Nephelometry    0.1173660 0.29500321
#> logitsens            0.8186924 0.27519155
#> logitsens            1.0626994 0.32405871
#> logitsens            1.1206496 0.28834985
#> sens                  0.6939587 0.05844519
#> sens                  0.7432061 0.06184687
#> sens                  0.7541092 0.05346829
#>
#> $sensspec
#>      subgroup      spec conflevel logitsens Std_Error CI_Lower CI_Upper
#> 1          LA 0.8461538      0.95 0.8186924 0.2751916 0.2793269 1.358058
#> 2          ELISA 0.8461538      0.95 1.0626994 0.3240587 0.4275560 1.697843
#> 3 Nephelometry 0.8461538      0.95 1.1206496 0.2883499 0.5554942 1.685805
#>      Sens SensCI_Lower SensCI_Upper
#> 1 0.6939587      0.5693812      0.7954439
#> 2 0.7432061      0.6052899      0.8452528
#> 3 0.7541092      0.6354094      0.8436717
#>
#> $Reitsma_recovered
#>      mu_A.sens mu_B.spec sigma2_A.sens sigma2_B.spec sigma_AB
#> LA          0.6172735 1.970652      0.6018282      1.0488714 -0.1579725
#> ELISA        0.5498979 2.315536      0.6670470      0.9463206 -0.1579725
#> Nephelometry 1.2184583 1.594759      0.7065224      0.8934470 -0.1579725
#>
#> $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
```

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

The test below formally investigates whether the HSROC curve shapes 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
```

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)

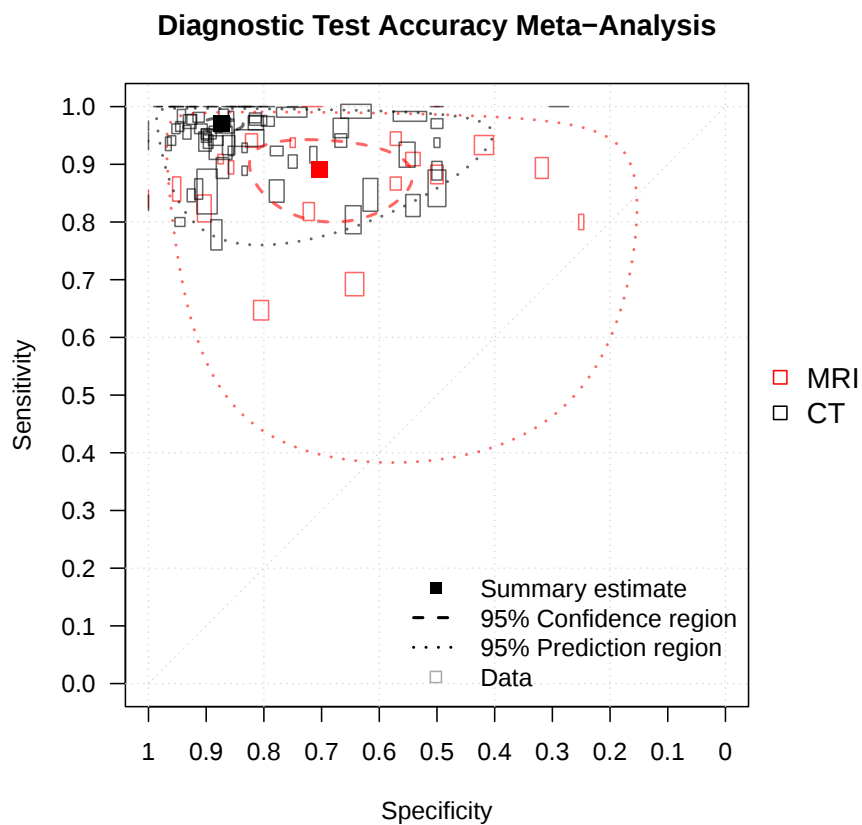
schuetzreitsma
#>
#> Reitsma Subgroup Model
#> -----
#>
#> Number of studies      : 108
#> Number of subgroups   : 2
#> Model fit              : Converged
#> -2 log likelihood      : 951.843 ( df = 7 )
#> AIC                    : 965.843
#> BIC                    : 984.618
#>
#>
#> Use summary() for parameter estimates.
summary(schuetzreitsma)
#> $estimates
#>
#>      Estimate Std_Error
#> mu_A.MRI      2.0934567 0.2639212
#> mu_B.MRI      0.8604695 0.2572264
#> mu_A.CT       3.4827280 0.1569859
#> mu_B.CT       1.9290688 0.1206664
#> sigma2_A.sens 0.8500392 0.2195458
#> sigma2_B.spec 0.8526826 0.1683378
#> sigma_AB      0.1857463 0.1342067
#> nu_A.CT       1.3893006 0.3015220
#> nu_B.CT       1.0686026 0.2834029
#>
#> $sensspec
#>
#>      type      Orig conflevel  CI_Lower  CI_Upper
#> mu_A.MRI sens 0.8902656      0.95 0.8286629 0.9315491
#> mu_B.MRI spec 0.7027587      0.95 0.5881481 0.7965102

```

```

#> mu_A.CT    sens 0.9701923      0.95 0.9598842 0.9779126
#> mu_B.CT    spec 0.8731463      0.95 0.8445615 0.8971148
#>
#> $RutterGatsonis_recovered
#>      Lambda      Theta      beta sigma2_alpha sigma2_theta
#> MRI 2.954884 0.6176402 0.001552424    2.074212    0.3328068
#> CT  5.413004 0.7789302 0.001552424    2.074212    0.3328068
#>
#> $subgroups
#> [1] "MRI" "CT"
plot(schuetzreitsma,
     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(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?

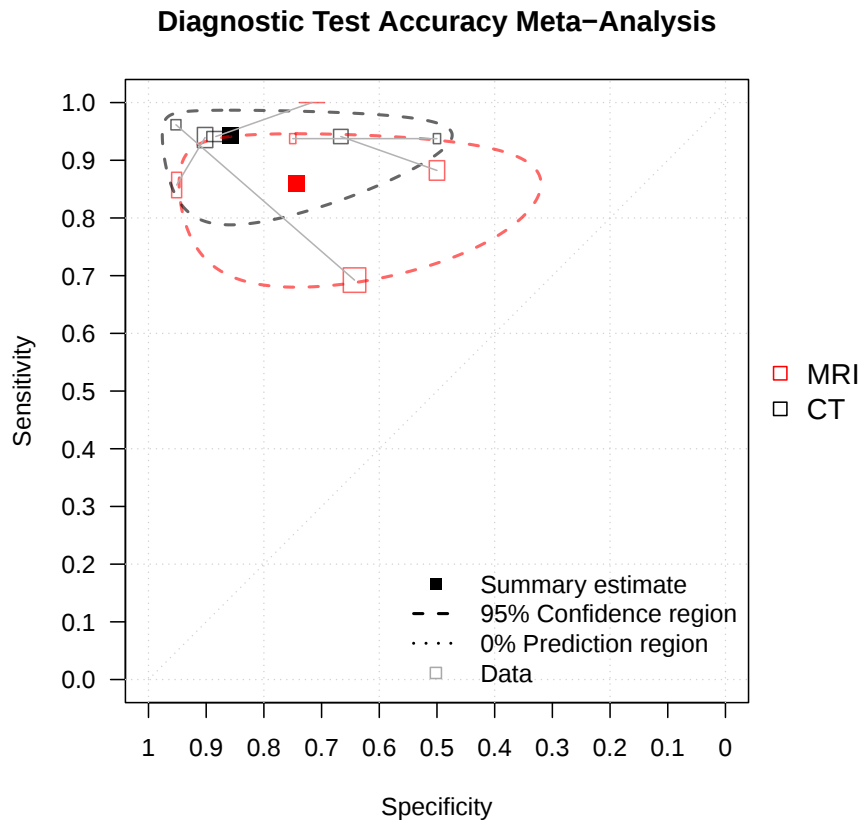
```

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

round(summary(schuetzreitsma4)$estimates,5)
#>           Estimate Std_Error
#> mu_A.MRI      1.80829   0.24124
#> mu_B.MRI      1.06318   0.41604
#> mu_A.CT       2.81341   0.34319
#> mu_B.CT       1.80140   0.43629
#> sigma2_A.sens  0.00000   0.00000
#> sigma2_B.spec  0.58153   0.43287
#> sigma_AB      0.00000   0.00000

```

```
#> nu_A.CT      1.00512    0.41949
#> nu_B.CT      0.73822    0.60560
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*.