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

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

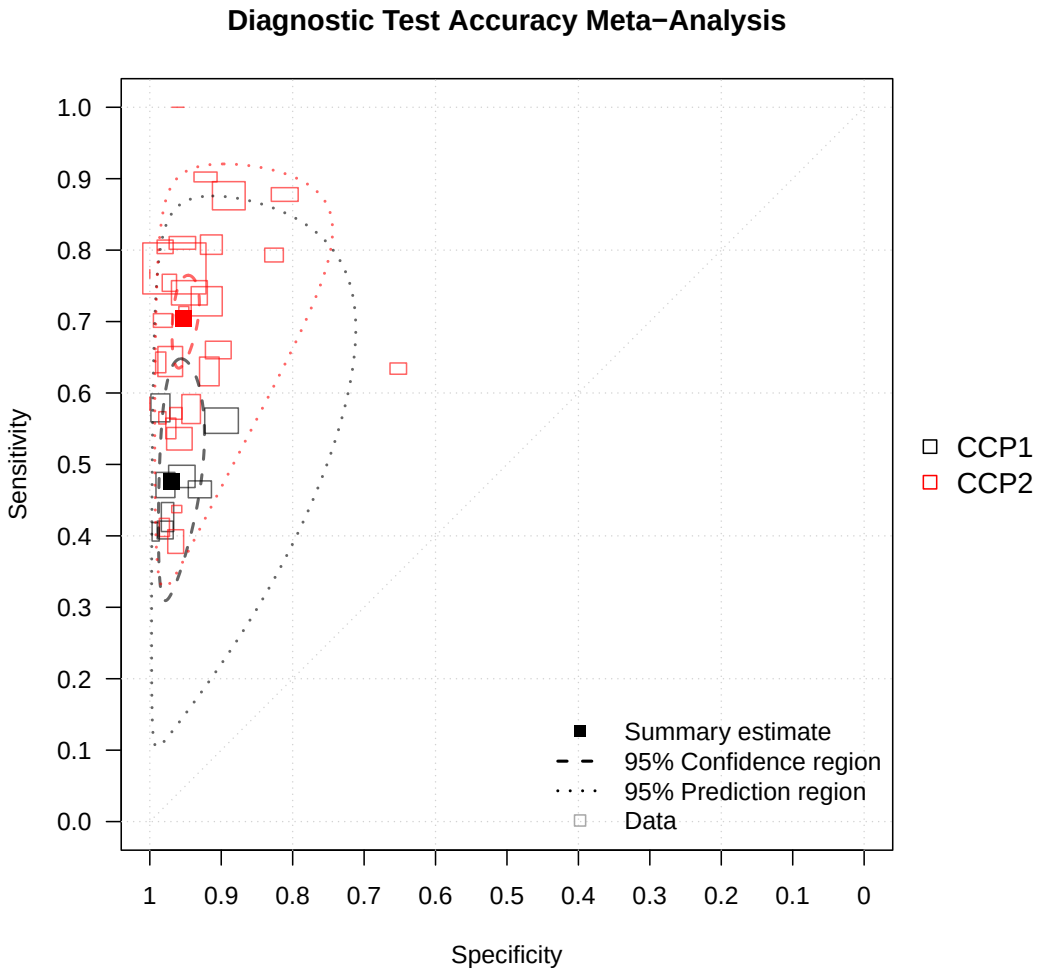
$$z_i^\top \mu_{A} = \begin{cases} \mu_{A} & \text{for subgroup 1,} \\ \mu_{A} + \nu_{A2} & \text{for subgroup 2,} \end{cases} \qquad z_i^\top \mu_{B} = \begin{cases} \mu_{B} & \text{for subgroup 1,} \\ \mu_{B} + \nu_{B2} & \text{for subgroup 2.} \end{cases}$$

```
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)
#> $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  Estimate 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"))
```

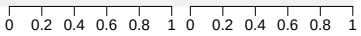


How do I get a coupled forest plot?

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")
```

Study	Generation	TP	FP	FN	TN	Sensitivity (95%−CI)	Specificity (95%−CI)
Aotsuka 2005	CCP2	115	17	16	73	0.88 [0.81, 0.93]	0.81 [0.71, 0.89]
Bas 2003	CCP1	110	24	86	215	0.56 [0.49, 0.63]	0.90 [0.85, 0.93]
Bizzaro 2001	CCP1	40	5	58	227	0.41 [0.31, 0.51]	0.98 [0.95, 0.99]
Bombardieri 2004	CCP2	23	0	7	39	0.77 [0.58, 0.90]	1.00 [0.91, 1.00]
Choi 2005	CCP2	236	20	88	231	0.73 [0.68, 0.78]	0.92 [0.88, 0.95]
Correa 2004	CCP2	74	11	8	130	0.90 [0.82, 0.96]	0.92 [0.86, 0.96]
De Rycke 2004	CCP2	89	4	29	142	0.75 [0.67, 0.83]	0.97 [0.93, 0.99]
Dubucquoi 2004	CCP2	90	2	50	129	0.64 [0.56, 0.72]	0.98 [0.95, 1.00]
Fernandez−Suarez 2005	CCP2	31	0	22	75	0.58 [0.44, 0.72]	1.00 [0.95, 1.00]
Garcia−Berrocal 2005	CCP2	69	8	18	38	0.79 [0.69, 0.87]	0.83 [0.69, 0.92]
Girelli 2004	CCP2	25	2	10	40	0.71 [0.54, 0.85]	0.95 [0.84, 0.99]
Goldbach−Mansky 2000	CCP1	43	1	63	120	0.41 [0.31, 0.51]	0.99 [0.95, 1.00]
Greiner 2005	CCP2	70	5	17	228	0.80 [0.71, 0.88]	0.98 [0.95, 0.99]
Grootenboer−Mignot 2004	CCP2	167	8	98	88	0.63 [0.57, 0.69]	0.92 [0.84, 0.96]
Hitchon 2004	CCP2	26	8	15	15	0.63 [0.47, 0.78]	0.65 [0.43, 0.84]
Jansen 2003	CCP1	110	3	148	118	0.43 [0.37, 0.49]	0.98 [0.93, 0.99]
Kamali 2005	CCP2	26	1	20	56	0.57 [0.41, 0.71]	0.98 [0.91, 1.00]
Kumagai 2004	CCP2	64	14	15	293	0.81 [0.71, 0.89]	0.95 [0.92, 0.97]
Kwok 2005	CCP2	71	2	58	66	0.55 [0.46, 0.64]	0.97 [0.90, 1.00]
Lee 2003	CCP2	68	14	35	132	0.66 [0.56, 0.75]	0.90 [0.84, 0.95]
Lopez−Hoyos 2004	CCP2	38	3	0	73	1.00 [0.91, 1.00]	0.96 [0.89, 0.99]
Nell 2005	CCP2	42	2	60	96	0.41 [0.32, 0.51]	0.98 [0.93, 1.00]
Nielen 2005	CCP2	149	7	109	114	0.58 [0.51, 0.64]	0.94 [0.88, 0.98]
Quinn 2006	CCP2	147	10	35	106	0.81 [0.74, 0.86]	0.91 [0.85, 0.96]
Rantapaa−Dahlqvist 2003	CCP2	47	7	20	375	0.70 [0.58, 0.81]	0.98 [0.96, 0.99]
Raza 2005	CCP2	24	3	18	79	0.57 [0.41, 0.72]	0.96 [0.90, 0.99]
Saraux 2003	CCP1	40	11	46	146	0.47 [0.36, 0.58]	0.93 [0.88, 0.96]
Sauerland 2005	CCP2	171	26	60	443	0.74 [0.68, 0.80]	0.94 [0.92, 0.96]
Schellekens 2000	CCP1	72	14	77	298	0.48 [0.40, 0.57]	0.96 [0.93, 0.98]
Soderlin 2004	CCP2	7	2	9	51	0.44 [0.20, 0.70]	0.96 [0.87, 1.00]
Suzuki 2003	CCP2	481	23	68	185	0.88 [0.85, 0.90]	0.89 [0.84, 0.93]
Vallbracht 2004	CCP2	190	12	105	408	0.64 [0.59, 0.70]	0.97 [0.95, 0.99]
van Gaalen 2005	CCP2	82	13	71	301	0.54 [0.45, 0.62]	0.96 [0.93, 0.98]
van Venrooij 2004	CCP2	865	79	252	2218	0.77 [0.75, 0.80]	0.97 [0.96, 0.97]
Vincent 2002	CCP1	139	7	101	464	0.58 [0.51, 0.64]	0.99 [0.97, 0.99]
Vittecoq 2004	CCP2	69	5	107	133	0.39 [0.32, 0.47]	0.96 [0.92, 0.99]
Zeng 2003	CCP1	90	7	101	313	0.47 [0.40, 0.54]	0.98 [0.96, 0.99]



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")`.

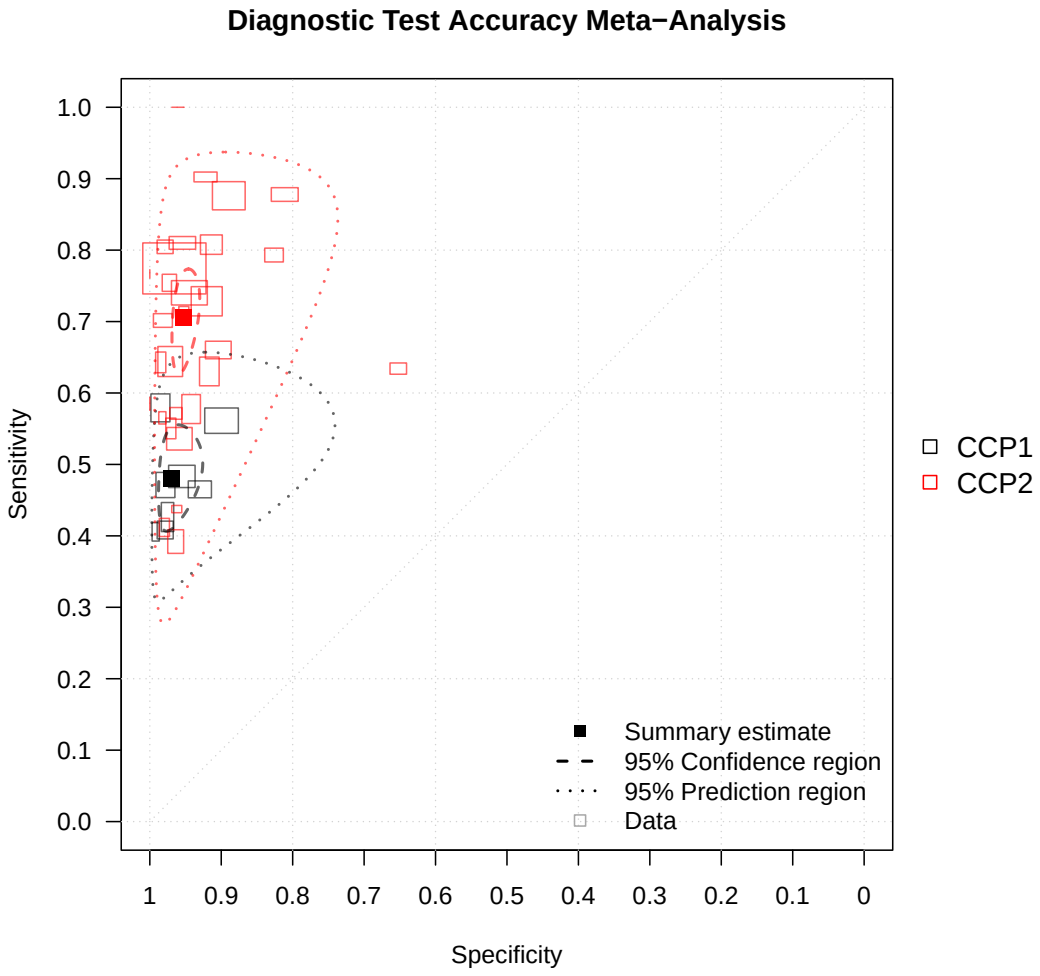
How can I compare `constrainsens` with the full model?

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")
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

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

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

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

$$z_i^\top \theta_i = \begin{cases} \theta_i & \text{for subgroup 1,} \\ \theta_i + \gamma_2 & \text{for subgroup 2,} \\ \theta_i + \gamma_3 & \text{for subgroup 3,} \end{cases} \qquad z_i^\top \alpha_i = \begin{cases} \alpha_i & \text{for subgroup 1,} \\ \alpha_i + \xi_2 & \text{for subgroup 2,} \\ \alpha_i + \xi_3 & \text{for subgroup 3,} \end{cases}$$

$$z_i^\top \beta = \begin{cases} \beta & \text{for subgroup 1,} \\ \beta + \delta_2 & \text{for subgroup 2,} \\ \beta + \delta_3 & \text{for subgroup 3,} \end{cases}$$

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

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?

```r
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
```

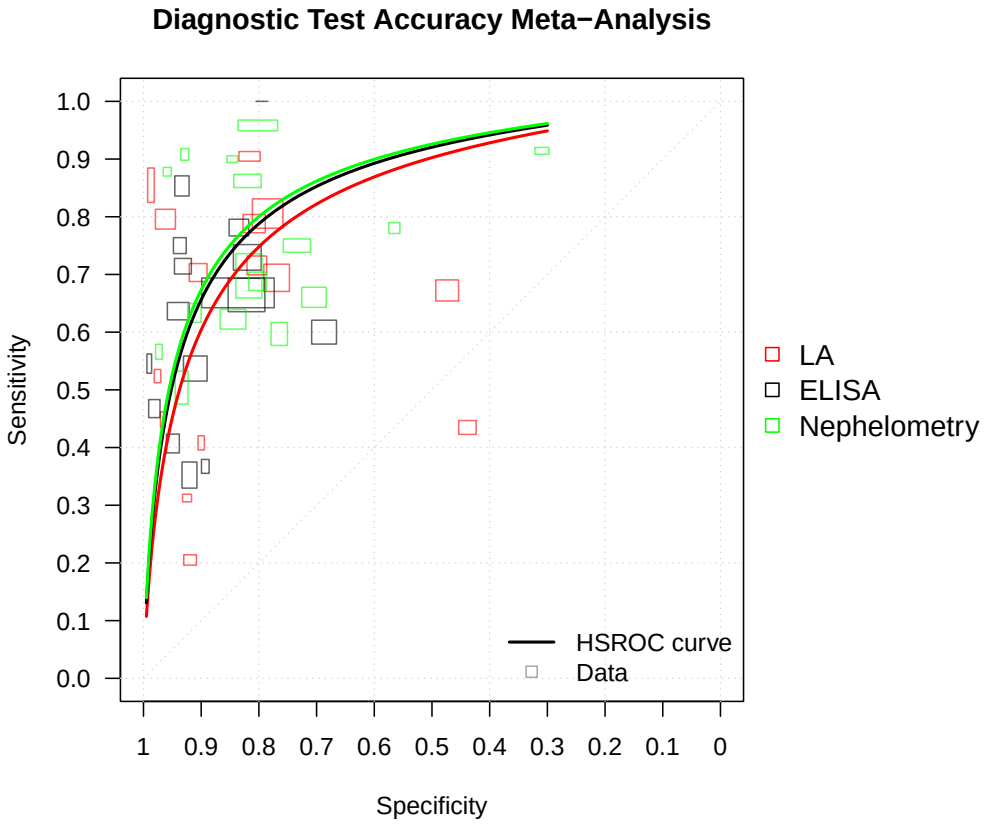
```r
ruttergatsonissub
#>
#> Rutter & Gatsonis Subgroup Model
#> --------------------------------
#>
#> Number of studies   : 47
#> Number of subgroups : 3
#> Optimizer           : Converged
#> Hessian             : Positive definite
#> Max |grad|          : 0.0001105815
#> -2 log likelihood   : 753.722 ( df = 9 )
#> AIC                 : 771.722
#> BIC                 : 788.374
#>
#> Use summary() for parameter estimates.
summary(ruttergatsonissub)
#> $estimates
#>                         Estimate Std. Error
#> Lambda_LA            2.45630477 0.32357688
#> xi_ELISA             0.24853315 0.43954107
#> xi_Nephelometry      0.33142788 0.44260056
#> Theta_LA            -0.55005213 0.21334028
#> gamma_ELISA         -0.19625486 0.26096194
#> gamma_Nephelometry   0.49585945 0.26223043
#> beta_LA              0.19768573 0.16983959
#> delta_ELISA          0.00000000 0.00000000
#> delta_Nephelometry   0.00000000 0.00000000
#> sigma2_alpha         1.27791458 0.30852474
```

```
#> sigma2_theta          0.47684025 0.11344515
#> Lambda_LA             2.45630477 0.32357688
#> Lambda_ELISA          2.70483792 0.32695410
#> Lambda_Nephelometry   2.78773265 0.30577909
#> Theta_LA             -0.55005213 0.21334028
#> Theta_ELISA          -0.74630699 0.20991299
#> Theta_Nephelometry   -0.05419268 0.21213124
#> beta_LA               0.19768573 0.16983959
#> beta_ELISA            0.19768573 0.16983959
#> beta_Nephelometry     0.19768573 0.16983959
#> logitsens             0.82616623 0.28629916
#> logitsens             1.05130869 0.28606874
#> logitsens             1.12640178 0.27689464
#> sens                  0.69554369 0.06062747
#> sens                  0.74102612 0.05489842
#> sens                  0.75517425 0.05119397
#>
#> $sensspec
#>       subgroup      spec conflevel logitsens Std_Error  CI_Lower CI_Upper
#> 1           LA 0.8461538      0.95 0.8261662 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 1.1264018 0.2768946 0.5836983 1.669105
#>        Sens SensCI_Lower SensCI_Upper
#> 1 0.6955437    0.5658724    0.8001612
#> 2 0.7410261    0.6202535    0.8336879
#> 3 0.7551742    0.6419179    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     0.9703778 -0.1573616
#> ELISA        0.5490678  2.316769     0.6534814     0.9703778 -0.1573616
#> Nephelometry 1.2135903  1.598502     0.6534814     0.9703778 -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)
```

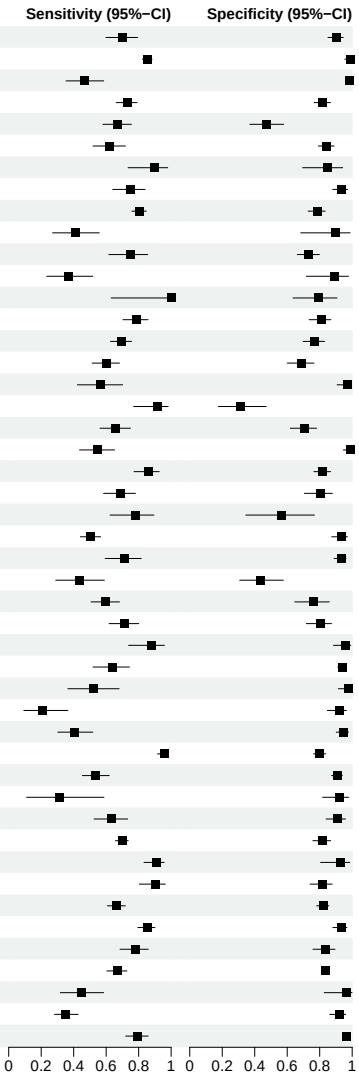


How do I get a coupled forest plot?

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")
```

Study	Method	TP	FP	FN	TN	Sensitivity (95%−CI)	Specificity (95%−CI)
Aho 1999	LA	64	16	27	153	0.70 [0.60, 0.79]	0.91 [0.85, 0.94]
Anuradha 2005	LA	482	2	82	153	0.85 [0.82, 0.88]	0.99 [0.95, 1.00]
Banchuin 1992	ELISA	36	6	41	313	0.47 [0.35, 0.58]	0.98 [0.96, 0.99]
Bas 2003	ELISA	143	43	53	196	0.73 [0.66, 0.79]	0.82 [0.77, 0.87]
Berthelot 1995	LA	80	50	39	45	0.67 [0.58, 0.76]	0.47 [0.37, 0.58]
Bizzaro 2001	Nephelometry	61	36	37	196	0.62 [0.52, 0.72]	0.84 [0.79, 0.89]
Bombardieri 2004	Nephelometry	27	6	3	33	0.90 [0.73, 0.98]	0.85 [0.69, 0.94]
Carpenter 1989	ELISA	60	8	20	119	0.75 [0.64, 0.84]	0.94 [0.88, 0.97]
Choi 2005	LA	261	54	63	197	0.81 [0.76, 0.85]	0.78 [0.73, 0.83]
Cordonnier 1996	LA	20	2	29	18	0.41 [0.27, 0.56]	0.90 [0.68, 0.99]
Das 2004	Nephelometry	42	46	14	127	0.75 [0.62, 0.86]	0.73 [0.66, 0.80]
Davis 1989	ELISA	18	3	31	25	0.37 [0.23, 0.52]	0.89 [0.72, 0.98]
de Bois 1996	ELISA	8	8	0	31	1.00 [0.63, 1.00]	0.79 [0.64, 0.91]
De Rycke 2004	LA	93	28	25	118	0.79 [0.70, 0.86]	0.81 [0.73, 0.87]
Despres 1994	LA	143	39	63	130	0.69 [0.63, 0.76]	0.77 [0.70, 0.83]
Dubucquoi 2004	ELISA	84	41	56	90	0.60 [0.51, 0.68]	0.69 [0.60, 0.77]
Fernandez−Suarez 2005	Nephelometry	30	2	23	73	0.57 [0.42, 0.70]	0.97 [0.91, 1.00]
Girelli 2004	Nephelometry	32	29	3	13	0.91 [0.77, 0.98]	0.31 [0.18, 0.47]
Goldbach−Mansky 2000	Nephelometry	70	39	36	93	0.66 [0.56, 0.75]	0.70 [0.62, 0.78]
Gomes−Daudrix 1994	ELISA	48	1	40	99	0.55 [0.44, 0.65]	0.99 [0.95, 1.00]
Greiner 2005	Nephelometry	75	42	12	191	0.86 [0.77, 0.93]	0.82 [0.76, 0.87]
Grootenboer−Mignot 2004	Nephelometry	64	18	29	73	0.69 [0.58, 0.78]	0.80 [0.71, 0.88]
Hitchon 2004	Nephelometry	32	10	9	13	0.78 [0.62, 0.89]	0.57 [0.34, 0.77]
Jansen 2003	Nephelometry	130	8	128	113	0.50 [0.44, 0.57]	0.93 [0.87, 0.97]
Jonsson 1998	ELISA	50	14	20	191	0.71 [0.59, 0.82]	0.93 [0.89, 0.96]
Kamali 2005	LA	20	32	26	25	0.43 [0.29, 0.59]	0.44 [0.31, 0.58]
Kwok 2005	Nephelometry	77	16	52	52	0.60 [0.51, 0.68]	0.76 [0.65, 0.86]
Lee 2003	LA	73	22	29	90	0.72 [0.62, 0.80]	0.80 [0.72, 0.87]
Lopez−Hoyos 2004	Nephelometry	36	3	5	70	0.88 [0.74, 0.96]	0.96 [0.88, 0.99]
Rantapaa−Dahlqvist 2003	ELISA	49	23	28	359	0.64 [0.52, 0.74]	0.94 [0.91, 0.96]
Raza 2005	LA	22	2	20	80	0.52 [0.36, 0.68]	0.98 [0.91, 1.00]
Saraux 1995	LA	8	8	31	91	0.21 [0.09, 0.36]	0.92 [0.85, 0.96]
Saraux 2003	ELISA	35	8	51	149	0.41 [0.30, 0.52]	0.95 [0.90, 0.98]
Sauerland 2005	Nephelometry	161	89	7	360	0.96 [0.92, 0.98]	0.80 [0.76, 0.84]
Schellekens 2000	ELISA	80	28	69	284	0.54 [0.45, 0.62]	0.91 [0.87, 0.94]
Soderlin 2004	LA	5	4	11	49	0.31 [0.11, 0.59]	0.92 [0.82, 0.98]
Spiritus 2004	Nephelometry	57	9	33	93	0.63 [0.53, 0.73]	0.91 [0.84, 0.96]
Suzuki 2003	Nephelometry	383	38	166	170	0.70 [0.66, 0.74]	0.82 [0.76, 0.87]
Swedler	Nephelometry	89	3	9	39	0.91 [0.83, 0.96]	0.93 [0.81, 0.99]
Thammanichanond 2005	LA	57	25	6	111	0.90 [0.80, 0.96]	0.82 [0.74, 0.88]
Vallbracht 2004	ELISA	196	75	99	345	0.66 [0.61, 0.72]	0.82 [0.78, 0.86]
van Leeuwen 1988	ELISA	163	10	28	140	0.85 [0.80, 0.90]	0.93 [0.88, 0.97]
Vasiliauskiene 2001	ELISA	75	21	21	106	0.78 [0.69, 0.86]	0.83 [0.76, 0.89]
Visser 1996	ELISA	157	287	78	1466	0.67 [0.60, 0.73]	0.84 [0.82, 0.85]
Vittecoq 2001	LA	26	1	32	29	0.45 [0.32, 0.58]	0.97 [0.83, 1.00]
Vittecoq 2004	ELISA	62	11	114	127	0.35 [0.28, 0.43]	0.92 [0.86, 0.96]
Winkles 1989	LA	113	19	29	481	0.80 [0.72, 0.86]	0.96 [0.94, 0.98]



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^2_α to zero,
- `constrain="sigma2_theta"` sets σ^2_θ to zero,

- `constrain="accuracy"` assumes equal accuracy parameters across subgroups, i.e. $\xi_2 = \xi_3 = \cdots = 0$,
- `constrain="threshold"` assumes equal threshold parameters across subgroups, i.e. $\gamma_2 = \gamma_3 = \cdots = 0$,
- `constrain="shape"` assumes equal shape parameters across subgroups, i.e. $\delta_2 = \delta_3 = \cdots = 0$,
- `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)
#> $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. 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)
#> $estimates
#>                       Estimate Std. Error
```

```
#> Lambda_LA            2.4243669 0.33049225
#> xi_ELISA             0.2974583 0.51315392
#> xi_Nephelometry      0.3716007 0.45086693
#> Theta_LA            -0.5029487 0.24451662
#> gamma_ELISA         -0.2578242 0.37024676
#> gamma_Nephelometry   0.3970656 0.35944299
#> beta_LA              0.2777491 0.26642343
#> delta_ELISA         -0.1028887 0.42661221
#> delta_Nephelometry  -0.1603831 0.39753788
#> sigma2_alpha         1.2730677 0.30798947
#> sigma2_theta         0.4762396 0.11343569
#> Lambda_LA            2.4243669 0.33049225
#> Lambda_ELISA         2.7218252 0.39299597
#> Lambda_Nephelometry  2.7959676 0.30708175
#> Theta_LA            -0.5029487 0.24451662
#> Theta_ELISA         -0.7607729 0.27817185
#> Theta_Nephelometry  -0.1058831 0.26322843
#> beta_LA              0.2777491 0.26642343
#> beta_ELISA           0.1748604 0.33321541
#> beta_Nephelometry    0.1173661 0.29500322
#> logitsens            0.8186925 0.27519154
#> logitsens            1.0626991 0.32405869
#> logitsens            1.1206495 0.28834985
#> sens                 0.6939587 0.05844518
#> sens                 0.7432060 0.06184687
#> sens                 0.7541092 0.05346829
#>
#> $sensspec
#>       subgroup      spec conflevel logitsens Std_Error  CI_Lower CI_Upper
#> 1           LA 0.8461538      0.95 0.8186925 0.2751915 0.2793270 1.358058
#> 2        ELISA 0.8461538      0.95 1.0626991 0.3240587 0.4275558 1.697843
#> 3 Nephelometry 0.8461538      0.95 1.1206495 0.2883499 0.5554942 1.685805
#>        Sens SensCI_Lower SensCI_Upper
#> 1 0.6939587    0.5693812    0.7954439
#> 2 0.7432060    0.6052899    0.8452527
#> 3 0.7541092    0.6354094    0.8436717
#>
#> $Reitsma_recovered
#>              mu_A.sens mu_B.spec sigma2_A.sens sigma2_B.spec   sigma_AB
#> LA           0.6172734  1.970652     0.6018282     1.0488717 -0.1579727
#> ELISA        0.5498977  2.315536     0.6670472     0.9463208 -0.1579727
#> Nephelometry 1.2184583  1.594759     0.7065226     0.8934472 -0.1579727
```

```
#>
#> $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)
#> $estimates
#>                       Estimate Std. Error
#> Lambda_LA           2.65809857 0.19181608
#> xi_ELISA            0.00000000 0.00000000
#> xi_Nephelometry     0.00000000 0.00000000
#> Theta_LA           -0.55745914 0.21293277
#> gamma_ELISA        -0.19385254 0.26093654
#> gamma_Nephelometry  0.49702816 0.26213094
#> beta_LA             0.18949665 0.16472998
#> delta_ELISA         0.00000000 0.00000000
#> delta_Nephelometry  0.00000000 0.00000000
#> sigma2_alpha        1.29064650 0.31233907
#> sigma2_theta        0.47671277 0.11344490
#> Lambda_LA           2.65809857 0.19181608
#> Lambda_ELISA        2.65809857 0.19181608
```

```
#> Lambda_Nephelometry  2.65809857 0.19181608
#> Theta_LA            -0.55745914 0.21293277
#> Theta_ELISA         -0.75131168 0.20809954
#> Theta_Nephelometry  -0.06043098 0.20958332
#> beta_LA              0.18949665 0.16472998
#> beta_ELISA           0.18949665 0.16472998
#> beta_Nephelometry    0.18949665 0.16472998
#> logitsens            1.00734453 0.16526474
#> logitsens            1.00734453 0.16526474
#> logitsens            1.00734453 0.16526474
#> sens                 0.73250015 0.03238258
#> sens                 0.73250015 0.03238258
#> sens                 0.73250015 0.03238258
#>
#> $sensspec
#>       subgroup      spec conflevel logitsens Std_Error  CI_Lower CI_Upper
#> 1           LA 0.8461538      0.95  1.007345 0.1652647 0.6834316 1.331257
#> 2        ELISA 0.8461538      0.95  1.007345 0.1652647 0.6834316 1.331257
#> 3 Nephelometry 0.8461538      0.95  1.007345 0.1652647 0.6834316 1.331257
#>        Sens SensCI_Lower SensCI_Upper
#> 1 0.7325001    0.6645042    0.7910486
#> 2 0.7325001    0.6645042    0.7910486
#> 3 0.7325001    0.6645042    0.7910486
#>
#> $Reitsma_recovered
#>              mu_A.sens mu_B.spec sigma2_A.sens sigma2_B.spec   sigma_AB
#> LA           0.7018398  2.073994     0.6613828     0.9661567 -0.1540511
#> ELISA        0.5255112  2.287112     0.6613828     0.9661567 -0.1540511
#> Nephelometry 1.1539375  1.527570     0.6613828     0.9661567 -0.1540511
#>
#> $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.

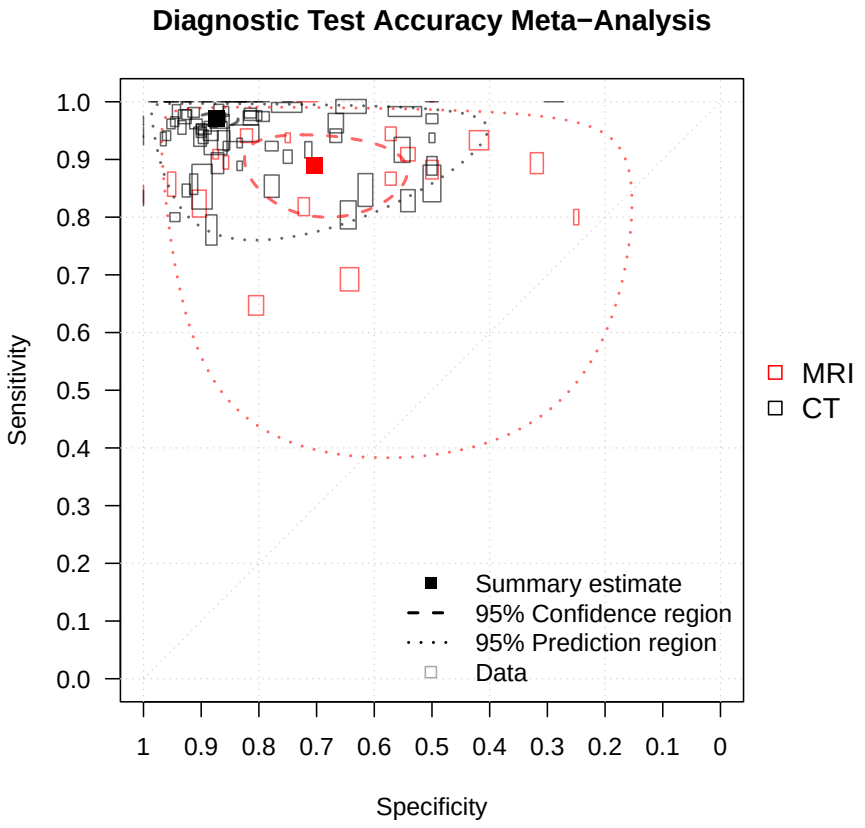
```
anova(ruttergatsonissub,
      ruttergatsonissubfull)
#>         Df  logLik Df.diff  Chisq Pr(>Chisq)
#> Model 1  9 -376.86
#> Model 2 11 -376.78       2 0.1692     0.9189
anova(ruttergatsonissubacc,
      ruttergatsonissub)
#>         Df  logLik Df.diff  Chisq Pr(>Chisq)
#> Model 1  7 -377.17
#>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

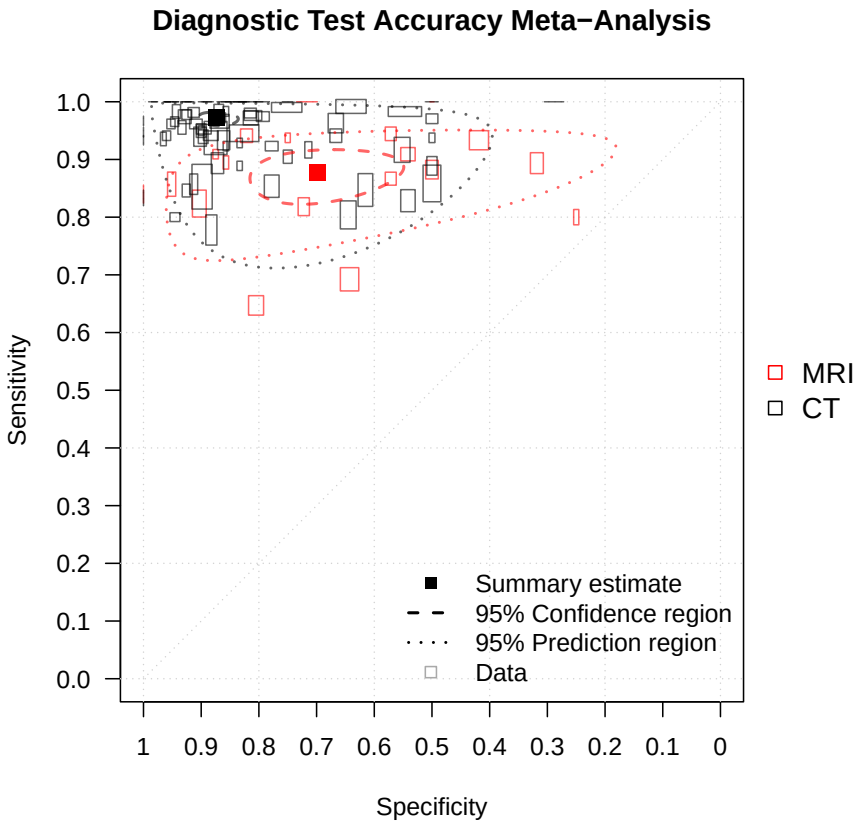
```
data(schuetz)
head(schuetz)
#>   test          study TP FP FN TN indirect
#> 1   CT Achenbach 2005 25  4  0 19        1
#> 2   CT   Alkadhi 2008 57 12  2 79        1
#> 3   CT  Andreini 2007 17  0  0 44        1
#> 4   CT    Bayrak 2008 64  4  0 32        1
#> 5  MRI    Bedaux 2002  7  1  0  1  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




```
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)
#> $estimates
#>                Estimate Std_Error
#> mu_A.MRI      1.9670363 0.1624331
#> mu_B.MRI      0.8410193 0.2406372
#> sigma2_A.MRI  0.1126405 0.1600126
#> sigma2_B.MRI  0.7121292 0.3424753
#> sigma_AB.MRI -0.1462045 0.1370573
#> mu_A.CT       3.5594295 0.1739508
#> mu_B.CT       1.9316450 0.1224654
#> sigma2_A.CT   1.1069322 0.3099921
#> sigma2_B.CT   0.8800412 0.1903877
#> sigma_AB.CT   0.3048890 0.1746537
#> nu_A.CT       1.5923912 0.2379991
#> nu_B.CT       1.0906240 0.2700068
#>
#> $sensspec
#>          type  Estimate conflevel  CI_Lower  CI_Upper
#> mu_A.MRI sens 0.8772924      0.95 0.8387117 0.9076606
#> mu_B.MRI spec 0.6986799      0.95 0.5913089 0.7879579
#> mu_A.CT  sens 0.9723322      0.95 0.9615243 0.9801668
#> mu_B.CT  spec 0.8734314      0.95 0.8444367 0.8976767
#>
#> $RutterGatsonis_recovered
#>       Lambda     Theta       beta sigma2_alpha sigma2_theta
#> MRI 3.649477 1.2943550  0.9220289    0.2740348    0.2147132
#> CT  5.406709 0.6577024 -0.1146895    2.5837546    0.3410496
#>
#> $subgroups
#> [1] "MRI" "CT"
plot(schuetzreitsma1,
     nudge_legend=-0.2,
     size="se",
     col=c("red","black"))
```



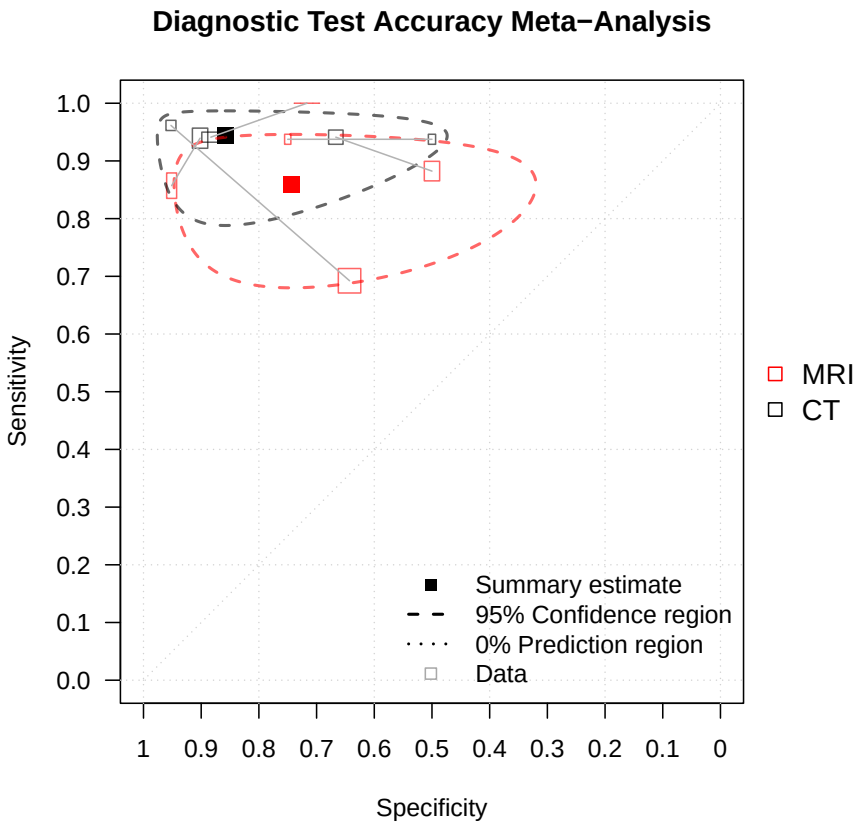

```
schuetzreitsma2 <- fitReitsmaSubgroup(data=schuetz,
                                      TP=TP,FP=FP,FN=FN,TN=TN,
                                      study=study,
                                      subgroup=test,
                                      sensspec_constrain="sens")
schuetzreitsma3 <- 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.
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.*