Blood Xpert-Ultra on biobanked KDHTB samples - analysis

Linda Boloko, David Barr, KDHTB study team

29 November 2019

Contents

1 Introduction

First objective is to compare *sensitivity* and *daignostic yield* of blood Xpert with different TB diagnostic tests applied routinely in KDHTB. We define these as:

Sensitivity = number of patients who have positive result on the index test divided by total number of patients who had:

1. A valid index test performed i.e. unable to obtain sample or technical problem with processing are excluded; AND
2. TB diagnosis confirmed by a *strict microbiological reference standard*: any positive TB culture result (sputum, blood or any other site) and/or positive Xpert from sputum urine or other site (blood Xpert not used in this reference standard as unvalidated). urine-LAM is also excluded.

Diagnostic yield = number of patients who have positive result on the index test divided by total number of patients who had TB diagnosis confirmed by *any TB diagnostic* including any positive TB culture from any site, any positive Xpert result (sputum, urine, blood, other), and/or positive urine LAM (Alere). Patients with a missing test result due to inability to obtain sample or technical failure of the index test are included as negative results in the numerator.

2 Overall study numbers for CONSORT diagram

```
# Data frame includes only patienst meeting global KDHTB inclusion criteria:
(N_kdhtb <- nrow(df))
```

```
## [1] 659
```

```
# excluding patients for whom blood sample not available
# (used elsewhere) - note have confirmed this manually
# with the raw data files - those coded NA there are no
# samples processed for.
(sum(is.na(df$blood_Xpert_MTB)))
```

```
## [1] 77
```

```
# Which leaves
df <- filter(df, !is.na(blood_Xpert_MTB))
(n_inclusion <- nrow(df))
```

```
## [1] 582
```

```
# Numbers from this n=582 meeting the 2 TB diagnosis ref standards
df %>%
  mutate(strict_micro_ref =
           (!is.na(df$sputumCulture1_cultureID) & df$sputumCulture1_cultureID=="MTB") |
           (!is.na(df$sputumCulture2_cultureID) & df$sputumCulture2_cultureID=="MTB") |
           (!is.na(df$sputumCulture3_cultureID) & df$sputumCulture3_cultureID=="MTB") |

           (!is.na(df$sputumGXP1_GeneXpert) &  df$sputumGXP1_GeneXpert=="MTB") |
           (!is.na(df$sputumGXP2_GeneXpert) & df$sputumGXP2_GeneXpert=="MTB") |
           (!is.na(df$sputumGXP3_GeneXpert) & df$sputumGXP3_GeneXpert=="MTB") |
           (!is.na(df$MBC1_cultureID) & df$MBC1_cultureID == "MTB") |
           (!is.na(df$MBC2_cultureID) & df$MBC2_cultureID == "MTB") |
           (!is.na(df$MBC3_cultureID) & df$MBC3_cultureID == "MTB") |

           (!is.na(df$uMTBculture) & df$uMTBculture == "MTB") |
           (!is.na(df$otherCul1_cultureID) & df$otherCul1_cultureID=="MTB") |
           (!is.na(df$otherCul2_cultureID) & df$otherCul2_cultureID=="MTB") |

           (!is.na(df$uGXP) & df$uGXP=="MTB") |
           (!is.na(df$otherGXP) & df$otherGXP=="MTB"),

         anyTBtest_pos =
           (strict_micro_ref==TRUE) |

           (!is.na(df$ALERE_FC) & df$ALERE_FC==1) |
    #       (!is.na(df$FUJISAI_FC) & df$FUJISAI_FC==1) |
           (!is.na(df$blood_Xpert_MTB) &
              df$blood_Xpert_MTB!="Negative" & df$blood_Xpert_MTB!="Error"),

         bld_xpert_pos = (!is.na(df$blood_Xpert_MTB) &
              df$blood_Xpert_MTB!="Negative" & df$blood_Xpert_MTB!="Error")
  ) -> df


#foo <- as.numeric(as.Date(df$StudyDate) - as.Date(df$DateOfAdmission))
#foo <- foo[foo<14]

# 4 patients wwho were blood xpert +ve but negative by all other TB diagnostics
q_FP_bldxpt <- df$UID[df$bld_xpert_diagnosed & df$strict_micro_ref==FALSE]
```

```
## Warning: Unknown or uninitialised column: `bld_xpert_diagnosed`.
```

```
df$FUJISAI_FC[df$UID %in% q_FP_bldxpt] # 1/2 fuji +ve
```

```
## numeric(0)
```

```
(n_strict <- sum(df$strict_micro_ref))
```

```
## [1] 424
```

```
(n_any <- sum(df$anyTBtest_pos))
```

```
## [1] 447
```

3 Cohort description table

This was discussed as supplementary table in Nov 2019 skype call. We can do complete cohort (n=582), or disagregate by TB diagnosis status (strict micro ref or any TB test positive), or disgregate by blood Xpert status. All 3 disagregated versions are shown below - can discuss which to include.

```
mycontrols <- tableby.control(numeric.test = "kwt",
                              cat.test="fe",
                              cat.simplify = FALSE,
                              numeric.stats = c("median", "q1q3"))
```

```r
tab1 <- tableby(strict_micro_ref ~ age + Sex + CD4 + ARTstatus +
                  HR + lactate + Haemoglobin +
                  creatinine + CRP + Sodium +
                  Cough + LossOfAppetite + DrenchingNightSweats + LossOfWeight +
                  survival.12weeks,
                data = df, control=mycontrols)

tab2 <- tableby(anyTBtest_pos ~ age + Sex + CD4 + ARTstatus +
                  HR + lactate + Haemoglobin +
                  creatinine + CRP + Sodium +
                  Cough + LossOfAppetite + DrenchingNightSweats + LossOfWeight +
                  survival.12weeks,
                data = df, control=mycontrols)



tab3 <- tableby(bld_xpert_pos ~ age + Sex + CD4 + ARTstatus +
                  HR + lactate + Haemoglobin +
                  creatinine + CRP + Sodium +
                  Cough + LossOfAppetite + DrenchingNightSweats + LossOfWeight +
                  survival.12weeks,
                data = df[df$blood_Xpert_MTB!="Error",], control=mycontrols)
```

3.1 Disaggregated by strict micro reference standard T/F

```r
summary(tab1, text=TRUE)
```

	FALSE (N=158)	TRUE (N=424)	Total (N=582)	p value
age				0.128
- Median	38.133	35.860	36.274	
- Q1, Q3	31.067, 44.842	30.925, 43.213	30.955, 43.999	
Sex				0.113
- F	91 (57.6%)	212 (50.0%)	303 (52.1%)	
- M	67 (42.4%)	212 (50.0%)	279 (47.9%)	
CD4				< 0.001
- Median	90.500	55.000	62.000	
- Q1, Q3	36.250, 175.500	18.000, 115.000	22.000, 132.500	
ARTstatus				0.145
- N-Miss	4	3	7	
- Defaulted	38 (24.7%)	95 (22.6%)	133 (23.1%)	
- Naive	49 (31.8%)	171 (40.6%)	220 (38.3%)	
- On_ART	67 (43.5%)	155 (36.8%)	222 (38.6%)	
HR				< 0.001
- Median	96.500	107.000	104.000	
- Q1, Q3	86.000, 110.000	98.000, 120.000	94.000, 120.000	
lactate				< 0.001
- Median	1.500	1.900	1.800	
- Q1, Q3	1.100, 2.100	1.400, 2.600	1.300, 2.500	
Haemoglobin				< 0.001
- Median	9.850	8.400	8.800	
- Q1, Q3	8.100, 11.400	7.000, 10.100	7.300, 10.500	

	FALSE (N=158)	TRUE (N=424)	Total (N=582)	p value
creatinine				0.411
- Median	76.000	80.000	78.500	
- Q1, Q3	59.000, 117.500	59.750, 117.250	59.000, 117.750	
CRP				< 0.001
- Median	101.200	170.500	153.500	
- Q1, Q3	51.500, 228.300	103.650, 231.900	86.550, 231.500	
Sodium				< 0.001
- Median	131.000	128.000	129.000	
- Q1, Q3	127.000, 133.000	125.000, 131.000	125.000, 132.000	
Cough				0.760
- N-Miss	5	16	21	
- N	46 (30.1%)	129 (31.6%)	175 (31.2%)	
- Y	107 (69.9%)	279 (68.4%)	386 (68.8%)	
LossOfAppetite				0.615
- N-Miss	6	20	26	
- N	54 (35.5%)	133 (32.9%)	187 (33.6%)	
- Y	98 (64.5%)	271 (67.1%)	369 (66.4%)	
DrenchingNightSweats				1.000
- N-Miss	6	24	30	
- N	67 (44.1%)	178 (44.5%)	245 (44.4%)	
- Y	85 (55.9%)	222 (55.5%)	307 (55.6%)	
LossOfWeight				0.352
- N-Miss	6	21	27	
- N	19 (12.5%)	39 (9.7%)	58 (10.5%)	
- Y	133 (87.5%)	364 (90.3%)	497 (89.5%)	
survival.12weeks				0.889
- Died	33 (20.9%)	90 (21.2%)	123 (21.1%)	
- LTFU	4 (2.5%)	8 (1.9%)	12 (2.1%)	
- Survived	121 (76.6%)	326 (76.9%)	447 (76.8%)	

3.2 Disaggregated by any TB test positive T/F

```
summary(tab2, text=TRUE)
```

	FALSE (N=135)	TRUE (N=447)	Total (N=582)	p value
age				0.086
- Median	38.912	35.866	36.274	
- Q1, Q3	30.971, 46.121	30.949, 43.138	30.955, 43.999	
Sex				0.202
- F	77 (57.0%)	226 (50.6%)	303 (52.1%)	
- M	58 (43.0%)	221 (49.4%)	279 (47.9%)	
CD4				< 0.001
- Median	91.000	55.000	62.000	
- Q1, Q3	37.500, 184.000	18.000, 117.500	22.000, 132.500	
ARTstatus				0.085
- N-Miss	4	3	7	
- Defaulted	31 (23.7%)	102 (23.0%)	133 (23.1%)	
- Naive	40 (30.5%)	180 (40.5%)	220 (38.3%)	
- On_ART	60 (45.8%)	162 (36.5%)	222 (38.6%)	
HR				< 0.001

	FALSE (N=135)	TRUE (N=447)	Total (N=582)	p value
- Median	96.000	107.000	104.000	
- Q1, Q3	86.000, 110.500	97.000, 120.000	94.000, 120.000	
lactate				< 0.001
- Median	1.450	1.850	1.800	
- Q1, Q3	1.100, 2.000	1.400, 2.700	1.300, 2.500	
Haemoglobin				< 0.001
- Median	10.000	8.400	8.800	
- Q1, Q3	8.250, 11.600	7.000, 10.100	7.300, 10.500	
creatinine				0.638
- Median	76.000	79.000	78.500	
- Q1, Q3	59.500, 117.000	59.000, 117.500	59.000, 117.750	
CRP				< 0.001
- Median	104.200	167.000	153.500	
- Q1, Q3	51.550, 239.025	101.000, 231.000	86.550, 231.500	
Sodium				< 0.001
- Median	131.000	128.000	129.000	
- Q1, Q3	127.000, 133.000	125.000, 131.000	125.000, 132.000	
Cough				0.107
- N-Miss	5	16	21	
- N	33 (25.4%)	142 (32.9%)	175 (31.2%)	
- Y	97 (74.6%)	289 (67.1%)	386 (68.8%)	
LossOfAppetite				0.243
- N-Miss	6	20	26	
- N	49 (38.0%)	138 (32.3%)	187 (33.6%)	
- Y	80 (62.0%)	289 (67.7%)	369 (66.4%)	
DrenchingNightSweats				0.840
- N-Miss	6	24	30	
- N	56 (43.4%)	189 (44.7%)	245 (44.4%)	
- Y	73 (56.6%)	234 (55.3%)	307 (55.6%)	
LossOfWeight				0.073
- N-Miss	6	21	27	
- N	19 (14.7%)	39 (9.2%)	58 (10.5%)	
- Y	110 (85.3%)	387 (90.8%)	497 (89.5%)	
survival.12weeks				0.625
- Died	27 (20.0%)	96 (21.5%)	123 (21.1%)	
- LTFU	4 (3.0%)	8 (1.8%)	12 (2.1%)	
- Survived	104 (77.0%)	343 (76.7%)	447 (76.8%)	

3.3 Disaggregated by blood Xpert positive T/F

```
summary(tab3, text=TRUE)
```

	FALSE (N=413)	TRUE (N=165)	Total (N=578)	p value
age				0.849
- Median	36.307	36.003	36.221	
- Q1, Q3	30.964, 44.008	30.928, 43.761	30.947, 43.859	
Sex				0.065
- F	227 (55.0%)	76 (46.1%)	303 (52.4%)	
- M	186 (45.0%)	89 (53.9%)	275 (47.6%)	
CD4				< 0.001

	FALSE (N=413)	TRUE (N=165)	Total (N=578)	p value
- Median	86.000	25.000	62.000	
- Q1, Q3	34.000, 160.000	8.000, 60.000	22.000, 133.000	
ARTstatus				< 0.001
- N-Miss	6	1	7	
- Defaulted	81 (19.9%)	49 (29.9%)	130 (22.8%)	
- Naive	150 (36.9%)	70 (42.7%)	220 (38.5%)	
- On_ART	176 (43.2%)	45 (27.4%)	221 (38.7%)	
HR				< 0.001
- Median	102.500	111.000	104.000	
- Q1, Q3	92.000, 117.000	98.000, 123.000	94.000, 120.000	
lactate				< 0.001
- Median	1.700	2.100	1.800	
- Q1, Q3	1.200, 2.300	1.500, 3.100	1.300, 2.500	
Haemoglobin				< 0.001
- Median	9.300	8.000	8.800	
- Q1, Q3	7.600, 10.800	6.700, 9.300	7.300, 10.500	
creatinine				< 0.001
- Median	76.000	95.000	78.000	
- Q1, Q3	58.000, 105.000	66.000, 161.000	59.000, 117.000	
CRP				< 0.001
- Median	137.000	196.000	153.500	
- Q1, Q3	75.275, 225.075	130.000, 251.000	86.550, 231.800	
Sodium				< 0.001
- Median	130.000	127.000	129.000	
- Q1, Q3	126.000, 132.000	124.000, 130.000	125.000, 132.000	
Cough				0.544
- N-Miss	13	8	21	
- N	122 (30.5%)	52 (33.1%)	174 (31.2%)	
- Y	278 (69.5%)	105 (66.9%)	383 (68.8%)	
LossOfAppetite				0.190
- N-Miss	14	12	26	
- N	140 (35.1%)	44 (28.8%)	184 (33.3%)	
- Y	259 (64.9%)	109 (71.2%)	368 (66.7%)	
DrenchingNightSweats				0.848
- N-Miss	18	12	30	
- N	178 (45.1%)	67 (43.8%)	245 (44.7%)	
- Y	217 (54.9%)	86 (56.2%)	303 (55.3%)	
LossOfWeight				0.759
- N-Miss	16	11	27	
- N	43 (10.8%)	15 (9.7%)	58 (10.5%)	
- Y	354 (89.2%)	139 (90.3%)	493 (89.5%)	
survival.12weeks				< 0.001
- Died	70 (16.9%)	51 (30.9%)	121 (20.9%)	
- LTFU	10 (2.4%)	2 (1.2%)	12 (2.1%)	
- Survived	333 (80.6%)	112 (67.9%)	445 (77.0%)	

4 Sensitivity & diagnostic yield

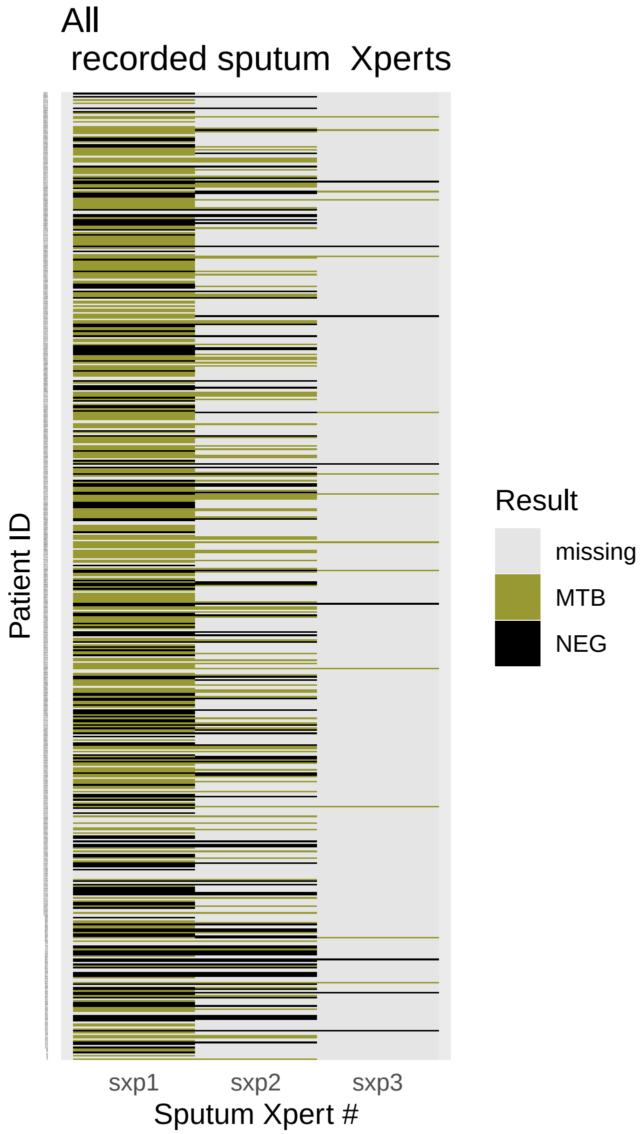
4.1 Which patients had sputum samples collected?

We want the sensitivity of a single sputum Xpert to compare against the sesnitivity of a single blood xpert etc. Some patients didn't have a *study* sputum sample because they already had one performed in routine care, either on admission to hospital or prior to admission at clinics. When sputums were recorded in general any positive result was recorded first in the three recoded sputum Xpert variables ("sputumGXP1", "sputumGXP2", "sputumGXP3"). In addition, any positive sputum Xpert seen on review of the NHLS electronic record was added to the KDHTB database, even some that were performed months before or after date of recruitment to study.

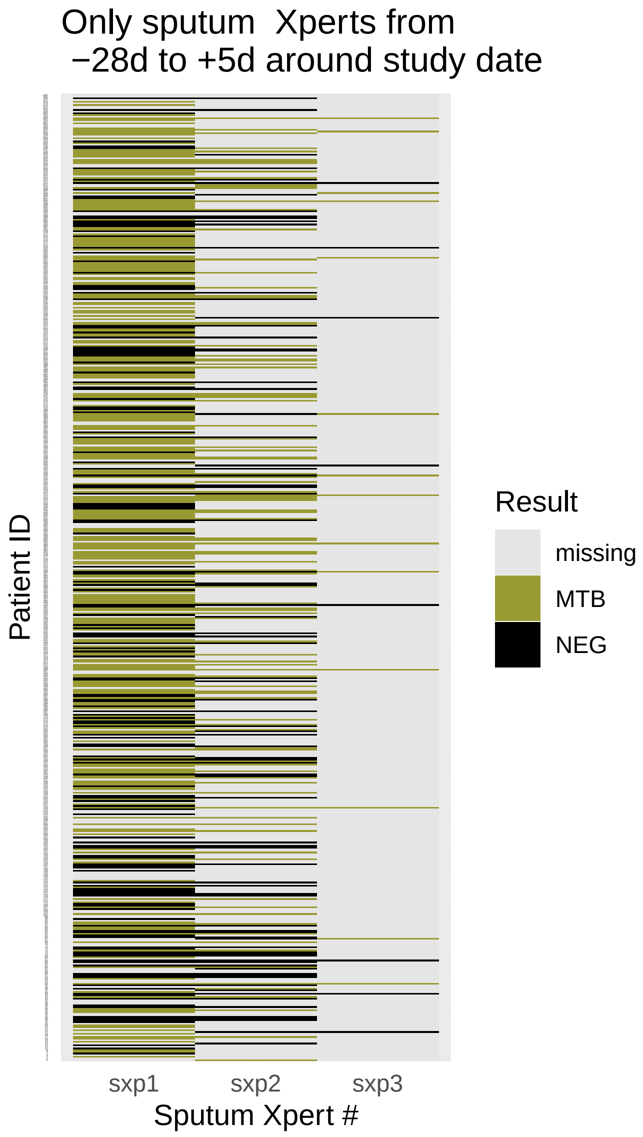
Proposal (for discussion) is to select only sputum Xperts collected between 28 days before and 5 days after day of study recruitment, and if there are more than one, select the one closest to day of recruitment, and if >1 at that timepoint select one of these at random.

Below are the sputum Xpert results by patient. On left are all the results, on right are the results limited to sputa between 28 days before and 5 days after date of study recruitmnent.

```
df %>%
    mutate(
    sptmxpert1_day = as.numeric(as.Date(df$sputumGXP1_Date) - df$StudyDate),
    sptmxpert2_day = as.numeric(as.Date(df$sputumGXP2_Date) - df$StudyDate),
    sptmxpert3_day = as.numeric(as.Date(df$sputumGXP3_Date) - df$StudyDate),
    # same for 2 blood cultures?
    mfl1_day = as.numeric(as.Date(df$MBC1_Date) - df$StudyDate),
    mfl2_day = as.numeric(as.Date(df$MBC2_Date) - df$StudyDate)) -> df

#df %>%
#  select(UID,
#         sptmxpert1_day, sptmxpert2_day, sptmxpert3_day) %>%
#  gather(key="sputum_num", value = "day", 2:4) %>%
#  ggplot(aes(day, UID)) +
#  geom_point(size=0.7, colour="#882255") +
#  xlim(-20, 50) +
#  xlab("Days from study recruitment date") +
#  ylab("Patient's study ID") +
#  theme_minimal()


df %>%
  mutate(sxp1 = sputumGXP1_GeneXpert,
         sxp2 = sputumGXP2_GeneXpert,
         sxp3 = sputumGXP3_GeneXpert) %>%
  select(UID, sxp1, sxp2, sxp3) %>%
  gather(key="sptm_xpert",
         value="Result", 2:4) %>%
  replace_na(list(Result = "missing")) %>%
  ggplot(aes(sptm_xpert, as.factor(UID))) +
  geom_tile(aes(fill=Result)) +
  scale_fill_manual(
    values = c("grey90", "#999933", "black")) +
  labs(x = "Sputum Xpert #", y = "Patient ID") +
  theme_minimal() +
  theme(axis.text.y = element_text(size=0.5)) +
  ggtitle("All\n recorded sputum  Xperts")
```



```
df %>%
  mutate(sxp1 = sputumGXP1_GeneXpert,
         sxp2 = sputumGXP2_GeneXpert,
         sxp3 = sputumGXP3_GeneXpert) %>%
  mutate(sxp1 = replace(sxp1,
                        (!is.na(sptmxpert1_day) &
                           (sptmxpert1_day <= -28 |
                           sptmxpert1_day>5)), NA),
         sxp2 = replace(sxp2,
                        (!is.na(sptmxpert2_day) &
                           (sptmxpert2_day <= -29 |
                           sptmxpert2_day>5)), NA),
         sxp3 = replace(sxp3,
                        (!is.na(sptmxpert3_day) &
                           (sptmxpert3_day <= -29 |
                                  sptmxpert3_day>5)), NA)) %>%
  select(UID, sxp1, sxp2, sxp3) %>%
  gather(key="sptm_xpert",
         value="Result", 2:4) %>%
  replace_na(list(Result = "missing")) %>%
  ggplot(aes(sptm_xpert, as.factor(UID))) +
  geom_tile(aes(fill=Result)) +
  scale_fill_manual(
    values = c("grey90", "#999933", "black")) +
  labs(x = "Sputum Xpert #", y = "Patient ID") +
  theme_minimal() +
  theme(axis.text.y = element_text(size=0.5)) +
  ggtitle("Only sputum  Xperts from\n -28d to +5d around study date")
```



Same data cross tabulated of sputum 1 (rows) and sputum 2 (columns):

All results

```r
sptm_xpert_1 <- df$sputumGXP1_GeneXpert
sptm_xpert_2 <- df$sputumGXP2_GeneXpert
sptm_xpert_1[is.na(sptm_xpert_1)] <- "missing"
sptm_xpert_2[is.na(sptm_xpert_2)] <- "missing"

mfl1 <- df$MBC1_cultureID=="MTB"
mfl2 <- df$MBC2_cultureID=="MTB"

mfl1[df$mfl1_day < -5 | df$mfl1_day > 5]
```

```
##  [1]    NA    NA    NA    NA    NA  TRUE    NA    NA    NA    NA    NA
## [12]    NA FALSE  TRUE FALSE    NA FALSE    NA    NA FALSE    NA FALSE
## [23]    NA FALSE    NA FALSE FALSE FALSE    NA  TRUE FALSE    NA FALSE
## [34]    NA    NA    NA
```

```r
kable(table(sptm_xpert_1, sptm_xpert_2),
     "latex", booktabs=T)
```

	missing	MTB	NEG
missing	125	0	0
MTB	166	99	11
NEG	91	11	79

Day -28 to +5 results only

```r
sptm_xpert_1[!is.na(df$sptmxpert1_day) &
  (df$sptmxpert1_day <= -29 | df$sptmxpert1_day>5)] <- "missing"

sptm_xpert_2[!is.na(df$sptmxpert2_day) &
  (df$sptmxpert2_day <= -29 | df$sptmxpert2_day>5)] <- "missing"

kable(table(sptm_xpert_1, sptm_xpert_2),
     "latex", booktabs=T)
```

	missing	MTB	NEG
missing	137	8	9
MTB	168	91	7
NEG	85	6	71

Are some of these ‘missing’ sputum Xperts because there was already a sputum culture result so we didn’t try to obtain a sputum Xpert? Probably not - here is cross tabulation of Sputum Xpert 1 (rows) with sputum culture 1 (columns); there are only 13 patients who have a sputum culture but no sputum Xpert:

```r
sptm_xpert_1 <- df$sputumGXP1_GeneXpert
sptm_xpert_1[is.na(sptm_xpert_1)] <- "missing"
sptm_culture_1 <- df$sputumCulture1_cultureID
sptm_culture_1[is.na(sptm_culture_1)] <- "missing"

kable(table(sptm_xpert_1, sptm_culture_1),
     "latex", booktabs=T)
```

	AFB	contaminated	missing	MTB	negative	NTM
missing	0	0	112	10	3	0
MTB	1	12	30	219	13	1
NEG	1	10	18	43	108	1

Will proceed with selecting sputum Xpert between -28 and +5 days from date of study recruitment closest to day of recruitment as the sputum Xpert variable for this study (but this can be changed easily on discussion).

```
# set up some new variables which are just copies
# of the orginal GXP variables
df$sputumGXP1 <- df$sputumGXP1_GeneXpert
df$sputumGXP2 <- df$sputumGXP2_GeneXpert
df$sputumGXP3 <- df$sputumGXP3_GeneXpert

# remove the results outside our date range
df$sputumGXP1[!is.na(df$sptmxpert1_day) &
  (df$sptmxpert1_day <= -29 | df$sptmxpert1_day>5)] <- NA
df$sputumGXP2[!is.na(df$sptmxpert2_day) &
  (df$sptmxpert2_day <= -29 | df$sptmxpert2_day>5)] <- NA
df$sputumGXP3[!is.na(df$sptmxpert3_day) &
  (df$sptmxpert3_day <= -29 | df$sptmxpert3_day>5)] <- NA

# also remove the day of collection from those samples so it doesn't mess with later for loop
df$sptmxpert1_day[is.na(df$sputumGXP1)] <- NA
df$sptmxpert2_day[is.na(df$sputumGXP2)] <- NA
df$sptmxpert3_day[is.na(df$sputumGXP3)] <- NA

# set seed to make random picking of the results reproducable
set.seed(123)

# create a new variable which will be our final sputum Xpert result
df$sputum_xpert <- rep("foo", nrow(df))

# This for loop now populates that new sputum variable so that it is:
## NA if all 3 sputum Xperts are NA
## gets result of single Xpert result if only one available
## picks closest to recruitment date or 'samples' one at random if 2 or 3 are available on same day

for(i in 1:nrow(df)){

  if(is.na(df$sputumGXP1[i]) &
     is.na(df$sputumGXP2[i]) &
     is.na(df$sputumGXP3[i])){
    df$sputum_xpert[i] <- NA  # If all 3 NA then result is NA
     } else

  if(!is.na(df$sputumGXP1[i]) &
     is.na(df$sputumGXP2[i]) &
     is.na(df$sputumGXP3[i])){
    df$sputum_xpert[i] <- df$sputumGXP1[i]
     } else

  if(is.na(df$sputumGXP1[i]) &
     !is.na(df$sputumGXP2[i]) &
     is.na(df$sputumGXP3[i])){
    df$sputum_xpert[i] <- df$sputumGXP2[i]
     } else

  if(is.na(df$sputumGXP1[i]) &
     is.na(df$sputumGXP2[i]) &
     !is.na(df$sputumGXP3[i])){
    df$sputum_xpert[i] <- df$sputumGXP3[i]
     } else
                    # If only 1/3 recorded then result is that one
  if(!is.na(df$sputumGXP1[i]) &
     !is.na(df$sputumGXP2[i]) &
     is.na(df$sputumGXP3[i])){
    if(
      (abs(df$sptmxpert1_day[i])<abs(df$sptmxpert2_day[i]) &
         !is.na(abs(df$sptmxpert1_day[i])<abs(df$sptmxpert2_day[i])))
    ){
      df$sputum_xpert[i] <- df$sputumGXP1[i]
      }else
        if(
      (abs(df$sptmxpert1_day[i])>abs(df$sptmxpert2_day[i]) &
         !is.na(abs(df$sptmxpert1_day[i])>abs(df$sptmxpert2_day[i])))
    ){
      df$sputum_xpert[i] <- df$sputumGXP2[i]
      } else
        if(
          (abs(df$sptmxpert1_day[i])==abs(df$sptmxpert2_day[i]) &
         !is.na(abs(df$sptmxpert1_day[i])==abs(df$sptmxpert2_day[i])))
        ){
          df$sputum_xpert[i] <-
      sample(c(df$sputumGXP1[i],
               df$sputumGXP2[i]), 1)
        }
             } else
                    # if 2 result available sample 1 closest to recruitment and if both same day select

    if(is.na(df$sputumGXP1[i]) &
     !is.na(df$sputumGXP2[i]) &
     !is.na(df$sputumGXP3[i])){
    if(
      (abs(df$sptmxpert2_day[i])<abs(df$sptmxpert3_day[i]) &
         !is.na(abs(df$sptmxpert2_day[i])<abs(df$sptmxpert3_day[i])))
    ){
      df$sputum_xpert[i] <- df$sputumGXP2[i]
      } else
        if(
      (abs(df$sptmxpert2_day[i])>abs(df$sptmxpert3_day[i]) &
         !is.na(abs(df$sptmxpert2_day[i])>abs(df$sptmxpert3_day[i])))
    ){
      df$sputum_xpert[i] <- df$sputumGXP3[i]
      } else
        if(
          (abs(df$sptmxpert2_day[i])==abs(df$sptmxpert3_day[i]) &
         !is.na(abs(df$sptmxpert2_day[i])==abs(df$sptmxpert3_day[i])))
        ){
          df$sputum_xpert[i] <-
      sample(c(df$sputumGXP2[i],
               df$sputumGXP3[i]), 1)
        }
             } else

    if(!is.na(df$sputumGXP1[i]) &
     is.na(df$sputumGXP2[i]) &
     !is.na(df$sputumGXP3[i])){
    if(
      (abs(df$sptmxpert1_day[i])<abs(df$sptmxpert3_day[i]) &
         !is.na(abs(df$sptmxpert1_day[i])<abs(df$sptmxpert3_day[i])))
    ){
      df$sputum_xpert[i] <- df$sputumGXP1[i]
      } else
        if(
      (abs(df$sptmxpert1_day[i])>abs(df$sptmxpert3_day[i]) &
         !is.na(abs(df$sptmxpert1_day[i])>abs(df$sptmxpert3_day[i])))
    ){
      df$sputum_xpert[i] <- df$sputumGXP3[i]
      } else
        if(
          (abs(df$sptmxpert1_day[i])==abs(df$sptmxpert3_day[i]) &
         !is.na(abs(df$sptmxpert1_day[i])==abs(df$sptmxpert3_day[i])))
        ){
          df$sputum_xpert[i] <-
      sample(c(df$sputumGXP1[i],
               df$sputumGXP3[i]), 1)
        }
             } else
       # now for the times when all 3 results are available...
  if(!is.na(df$sputumGXP1[i]) &
     !is.na(df$sputumGXP2[i]) &
     !is.na(df$sputumGXP3[i])){

# one sample of 3 is closest to recruitment:
      if(
        (abs(df$sptmxpert1_day[i])<abs(df$sptmxpert2_day[i])) &
        (abs(df$sptmxpert1_day[i])<abs(df$sptmxpert3_day[i]))){
          df$sputum_xpert[i] <- df$sputumGXP1[i]}else
      if(
        (abs(df$sptmxpert2_day[i])<abs(df$sptmxpert1_day[i])) &
        (abs(df$sptmxpert2_day[i])<abs(df$sptmxpert3_day[i]))){
          df$sputum_xpert[i] <- df$sputumGXP2[i]}else
      if(
        (abs(df$sptmxpert3_day[i])<abs(df$sptmxpert2_day[i])) &
        (abs(df$sptmxpert3_day[i])<abs(df$sptmxpert1_day[i]))){
          df$sputum_xpert[i] <- df$sputumGXP3[i]}else

# now cases where 2 of 3 available are same day

      if(
        (abs(df$sptmxpert1_day[i])<abs(df$sptmxpert2_day[i])) &
        (abs(df$sptmxpert1_day[i])==abs(df$sptmxpert3_day[i]))){
          df$sputum_xpert[i] <- sample(
            c(df$sputumGXP1[i], df$sputumGXP3[i]), 1)}else
      if(
        (abs(df$sptmxpert1_day[i])<abs(df$sptmxpert3_day[i])) &
        (abs(df$sptmxpert1_day[i])==abs(df$sptmxpert2_day[i]))){
          df$sputum_xpert[i] <- sample(
            c(df$sputumGXP1[i], df$sputumGXP2[i]), 1)}else
      if(
        (abs(df$sptmxpert2_day[i])<abs(df$sptmxpert1_day[i])) &
        (abs(df$sptmxpert2_day[i])==abs(df$sptmxpert3_day[i]))){
          df$sputum_xpert[i] <- sample(
            c(df$sputumGXP2[i], df$sputumGXP3[i]), 1)}else

# all 3 are same day, sample one at random
      if(
        (abs(df$sptmxpert1_day[i])==abs(df$sptmxpert2_day[i])) &
        (abs(df$sptmxpert2_day[i])==abs(df$sptmxpert3_day[i]))
        ){
        df$sputum_xpert[i] <- sample(
          c(df$sputumGXP1[i], df$sputumGXP2[i], df$sputumGXP3[i]), 1)}
  }
}
```

When do this the final results for a single sputum Xpert are:

```
kable(table(df$sputum_xpert, useNA = "always", deparse.level = 2),
     "latex", booktabs=T)
```

df.sputum_xpert	Freq
MTB	275
NEG	170
NA	137

4.2 Sensitivity

% positive index tests from denominator of proven TB by strict micro reference and valid test performed.

4.2.1 In whole cohort

```
# make a function which collects the statistics we want for sensitivity: number with valid test for ind

sens_function <- function(reference_std, index_test){

  dat <- data.frame(reference_std, index_test)
  dat <- dat[!is.na(index_test),]
  n_valid_test = nrow(dat)
  n_provenTB = sum(dat$reference_std==TRUE)
  n_true_positive = sum(dat$reference_std==TRUE &
                          dat$index_test=="MTB")
  sens = round(n_true_positive / n_provenTB, 2)
  CI_95 = paste0(
    round(
    prop.test(
      n_true_positive, n_provenTB)$conf.int[1],
    2),
    " to ",
    round(
    prop.test(
      n_true_positive, n_provenTB)$conf.int[2],
    2))

  return(data.frame(n_valid_test, n_provenTB,
           n_true_positive, sens, CI_95))
}

# fix up how the results are coded in these variables:
df$ALERE_FC[df$ALERE_FC==1] <- "MTB"
df$ALERE_FC[df$ALERE_FC=="0"] <- "NEG"
df$bld_xpert_pos[df$bld_xpert_pos==TRUE] <- "MTB"
df$bld_xpert_pos[df$bld_xpert_pos=="FALSE"] <- "NEG"
df$bld_xpert_pos[df$blood_Xpert_MTB=="Error"] <- NA

# can now simply apply our function to each variable of interest
# and combine them in an data frame
bind_rows(
  sens_function(df$strict_micro_ref, df$ALERE_FC),
  sens_function(df$strict_micro_ref, df$bld_xpert_pos)) %>%
  mutate(index_test =
           c("Alere_LAM", "blood_Xpert")) %>%
  select(index_test, everything()) -> sens_table1

kable(sens_table1,
     "latex", booktabs=T)
```

index_test	n_valid_test	n_provenTB	n_true_positive	sens	CI_95
Alere_LAM	519	375	171	0.46	0.4 to 0.51
blood_Xpert	578	423	161	0.38	0.33 to 0.43

Sputum Xpert now removed as incorporation bias

4.2.2 In pre-specified sub-groups

```
# make dataframes which are sub-groups of interest

df[df$CD4<100,] -> cd4_df
df[df$lactate>2.5 & !is.na(df$lactate),] -> lact_df
df[df$Haemoglobin<8,] -> hb_df
df[df$survival.12weeks=="Died",] -> died_df

# re-use our function from above
bind_rows(
  sens_function(cd4_df$strict_micro_ref, cd4_df$ALERE_FC),
  sens_function(cd4_df$strict_micro_ref, cd4_df$bld_xpert_pos)) %>%
  mutate(index_test =
           c("Alere_LAM", "blood_Xpert")) %>%
  select(index_test, everything()) -> sens_table_cd4
```

```r
bind_rows(
  sens_function(lact_df$strict_micro_ref, lact_df$ALERE_FC),
  sens_function(lact_df$strict_micro_ref, lact_df$bld_xpert_pos)) %>%
  mutate(index_test =
           c("Alere_LAM", "blood_Xpert")) %>%
  select(index_test, everything()) -> sens_table_lact

bind_rows(
  sens_function(hb_df$strict_micro_ref, hb_df$ALERE_FC),
  sens_function(hb_df$strict_micro_ref, hb_df$bld_xpert_pos)) %>%
  mutate(index_test =
           c("Alere_LAM", "blood_Xpert")) %>%
  select(index_test, everything()) -> sens_table_hb

bind_rows(
  sens_function(died_df$strict_micro_ref, died_df$ALERE_FC),
  sens_function(died_df$strict_micro_ref, died_df$bld_xpert_pos)) %>%
  mutate(index_test =
           c("Alere_LAM", "blood_Xpert")) %>%
  select(index_test, everything()) -> sens_table_died
```

4.2.2.1 CD4 $<$ 100

```r
kable(sens_table_cd4,
     "latex", booktabs=T)
```

index_test	n_valid_test	n_provenTB	n_true_positive	sens	CI_95
Alere_LAM	328	253	144	0.57	0.51 to 0.63
blood_Xpert	371	287	144	0.50	0.44 to 0.56

4.2.2.2 Haemoglobin $<$ 8

```r
kable(sens_table_hb,
     "latex", booktabs=T)
```

index_test	n_valid_test	n_provenTB	n_true_positive	sens	CI_95
Alere_LAM	173	142	87	0.61	0.53 to 0.69
blood_Xpert	204	168	78	0.46	0.39 to 0.54

4.2.2.3 Lactate $>$ 2.5

```r
kable(sens_table_lact,
     "latex", booktabs=T)
```

index_test	n_valid_test	n_provenTB	n_true_positive	sens	CI_95
Alere_LAM	117	95	47	0.49	0.39 to 0.6
blood_Xpert	137	113	57	0.50	0.41 to 0.6

4.2.2.4 Those who died by 12 weeks

```
kable(sens_table_died,
     "latex", booktabs=T)
```

index_test	n_valid_test	n_provenTB	n_true_positive	sens	CI_95
Alere_LAM	101	74	39	0.53	0.41 to 0.64
blood_Xpert	121	89	49	0.55	0.44 to 0.65

4.3 Diagnostic yield

Reference standard is any positive TB test; those with missing index test result are included as negative test.

4.3.1 In whole cohort

```
yield_function <- function(reference_std, index_test){
  dat <- data.frame(reference_std, index_test,
                    stringsAsFactors = FALSE)
  dat$index_test[is.na(dat$index_test)] <- "neg"
  # dat <- dat[!is.na(index_test),] # keep these in
  N = nrow(dat) # this is now all the patients
  n_TB = sum(dat$reference_std==TRUE)
  n_true_positive = sum(dat$reference_std==TRUE &
                          dat$index_test=="MTB")
  diag_yield = round(n_true_positive / n_TB, 2)
  CI_95 = paste0(
    round(
    prop.test(
      n_true_positive, n_TB)$conf.int[1],
    2),
    " to ",
    round(
    prop.test(
      n_true_positive, n_TB)$conf.int[2],
    2))

  return(data.frame(N, n_TB,
           n_true_positive, diag_yield, CI_95))
}

bind_rows(
  yield_function(df$anyTBtest_pos, df$sputum_xpert),
  yield_function(df$anyTBtest_pos, df$ALERE_FC),
  yield_function(df$anyTBtest_pos, df$bld_xpert_pos)) %>%
  mutate(index_test =
           c("sputum_Xpert", "Alere_LAM", "blood_Xpert")) %>%
  select(index_test, everything()) -> yield_table1

bind_rows(
  yield_function(cd4_df$anyTBtest_pos, cd4_df$sputum_xpert),
  yield_function(cd4_df$anyTBtest_pos, cd4_df$ALERE_FC),
  yield_function(cd4_df$anyTBtest_pos, cd4_df$bld_xpert_pos)) %>%
  mutate(index_test =
```

```r
          c("sputum_Xpert", "Alere_LAM", "blood_Xpert")) %>%
  select(index_test, everything()) -> yield_table_cd4

bind_rows(
  yield_function(hb_df$anyTBtest_pos, hb_df$sputum_xpert),
  yield_function(hb_df$anyTBtest_pos, hb_df$ALERE_FC),
  yield_function(hb_df$anyTBtest_pos, hb_df$bld_xpert_pos)) %>%
  mutate(index_test =
          c("sputum_Xpert", "Alere_LAM", "blood_Xpert")) %>%
  select(index_test, everything()) -> yield_table_hb

bind_rows(
  yield_function(lact_df$anyTBtest_pos, lact_df$sputum_xpert),
  yield_function(lact_df$anyTBtest_pos, lact_df$ALERE_FC),
  yield_function(lact_df$anyTBtest_pos, lact_df$bld_xpert_pos)) %>%
  mutate(index_test =
          c("sputum_Xpert", "Alere_LAM", "blood_Xpert")) %>%
  select(index_test, everything()) -> yield_table_lact

bind_rows(
  yield_function(died_df$anyTBtest_pos, died_df$sputum_xpert),
  yield_function(died_df$anyTBtest_pos, died_df$ALERE_FC),
  yield_function(died_df$anyTBtest_pos, died_df$bld_xpert_pos)) %>%
  mutate(index_test =
          c("sputum_Xpert", "Alere_LAM", "blood_Xpert")) %>%
  select(index_test, everything()) -> yield_table_died

kable(yield_table1,
     "latex", booktabs=T)
```

index_test	N	n_TB	n_true_positive	diag_yield	CI_95
sputum_Xpert	582	447	275	0.62	0.57 to 0.66
Alere_LAM	582	447	190	0.43	0.38 to 0.47
blood_Xpert	582	447	165	0.37	0.32 to 0.42

4.3.1.1 CD4 < 100

```r
kable(yield_table_cd4,
     "latex", booktabs=T)
```

index_test	N	n_TB	n_true_positive	diag_yield	CI_95
sputum_Xpert	374	300	185	0.62	0.56 to 0.67
Alere_LAM	374	300	152	0.51	0.45 to 0.56
blood_Xpert	374	300	148	0.49	0.44 to 0.55

4.3.1.2 Haemoglobin < 8

```r
kable(yield_table_hb,
     "latex", booktabs=T)
```

index_test	N	n_TB	n_true_positive	diag_yield	CI_95
sputum_Xpert	205	177	112	0.63	0.56 to 0.7
Alere_LAM	205	177	92	0.52	0.44 to 0.59
blood_Xpert	205	177	82	0.46	0.39 to 0.54

4.3.1.3 Lactate > 2.5

```
kable(yield_table_lact,
     "latex", booktabs=T)
```

index_test	N	n_TB	n_true_positive	diag_yield	CI_95
sputum_Xpert	140	122	67	0.55	0.46 to 0.64
Alere_LAM	140	122	52	0.43	0.34 to 0.52
blood_Xpert	140	122	60	0.49	0.4 to 0.58

4.3.1.4 Those who died by 12 weeks

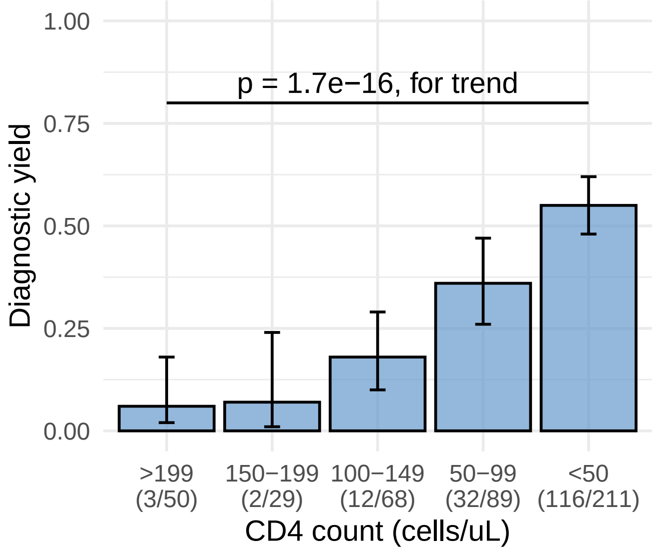
```
kable(yield_table_died,
     "latex", booktabs=T)
```

index_test	N	n_TB	n_true_positive	diag_yield	CI_95
sputum_Xpert	123	96	53	0.55	0.45 to 0.65
Alere_LAM	123	96	43	0.45	0.35 to 0.55
blood_Xpert	123	96	51	0.53	0.43 to 0.63

4.4 Diagnostic yield figures - a few options

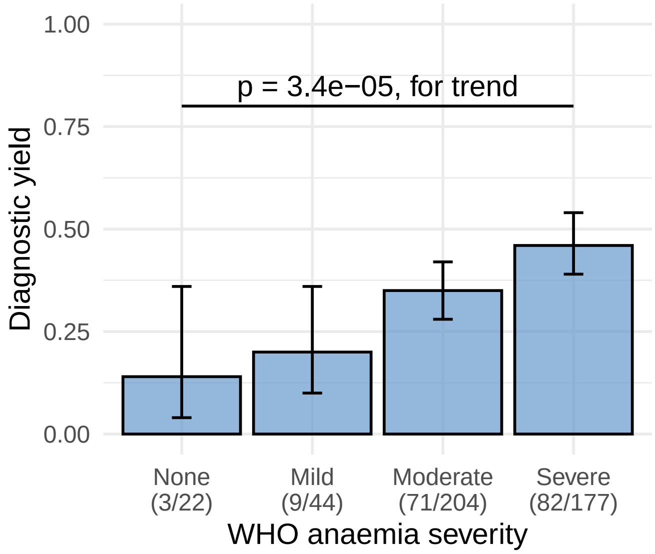
4.4.1 Re-creating the Steve Lawn figure

```
# We can re-use the "yield function" from above, but apply it to sub-setted data by CD4 count
# have run the function 5 times on each CD4 subset (bin)
# and bind teh resulst together as rows of a new data frame
# at the end also add ("mutate") a few new variables which will help make the later plot
bind_rows(
  yield_function(df$anyTBtest_pos[df$CD4>=200],
                 df$bld_xpert_pos[df$CD4>=200]),

  yield_function(df$anyTBtest_pos[df$CD4<200 & df$CD4>=150],
                 df$bld_xpert_pos[df$CD4<200 & df$CD4>=150]),

  yield_function(df$anyTBtest_pos[df$CD4<150 & df$CD4>=100],
                 df$bld_xpert_pos[df$CD4<150 & df$CD4>=100]),

  yield_function(df$anyTBtest_pos[df$CD4<100 & df$CD4>=50],
                 df$bld_xpert_pos[df$CD4<100 & df$CD4>=50]),

  yield_function(df$anyTBtest_pos[df$CD4<50],
                 df$bld_xpert_pos[df$CD4<50])) %>%
  mutate(CD4_bin = c(">199", "150-199", "100-149", "50-99", "<50"),
         no_obs = paste0("(",n_true_positive,"/",n_TB,")"),
         lwr_95 = as.numeric(sapply(strsplit(CI_95, " to "), '[', 1)),
         upr_95 = as.numeric(sapply(strsplit(CI_95, " to "), '[', 2))) -> cd4_tbl

# pull out the x axis tick labels and format them same as the lawn figure
x_axis_labels <- paste0(cd4_tbl$CD4_bin, "\n", cd4_tbl$no_obs)

# get the p value for the Chi squared test for trend
pvalue <- paste0(
  "p = ",
  signif(prop.trend.test(x = cd4_tbl$n_true_positive,
                n = cd4_tbl$n_TB)$p.value, digits=2),
  ", for trend")

# make sure R knows we want these in the correct order on the plot
cd4_tbl$CD4_bin <- factor(cd4_tbl$CD4_bin, levels = c(">199", "150-199", "100-149", "50-99", "<50"))

# plot
ggplot(cd4_tbl, aes(CD4_bin, diag_yield)) +
  geom_bar(stat = "identity",
           colour="black", fill="#6699CC", alpha=0.7) +
  geom_errorbar(aes(ymin=lwr_95, ymax=upr_95),
                width=0.15) +
  theme_minimal() +
  scale_x_discrete(labels=x_axis_labels) +
  xlab("CD4 count (cells/uL)") +
  ylab("Diagnostic yield") +
  ylim(0,1) +
  annotate("text", x=3, y=0.85, label=pvalue) +
  annotate("segment", x = 1, xend = 5, y = 0.8, yend = 0.8)
```



```
### Now repeating for haemoglobin

# first make a haemoglobin classification as per WHO
df$anaemia <- "foo"
df$anaemia[df$Sex=="M" & df$Haemoglobin>=13] <- "None"
df$anaemia[df$Sex=="F" & df$Haemoglobin>=12] <- "None"
```

```
df$anaemia[df$Sex=="M" & df$Haemoglobin<13 & df$Haemoglobin>=11] <- "Mild"
df$anaemia[df$Sex=="F" & df$Haemoglobin<12 & df$Haemoglobin>=11] <- "Mild"
df$anaemia[df$Haemoglobin<11 & df$Haemoglobin>=8] <- "Moderate" #both sexes
df$anaemia[df$Haemoglobin<8] <- "Severe" #both sexes

bind_rows(
  yield_function(df$anyTBtest_pos[df$anaemia=="None"],
               df$bld_xpert_pos[df$anaemia=="None"]),

  yield_function(df$anyTBtest_pos[df$anaemia=="Mild"],
               df$bld_xpert_pos[df$anaemia=="Mild"]),

  yield_function(df$anyTBtest_pos[df$anaemia=="Moderate"],
               df$bld_xpert_pos[df$anaemia=="Moderate"]),

  yield_function(df$anyTBtest_pos[df$anaemia=="Severe"],
               df$bld_xpert_pos[df$anaemia=="Severe"])) %>%

  mutate(hb_bin = c("None", "Mild", "Moderate", "Severe"),
         no_obs = paste0("(",n_true_positive,"/",n_TB,")"),
         lwr_95 = as.numeric(sapply(strsplit(CI_95, " to "), '[', 1)),
         upr_95 = as.numeric(sapply(strsplit(CI_95, " to "), '[', 2))) -> hb_tbl

# pull out the x axis tick labels and format them same as the lawn figure
x_axis_labels <- paste0(hb_tbl$hb_bin, "\n", hb_tbl$no_obs)

# get the p value for the Chi squared test for trend
pvalue <- paste0(
  "p = ",
  signif(prop.trend.test(x = hb_tbl$n_true_positive,
                n = hb_tbl$n_TB)$p.value, digits=2),
  ", for trend")

# make sure R knows we want these in the correct order on the plot
hb_tbl$hb_bin <- factor(hb_tbl$hb_bin,
                          levels = c("None", "Mild", "Moderate", "Severe"))

# plot
ggplot(hb_tbl, aes(hb_bin, diag_yield)) +
  geom_bar(stat = "identity",
           colour="black", fill="#6699CC", alpha=0.7) +
  geom_errorbar(aes(ymin=lwr_95, ymax=upr_95),
                width=0.15) +
  theme_minimal() +
  scale_x_discrete(labels=x_axis_labels) +
  xlab("WHO anaemia severity") +
  ylab("Diagnostic yield") +
  ylim(0,1) +
  annotate("text", x=2.5, y=0.85, label=pvalue) +
  annotate("segment", x = 1, xend = 4, y = 0.8, yend = 0.8)
```



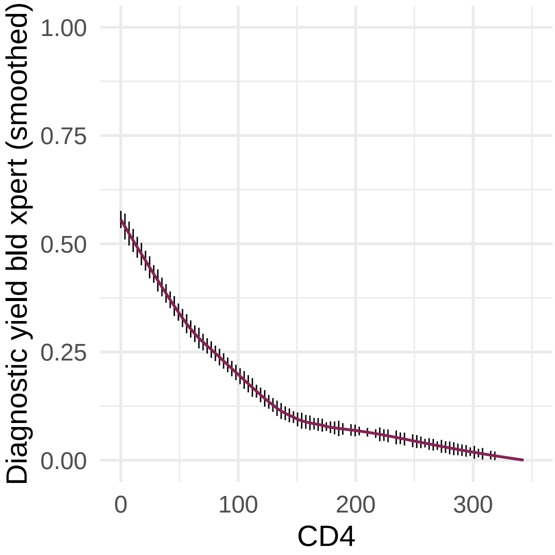
4.4.2 Alternative plots

The categories in plots above are arbitrary, and imbalanced (some have few patients, others many). This is inefficient use of the data for estimating precision and shape of relationship between predictor (CD4, Hb…) and diagnostic yield. Also, if you want to show more than one test (eg compare sputum and blood Xpert, or combinations), this requires more plots. Alternative is to model the realtionship as two continuous variables, rather than binning the predictor variable into ordered categories. Some examples shown below for illustration so we can discuss.

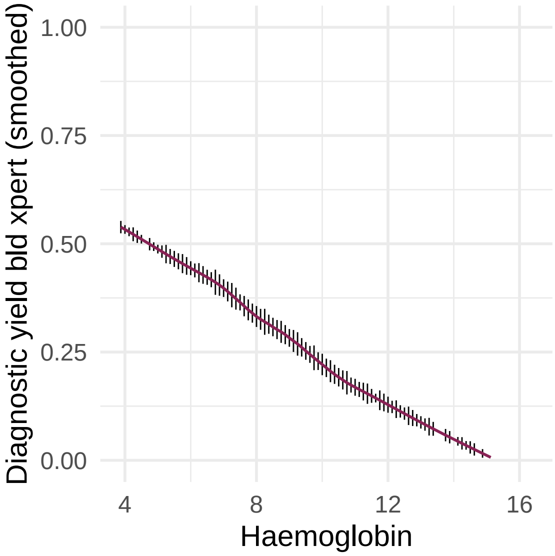
4.4.2.1 Frank Harrell's Hmisc package has these "spike histogram" plots

The red line is the smoothed (Loess) relationship between CD4 (or Hb) and diagnostic yield of blood Xpert. The little back verticle lines "spikes" are histogram of frequencies at different CD4 counts, giving an idea ho wwell the line is supported by data in agiven range.

```
df$bld_xpert_diagnosed <- df$bld_xpert_pos=="MTB" & !is.na(df$bld_xpert_pos)

ggplot(df, aes(CD4, bld_xpert_diagnosed, colour=as.factor(1))) +
  histSpikeg(bld_xpert_diagnosed ~ CD4, lowess=TRUE,
             data=df) +
  ylim(0,1) +
  theme_minimal() +
  scale_colour_manual(values = "#882255") +
  theme(legend.position = "none") +
  ylab("Diagnostic yield bld xpert (smoothed)")
```



```
ggplot(df, aes(Haemoglobin, bld_xpert_diagnosed, colour=as.factor(1))) +
  histSpikeg(bld_xpert_diagnosed ~ Haemoglobin, lowess=TRUE,
             data=df) +
  ylim(0,1) +
  theme_minimal() +
  scale_colour_manual(values = "#882255") +
  theme(legend.position = "none") +
  ylab("Diagnostic yield bld xpert (smoothed)")
```



4.4.2.2 Similar idea but replace the spikes with conf intervals for the model

Takes us a step further from the raw data but gives more flexibility in presentation. Still using a Loess smoothing function for the model, similar to the Harrell plots.

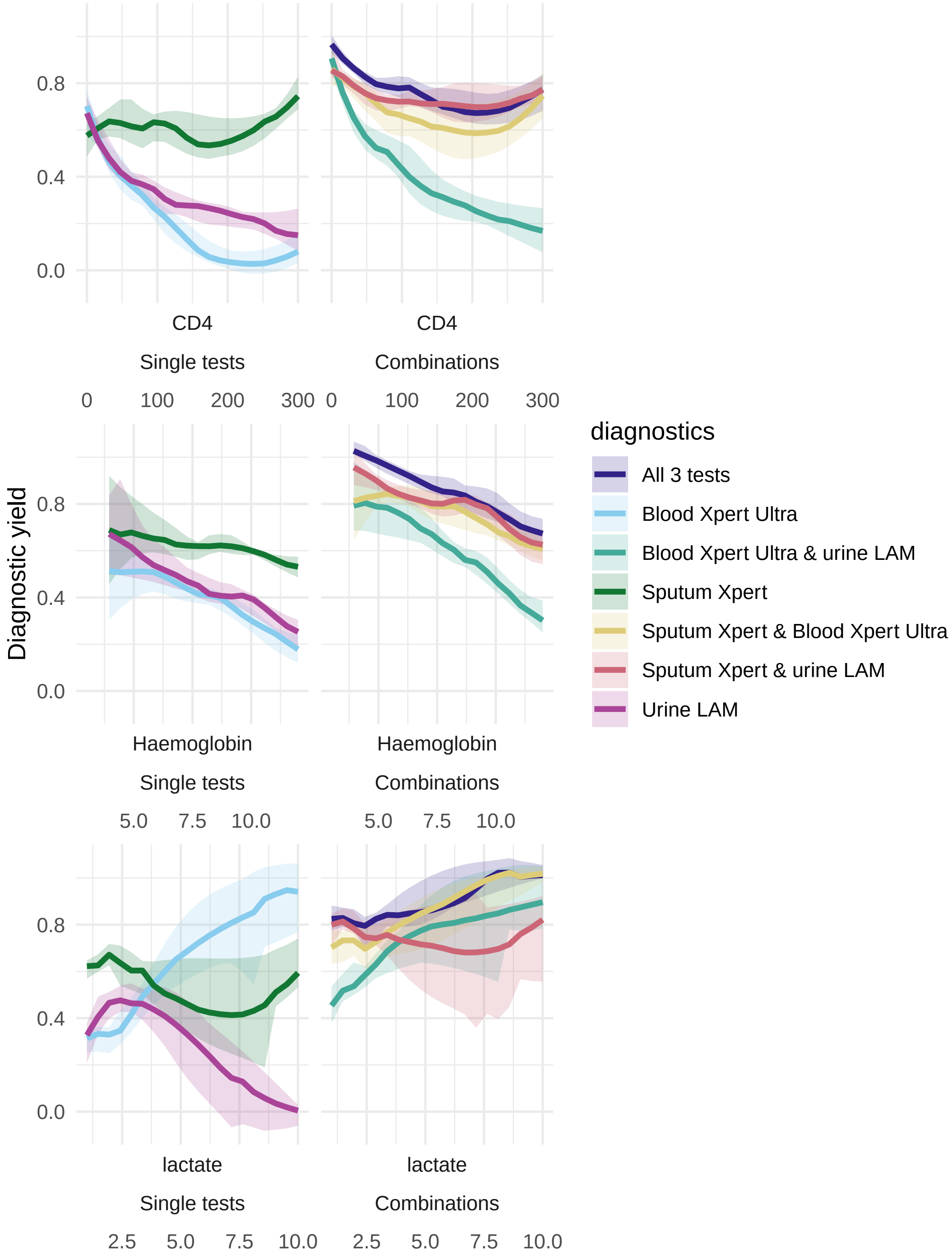
For each of the three diagnostic tests (left column) and four test combinations (right column) the diagnostic yield is modelled by a a dependent variable (CD4, haemoglobin, lactate; top, middle and bottom row) with a loess smoothing function. 95% confidence intervals derived from 1000 bootstraps of each model ((3 tests + 4 combos) * 3 dependent variables = 21 models, each bootstrapped 1000 times).

```r
# a dataframe with just the TB patients (as per diagnostic yield analysis definition)
dftb <- df[df$anyTBtest_pos, ] # n=447

# set up the dummy variables
dftb$sputum_xpert_diagnosed <- dftb$sputum_xpert=="MTB" & !is.na(dftb$sputum_xpert)
dftb$urine_LAM_diagnosed <- dftb$ALERE_FC=="MTB" & !is.na(dftb$ALERE_FC)
# COMBINATIONS
dftb$sputum_or_ulam <-
  dftb$sputum_xpert_diagnosed | dftb$urine_LAM_diagnosed
dftb$sputum_or_bldx <-
  dftb$sputum_xpert_diagnosed | dftb$bld_xpert_diagnosed
dftb$bldx_or_ulam <-
  dftb$urine_LAM_diagnosed | dftb$bld_xpert_diagnosed
dftb$sputum_ulam_bldx <-
  dftb$sputum_xpert_diagnosed |
  dftb$urine_LAM_diagnosed |
  dftb$bld_xpert_diagnosed


### BOOTSTRAPPING RESULTS

# a new data frame to get predictions on
newdata <- data.frame(CD4 = seq(0,300,length.out = 20),
                      lactate = seq(1, 10, length.out = 20),
                      Haemoglobin = seq(3, 12, length.out = 20))

# Funcion to apply in the bootstrap
f1 <- function(data, indicies, formula, span = 0.8, newdata){
  d <- data[indicies,]
  m <- loess(formula, data=d, span = span)
  preds <- predict(m, newdata=newdata, type = "response")
  return(preds)
}

# function to summaries the bootstrap results
sumBoot <- function(boot_data) {
  return(
    data.frame(lwr = apply(boot_data, 2,
                           function(x) as.numeric(
                             quantile(x, probs=0.025, na.rm = TRUE))),
               fit = apply(boot_data, 2,
                           function(x) as.numeric(
                             quantile(x, probs=0.5, na.rm = TRUE))),
               upr = apply(boot_data, 2,
                           function(x) as.numeric(
                             quantile(x, probs=0.975, na.rm = TRUE)))
    )
  )
}
```

```
set.seed(2212)

# run and summarise boot for each model we need,
# bind them all into one dataframe

### CD4

bind_rows(

  sumBoot(
  boot(data=dftb, statistic = f1,
     formula = bld_xpert_diagnosed ~ CD4,
     newdata=newdata,
     R=10)$t),

  sumBoot(
  boot(data=dftb, statistic = f1,
     formula = sputum_xpert_diagnosed ~ CD4,
     newdata=newdata,
     R=10)$t),

  sumBoot(
  boot(data=dftb, statistic = f1,
     formula = urine_LAM_diagnosed ~ CD4,
     newdata=newdata,
     R=10)$t),

  sumBoot(
  boot(data=dftb, statistic = f1,
     formula = sputum_or_ulam ~ CD4,
     newdata=newdata,
     R=10)$t),

  sumBoot(
  boot(data=dftb, statistic = f1,
     formula = sputum_or_bldx ~ CD4,
     newdata=newdata,
     R=10)$t),

  sumBoot(
  boot(data=dftb, statistic = f1,
     formula = bldx_or_ulam ~ CD4,
     newdata=newdata,
     R=10)$t),

  sumBoot(
  boot(data=dftb, statistic = f1,
     formula = sputum_ulam_bldx ~ CD4,
     newdata=newdata,
     R=10)$t)

  ) -> boot_cd4
```

```
boot_cd4$value <- rep(seq(0,300,length.out = 20), 7)

boot_cd4$diagnostic <- rep(
  c("Bld Xpert",
  "Sptm Xpert",
  "uLAM",
  "Sptm Xpert + uLAM",
  "Sptm Xpert + Bld Xpert",
  "Bld Xpert + uLAM",
  "All 3 tests"),
  each=20
)

boot_cd4$var <- rep("CD4", nrow(boot_cd4))

boot_cd4$panel <- factor(
  c(rep("Single tests", 3*20), rep("Combinations", 4*20)),
  levels = c("Single tests", "Combinations"))


### hb

bind_rows(

  sumBoot(
  boot(data=dftb, statistic = f1,
     formula = bld_xpert_diagnosed ~ Haemoglobin,
     newdata=newdata,
     R=10)$t),

  sumBoot(
  boot(data=dftb, statistic = f1,
     formula = sputum_xpert_diagnosed ~ Haemoglobin,
     newdata=newdata,
     R=10)$t),

  sumBoot(
  boot(data=dftb, statistic = f1,
     formula = urine_LAM_diagnosed ~ Haemoglobin,
     newdata=newdata,
     R=10)$t),

  sumBoot(
  boot(data=dftb, statistic = f1,
     formula = sputum_or_ulam ~ Haemoglobin,
     newdata=newdata,
     R=10)$t),

  sumBoot(
  boot(data=dftb, statistic = f1,
     formula = sputum_or_bldx ~ Haemoglobin,
     newdata=newdata,
     R=10)$t),
  sumBoot(
  boot(data=dftb, statistic = f1,
     formula = bldx_or_ulam ~ Haemoglobin,
     newdata=newdata,
     R=10)$t),

  sumBoot(
  boot(data=dftb, statistic = f1,
     formula = sputum_ulam_bldx ~ Haemoglobin,
     newdata=newdata,
     R=10)$t)

  ) -> boot_Haemoglobin

boot_Haemoglobin$value <- rep(seq(3,12,length.out = 20), 7)

boot_Haemoglobin$diagnostic <- rep(
  c("Bld Xpert",
  "Sptm Xpert",
  "uLAM",
  "Sptm Xpert + uLAM",
  "Sptm Xpert + Bld Xpert",
  "Bld Xpert + uLAM",
  "All 3 tests"),
  each=20
)

boot_Haemoglobin$var <- rep("Haemoglobin", nrow(boot_Haemoglobin))

boot_Haemoglobin$panel <- factor(
  c(rep("Single tests", 3*20), rep("Combinations", 4*20)),
  levels = c("Single tests", "Combinations"))

### LACTATE

bind_rows(

  sumBoot(
  boot(data=dftb, statistic = f1,
     formula = bld_xpert_diagnosed ~ lactate,
     newdata=newdata,
     R=10)$t),

  sumBoot(
  boot(data=dftb, statistic = f1,
     formula = sputum_xpert_diagnosed ~ lactate,
     newdata=newdata,
     R=10)$t),

  sumBoot(
  boot(data=dftb, statistic = f1,
     formula = urine_LAM_diagnosed ~ lactate,
     newdata=newdata,
     R=10)$t),

  sumBoot(
  boot(data=dftb, statistic = f1,
     formula = sputum_or_ulam ~ lactate,
     newdata=newdata,
     R=10)$t),

  sumBoot(
  boot(data=dftb, statistic = f1,
     formula = sputum_or_bldx ~ lactate,
     newdata=newdata,
     R=10)$t),

  sumBoot(
  boot(data=dftb, statistic = f1,
     formula = bldx_or_ulam ~ lactate,
     newdata=newdata,
     R=10)$t),

  sumBoot(
  boot(data=dftb, statistic = f1,
     formula = sputum_ulam_bldx ~ lactate,
     newdata=newdata,
     R=10)$t)

  ) -> boot_lactate

boot_lactate$value <- rep(seq(1,10,length.out = 20), 7)


boot_lactate$diagnostic <- rep(
  c("Bld Xpert",
  "Sptm Xpert",
  "uLAM",
  "Sptm Xpert + uLAM",
  "Sptm Xpert + Bld Xpert",
  "Bld Xpert + uLAM",
  "All 3 tests"),
  each=20
)

boot_lactate$var <- rep("lactate", nrow(boot_lactate))

boot_lactate$panel <- factor(
  c(rep("Single tests", 3*20), rep("Combinations", 4*20)),
  levels = c("Single tests", "Combinations"))

boot_df <- bind_rows(boot_cd4, boot_Haemoglobin, boot_lactate)
```

```
boot_df %>%
  mutate(diagnostics = case_when(
    diagnostic == "Bld Xpert" ~ "Blood Xpert Ultra",
    diagnostic == "Sptm Xpert"~ "Sputum Xpert",
    diagnostic == "uLAM" ~ "Urine LAM",
    diagnostic == "Sptm Xpert + uLAM" ~ "Sputum Xpert & urine LAM",
    diagnostic == "Sptm Xpert + Bld Xpert" ~ "Sputum Xpert & Blood Xpert Ultra",
    diagnostic == "Bld Xpert + uLAM" ~ "Blood Xpert Ultra & urine LAM",
    diagnostic == "All 3 tests" ~ "All 3 tests"
  )) %>%
  ggplot(aes(value, fit, colour=diagnostics)) +
  geom_ribbon(aes(ymin=lwr, ymax=upr, fill=diagnostics),
              alpha=0.2, linetype=0) +
  geom_line(size=1.2) +
  theme_minimal() +
  scale_colour_ptol() + scale_fill_ptol() +
  facet_wrap(var~panel, scales = "free_x", nrow = 3, switch = "x") +
  ylab("Diagnostic yield") + xlab("")
```



```
#bld_xpert_diagnosed
#sputum_xpert_diagnosed
#urine_LAM_diagnosed
#sputum_or_ulam
#sputum_or_bldx
```

```r
#bldx_or_ulam
#sputum_ulam_bldx
```

Blood Xpert has similar performance to Alere urine LAM, particularly at CD4<100. It could be argued that in (inpatient with advanced disease) settings without access to LAM but with Xpert available, blood Xpert could be used as a substitute for LAM, as an additional diagnostic on top of sputum Xpert.

Blood Xpert seems to perform particularly well, substantially better than LAM, in patients with raised lactate (although note that the confidence intervals widen at lactate > 5: it isn't as well supported by the data, there are less patients/observations in that area). Wonder why LAM performance falls off at higher lactate? Its against the normal trend of sicker patient = better LAM performance…).

Are the plots too busy / crowded? We could remove one from the right column eg "all three tests"?

5 Blood Xpert cycle threshold & mortality risk

*This analysis is limited to patients with confirmed TB, defined using the "any TB test positive" variable from earlier (can use alternative definition - to discuss)

5.1 Imputing CT values for trace positive samples

To determine the semi-quantitative readout result ("very low", "low", "medium", "high"), Xpert software uses the minimum CT value from the 4 rpoB probes when reporting a positive sample. Trace positive samples are those where the IS1081_IS6110 probe was positive but all rpoB probes negative. Since IS1081_IS6110 is multi-copy per genome, it may not be reliable as rpoB CT values to quantify bacilli. However, as shown below correlation between minimum rpoB CT value and IS1081_IS6110 CT value is quite strong. Therefore, we use IS1081_IS6110 CT value to impute the unobserved minimum rpoB CT value in 'trace' positive samples. This is useful because a lot of our blood Xperts are trace positive.

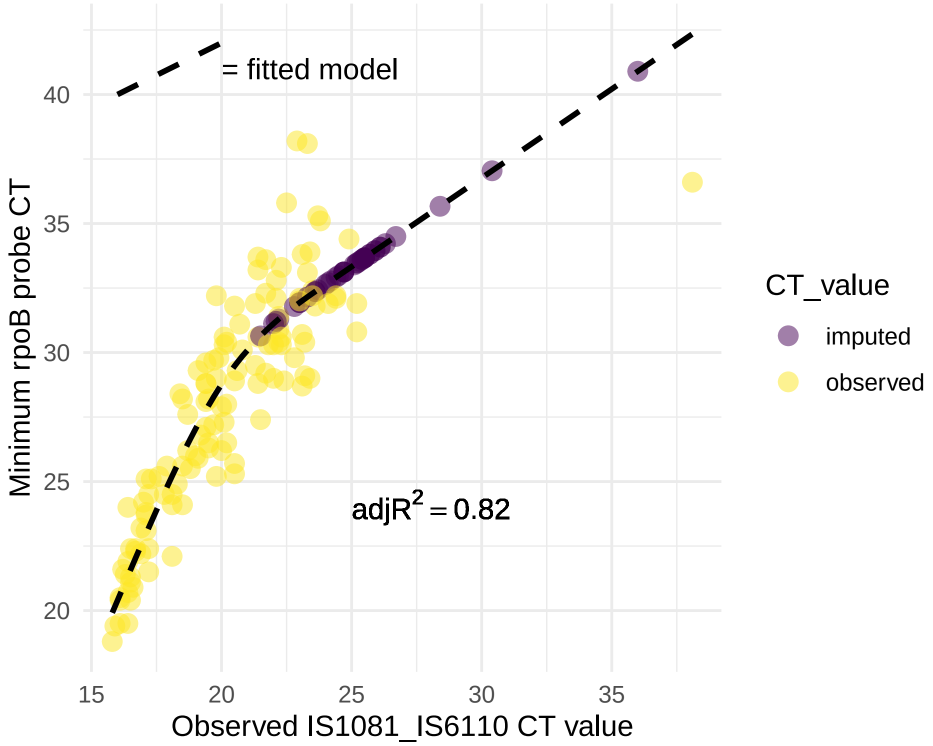
```r
kable(table(df$blood_Xpert_MTB, useNA = "always", deparse.level = 2),
     "latex", booktabs=T)
```

df.blood_Xpert_MTB	Freq
Error	4
High	1
Low	34
Medium	32
Negative	413
Trace	41
Very low	57
NA	0

```r
# missing values are coded as zero - fix this
df$rpoB1[df$rpoB1==0] <- NA
df$rpoB2[df$rpoB2==0] <- NA
df$rpoB3[df$rpoB3==0] <- NA
df$rpoB4[df$rpoB4==0] <- NA
df$IS1081_IS6110[df$IS1081_IS6110==0] <- NA

# get the minimum rpoB probe CT value
df %>%
  rowwise() %>%
  mutate(min_rpoB_CT =
           min(rpoB1, rpoB2, rpoB3, rpoB4, na.rm = TRUE)) -> df
# quick fix the 'infinite' values which NAs have been turned into
df$min_rpoB_CT[is.infinite(df$min_rpoB_CT)] <- NA

# exclude extreme outliers
foo <- df[df$IS1081_IS6110<35,]
# fit a model to the data (using a restricted cubic spline model - its just a curved line that allows f
fit1 <- lm(min_rpoB_CT ~ rcs(IS1081_IS6110,c(16,20,24)), data=foo)
rm(foo)
# get the model fit statistic for later use
R2 <- round(summary(fit1)$adj.r.squared, 2)
# use teh model to get predicted values of minimum rpoB CT
imputed_CT <- predict(fit1, newdata = df)

# call this something simpler
df$blood_Xpert_CT <- df$min_rpoB_CT

# this is just to help with the later graph
df$CT_value <- NA
df$CT_value[is.na(df$min_rpoB_CT) & !is.na(df$IS1081_IS6110)] <- "imputed"
df$CT_value[!is.na(df$min_rpoB_CT)] <- "observed"

# samples which dont have a rpoB Ct value but do have an IS1081_IS6110 CT value
# (which are trace positive samples) get an imputed Ct value, to make our final
# CT value "blood_xpert_CT"
df$blood_Xpert_CT[df$CT_value=="imputed" & !is.na(df$CT_value)] <- imputed_CT[df$CT_value=="imputed" &

df$imputed_CT <- imputed_CT
```

Here are the observed IS1081_IS6110 CT values versus the minimum rpoB CT values, either oberved (yellow points) or imputed (purple) using the model. The model fit is shown as dashed line (all imputed points therefore lie on this line).

```
ggplot(df[!is.na(df$IS1081_IS6110), ],
       aes(IS1081_IS6110, blood_Xpert_CT)) +
  geom_point(aes(colour=CT_value), alpha=0.5, size=3) +
  geom_line(aes(IS1081_IS6110, imputed_CT), size=1, linetype=2) +
  theme_minimal() +
  scale_colour_viridis_d() +
  xlab("Observed IS1081_IS6110 CT value") +
  ylab("Minimum rpoB probe CT") +
  annotate("segment", x = 16, xend = 20, y = 40, yend = 42, size=1, linetype=2) +
  annotate("text", x=20, y=41, label="= fitted model", hjust=0) +
  geom_text(aes(25,24, label=(paste0("adjR^2 == ", R2))),parse = TRUE, hjust=0)
```



Using these values for all subsequent analysis.

5.2 Visualising blood Xpert CT v mortality risk

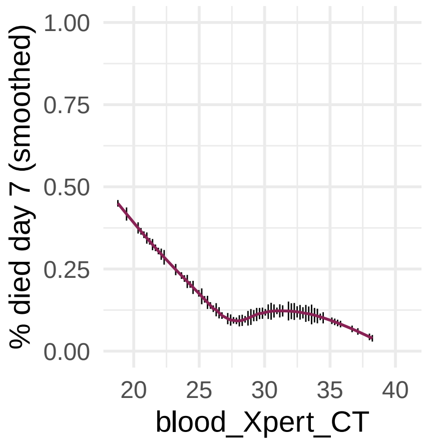
5.2.1 CT values treated as continuous variable

```
# Set up "end date" correctly for KM plots etc
df$dateDeath.x <- as.Date(df$dateDeath.x, format="%Y-%m-%d")
df$LTFU.censor.date <- as.Date(df$LTFU.censor.date, format="%d/%m/%Y")
df$StudyDate  <- as.Date(df$StudyDate, format="%Y-%m-%d")

# these are miscoded as mssing - not sure why
df$dateDeath.x[df$UID==480] <- "2016-01-26"
df$dateDeath.x[df$UID==485] <- "2016-03-15"
df$dateDeath.x[df$UID==490] <- "2016-02-14"
df$dateDeath.x[df$UID==498] <- "2016-02-09"
df$dateDeath.x[df$UID==263] <- "2015-06-02"

df$endDate <- as.Date("1900-01-01")
df$endDate[df$survival.12weeks=="LTFU"] <-
  df$LTFU.censor.date[df$survival.12weeks=="LTFU"]

df$endDate[df$survival.12weeks=="Survived"] <-
  df$StudyDate[df$survival.12weeks=="Survived"] + 84

df$endDate[df$survival.12weeks=="Died"] <-
  df$dateDeath.x[df$survival.12weeks=="Died"]
```

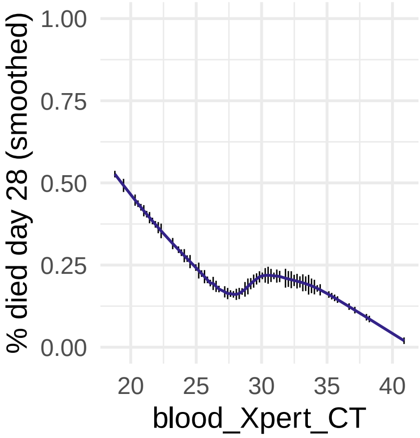
```r
# follow up time variable
df$time <- as.numeric(df$endDate - df$StudyDate)

# make 7, 28, and 84 day death variables
df$day7outcome <- "Survived"
df$day7outcome[df$time<8 & df$survival.12weeks=="Died"] <- "Died"
df$day7outcome[df$time<8 & df$survival.12weeks=="LTFU"] <- "LTFU"
df$day7death <- df$day7outcome=="Died"
df$day7death[df$day7outcome=="LTFU"] <- NA

df$day28outcome <- "Survived"
df$day28outcome[df$time<29 & df$survival.12weeks=="Died"] <- "Died"
df$day28outcome[df$time<29 & df$survival.12weeks=="LTFU"] <- "LTFU"
df$day28death <- df$day28outcome=="Died"
df$day28death[df$day28outcome=="LTFU"] <- NA

df$day84outcome <- df$survival.12weeks
df$day84death <- df$day84outcome=="Died"
df$day84death[df$day84outcome=="LTFU"] <- NA

# make dataframe with only the confirmed Tb cases

tbdf <- df[df$anyTBtest_pos==TRUE, ]
```

Proportion of patients dying by different time points modelled as a function of blood Xpert Ct value. The coloured line is aloess smoothing function - allowed to be pretty flexible and choose shape best fitting data; number of patients at each CT value shown by the height of the little verticle black lines. These plots are a way of seeing the ‘shape’ of the relationship between CT value and risk of death - not forcing the relationship to be linear (or logit like in logistic regression).

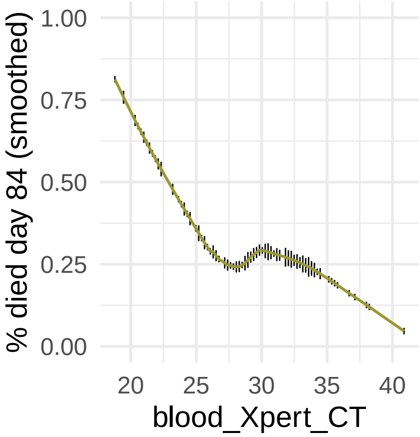
```r
ggplot(tbdf, aes(blood_Xpert_CT, day7death, colour=as.factor(1))) +
  histSpikeg(day7death ~ blood_Xpert_CT, lowess=TRUE,
             data=tbdf) +
  ylim(0,1) +
  theme_minimal() +
  scale_colour_manual(values = "#882255") +
  theme(legend.position = "none") +
  ylab("% died day 7 (smoothed)")
```



```
ggplot(tbdf, aes(blood_Xpert_CT, day28death, colour=as.factor(1))) +
  histSpikeg(day28death ~ blood_Xpert_CT, lowess=TRUE,
             data=tbdf) +
  ylim(0,1) +
  theme_minimal() +
  scale_colour_manual(values = "#332288") +
  theme(legend.position = "none") +
  ylab("% died day 28 (smoothed)")
```



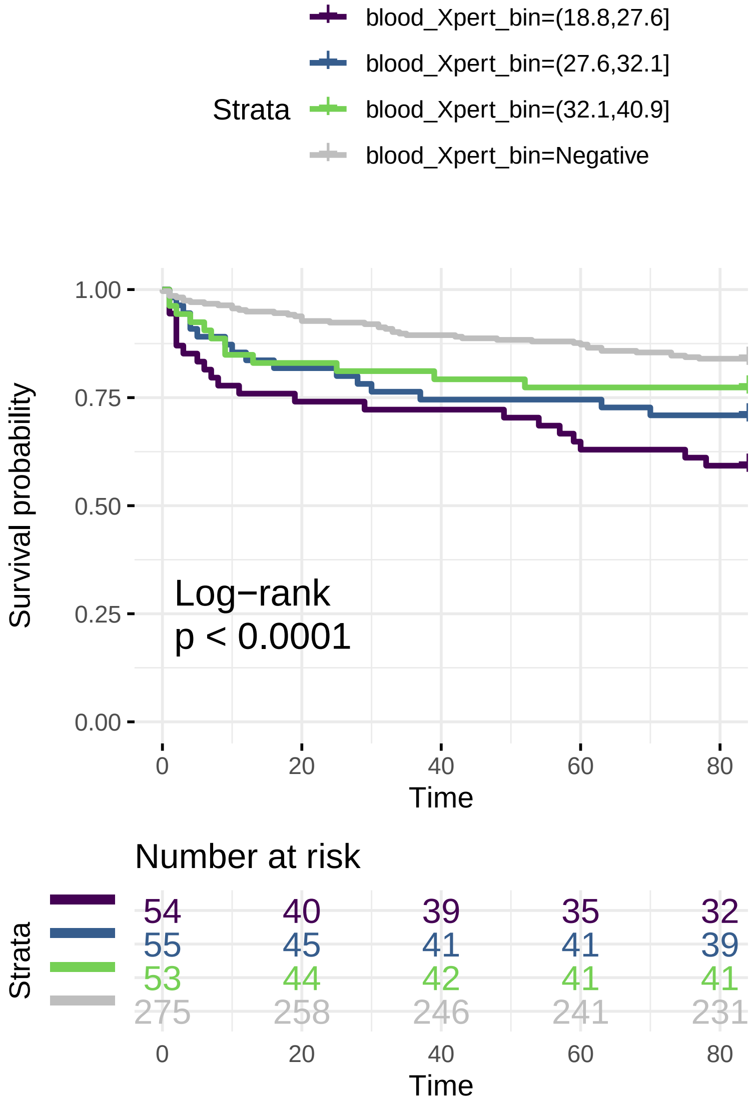
```
ggplot(tbdf, aes(blood_Xpert_CT, day84death, colour=as.factor(1))) +
  histSpikeg(day84death ~ blood_Xpert_CT, lowess=TRUE,
             data=tbdf) +
  ylim(0,1) +
  theme_minimal() +
  scale_colour_manual(values = "#999933") +
  theme(legend.position = "none") +
  ylab("% died day 84 (smoothed)")
```



5.2.2 CT values binned into categories, death as time-to-event (KM plots)

Included are patients with TB confirmed by any TB test. Patients are binned into three equal sized groups by blood Xpert CT value quantile (0.33 and 0.67 quantiles). With the "negative" blood Xpert group this gives 4 patient groups for KM plot.

```
#event <- xpdf$survival.12weeks == "Died"
#blood_Xpert_CT <- xpdf$blood_Xpert_CT

#print(summary(coxph(Surv(time,event) ~ blood_Xpert_CT, method="breslow")))


tbdf$blood_Xpert_bin <-
  as.character(cut(
    tbdf$blood_Xpert_CT,
    breaks = quantile(
      tbdf$blood_Xpert_CT, probs = c(0,0.33,0.67,1), na.rm = TRUE )))
tbdf$blood_Xpert_bin[tbdf$blood_Xpert_MTB=="Negative"] <- "Negative"

y <- Surv(tbdf$time, tbdf$day84death)
km <- survfit(y ~ tbdf$blood_Xpert_bin)

ggsurvplot(km, data = tbdf,
           risk.table = TRUE,
           palette = c("#440154FF", "#365D8DFF", "#75D054FF", "grey"),
           pval = TRUE, pval.method = TRUE,
           ggtheme = theme_minimal(),
           risk.table.col="strata",
           risk.table.y.text=FALSE) +
  guides(colour = guide_legend(nrow = 5))
```



6 Rif resistance detection

6.1 Some exploration of the TB drug sensitivity data

6.1.1 Catogories and concordance

This is complex data. Possible categories of patient, not mutually exclusive:

- Culture positive, DST = fully sensitive MTB
- Culture positive, DST = rif mono-resistant MTB
- Culture positive, DST = MDR or XDR MTB
- Culture negative, Xpert = rif sensitive
- Culture negative, Xpert = rif resistant

Further complicated by possibility of discordant results where DST is different when >1 positive culture, or rif probe is different when >1 positive Xpert.

```
tbdf$RRTB_DSTorXpert <- (
  (!is.na(tbdf$sputumGXP1_RifDST) & tbdf$sputumGXP1_RifDST == "Resistant") |
  (!is.na(tbdf$sputumGXP2_RifDST) & tbdf$sputumGXP2_RifDST == "Resistant") |
  (!is.na(tbdf$sputumGXP3_RifDST) & tbdf$sputumGXP3_RifDST == "Resistant") |

  (!is.na(tbdf$sputumCulture1_MTBDST) &
     (tbdf$sputumCulture1_MTBDST == "MDR" |
      tbdf$sputumCulture1_MTBDST == "Rif.mono" |
      tbdf$sputumCulture1_MTBDST == "XDR")) |
  (!is.na(tbdf$sputumCulture2_MTBDST) &
     (tbdf$sputumCulture2_MTBDST == "MDR" |
      tbdf$sputumCulture2_MTBDST == "Rif.mono" |
      tbdf$sputumCulture2_MTBDST == "XDR")) |
  (!is.na(tbdf$sputumCulture3_MTBDST) &
     (tbdf$sputumCulture3_MTBDST == "MDR" |
      tbdf$sputumCulture3_MTBDST == "Rif.mono" |
      tbdf$sputumCulture3_MTBDST == "XDR")) |

  (!is.na(tbdf$MBC1_MTBDST) &
     (tbdf$MBC1_MTBDST == "MDR" |
      tbdf$MBC1_MTBDST == "Rif.mono" |
      tbdf$MBC1_MTBDST == "XDR")) |
  (!is.na(tbdf$MBC2_MTBDST) &
     (tbdf$MBC2_MTBDST == "MDR" |
      tbdf$MBC2_MTBDST == "Rif.mono" |
      tbdf$MBC2_MTBDST == "XDR")) |
  (!is.na(tbdf$MBC3_MTBDST) &
     (tbdf$MBC3_MTBDST == "MDR" |
      tbdf$MBC3_MTBDST == "Rif.mono" |
      tbdf$MBC3_MTBDST == "XDR")) |

  (!is.na(tbdf$uMTBculture.MTBDST) &
     (tbdf$uMTBculture.MTBDST == "MDR" |
      tbdf$uMTBculture.MTBDST == "Rif.mono" |
      tbdf$uMTBculture.MTBDST == "XDR")) |
  (!is.na(tbdf$otherCul1_MTBDST) &
     (tbdf$otherCul1_MTBDST == "MDR" |
      tbdf$otherCul1_MTBDST == "Rif.mono" |
      tbdf$otherCul1_MTBDST == "XDR")) |
  (!is.na(tbdf$otherCul2_MTBDST) &
     (tbdf$otherCul2_MTBDST == "MDR" |
      tbdf$otherCul2_MTBDST == "Rif.mono" |
      tbdf$otherCul2_MTBDST == "XDR")) |

  (!is.na(tbdf$uGXP.Rifprobe) &
     tbdf$uGXP.Rifprobe == "Resistant") |
  (!is.na(tbdf$otherGXP.refprobe) &
     tbdf$otherGXP.refprobe == "Resistant") |

  (!is.na(tbdf$blood_Xpert_rif) &
    tbdf$blood_Xpert_rif  == "Resistance detected")
)
```

```
tbdf$MDR_XDR_TB <- (

  (!is.na(tbdf$sputumCulture1_MTBDST) &
     (tbdf$sputumCulture1_MTBDST == "MDR" |
      tbdf$sputumCulture1_MTBDST == "XDR")) |
  (!is.na(tbdf$sputumCulture2_MTBDST) &
     (tbdf$sputumCulture2_MTBDST == "MDR" |
      tbdf$sputumCulture2_MTBDST == "XDR")) |
  (!is.na(tbdf$sputumCulture3_MTBDST) &
     (tbdf$sputumCulture3_MTBDST == "MDR" |
      tbdf$sputumCulture3_MTBDST == "XDR")) |

  (!is.na(tbdf$MBC1_MTBDST) &
     (tbdf$MBC1_MTBDST == "MDR" |
      tbdf$MBC1_MTBDST == "XDR")) |
  (!is.na(tbdf$MBC2_MTBDST) &
     (tbdf$MBC2_MTBDST == "MDR" |
      tbdf$MBC2_MTBDST == "XDR")) |
  (!is.na(tbdf$MBC3_MTBDST) &
     (tbdf$MBC3_MTBDST == "MDR" |
      tbdf$MBC3_MTBDST == "XDR")) |

  (!is.na(tbdf$uMTBculture.MTBDST) &
     (tbdf$uMTBculture.MTBDST == "MDR" |
      tbdf$uMTBculture.MTBDST == "XDR")) |
  (!is.na(tbdf$otherCul1_MTBDST) &
     (tbdf$otherCul1_MTBDST == "MDR" |
      tbdf$otherCul1_MTBDST == "XDR")) |
  (!is.na(tbdf$otherCul2_MTBDST) &
     (tbdf$otherCul2_MTBDST == "MDR" |
      tbdf$otherCul2_MTBDST == "XDR"))
  )


tbdf$rif_monoDR <- (
  (!is.na(tbdf$sputumCulture1_MTBDST) &
      tbdf$sputumCulture1_MTBDST == "Rif.mono") |
  (!is.na(tbdf$sputumCulture2_MTBDST) &
      tbdf$sputumCulture2_MTBDST == "Rif.mono") |
  (!is.na(tbdf$sputumCulture3_MTBDST) &
      tbdf$sputumCulture3_MTBDST == "Rif.mono") |

  (!is.na(tbdf$MBC1_MTBDST) &
      tbdf$MBC1_MTBDST == "Rif.mono") |
  (!is.na(tbdf$MBC2_MTBDST) &
      tbdf$MBC2_MTBDST == "Rif.mono") |
  (!is.na(tbdf$MBC3_MTBDST) &
      tbdf$MBC3_MTBDST == "Rif.mono") |

  (!is.na(tbdf$uMTBculture.MTBDST) &
      tbdf$uMTBculture.MTBDST == "Rif.mono") |
  (!is.na(tbdf$otherCul1_MTBDST) &
      tbdf$otherCul1_MTBDST == "Rif.mono") |
```

```r
  (!is.na(tbdf$otherCul2_MTBDST) &
      tbdf$otherCul2_MTBDST == "Rif.mono")
  )


tbdf$INH_monoDR <- (
  (!is.na(tbdf$sputumCulture1_MTBDST) &
      tbdf$sputumCulture1_MTBDST == "INH.mono") |
  (!is.na(tbdf$sputumCulture2_MTBDST) &
      tbdf$sputumCulture2_MTBDST == "INH.mono") |
  (!is.na(tbdf$sputumCulture3_MTBDST) &
      tbdf$sputumCulture3_MTBDST == "INH.mono") |

  (!is.na(tbdf$MBC1_MTBDST) &
      tbdf$MBC1_MTBDST == "INH.mono") |
  (!is.na(tbdf$MBC2_MTBDST) &
      tbdf$MBC2_MTBDST == "INH.mono") |
  (!is.na(tbdf$MBC3_MTBDST) &
      tbdf$MBC3_MTBDST == "INH.mono") |

  (!is.na(tbdf$uMTBculture.MTBDST) &
      tbdf$uMTBculture.MTBDST == "INH.mono") |
  (!is.na(tbdf$otherCul1_MTBDST) &
      tbdf$otherCul1_MTBDST == "INH.mono") |
  (!is.na(tbdf$otherCul2_MTBDST) &
      tbdf$otherCul2_MTBDST == "INH.mono")
  )


tbdf$RH_sensitive <- (
  (!is.na(tbdf$sputumCulture1_MTBDST) &
      tbdf$sputumCulture1_MTBDST == "RH.sensitive") |
  (!is.na(tbdf$sputumCulture2_MTBDST) &
      tbdf$sputumCulture2_MTBDST == "RH.sensitive") |
  (!is.na(tbdf$sputumCulture3_MTBDST) &
      tbdf$sputumCulture3_MTBDST == "RH.sensitive") |

  (!is.na(tbdf$MBC1_MTBDST) &
      tbdf$MBC1_MTBDST == "RH.sensitive") |
  (!is.na(tbdf$MBC2_MTBDST) &
      tbdf$MBC2_MTBDST == "RH.sensitive") |
  (!is.na(tbdf$MBC3_MTBDST) &
      tbdf$MBC3_MTBDST == "RH.sensitive") |

  (!is.na(tbdf$uMTBculture.MTBDST) &
      tbdf$uMTBculture.MTBDST == "RH.sensitive") |
  (!is.na(tbdf$otherCul1_MTBDST) &
      tbdf$otherCul1_MTBDST == "RH.sensitive") |
  (!is.na(tbdf$otherCul2_MTBDST) &
      tbdf$otherCul2_MTBDST == "RH.sensitive")
  )

#tbdf %>%
#  filter(RRTB_DSTorXpert == TRUE &
#           MDR_XDR_TB == FALSE &
#           rif_monoDR == FALSE ) %>%
#  mutate(foo = paste(sputumCulture1_cultureID,
#         sputumCulture2_cultureID,
#         sputumCulture3_cultureID,
#         MBC1_cultureID,
#         MBC2_cultureID,
#         MBC3_cultureID,
#         uMTBculture,
#         otherCul1_cultureID,
#         otherCul2_cultureID, sep="_")) %>%
#  select(foo)
```

In this data set (n=447 with any TB test positive) there are 51 patients with any rif resistance (*any* DST or Xpert probe = rif resistance). Of these, 16 patients have rif mono-resistance by *any* culture DST result, and 27 MDR or XDR patients by *any* culture DST result. This leaves 8 patients who are rif resistant by any Xpert test but either culture negative or sensitive on culture DST. Culture results for these 8 (NA means culture not done or negative):

```
tbdf %>%
  filter(RRTB_DSTorXpert == TRUE &
           MDR_XDR_TB == FALSE &
           rif_monoDR == FALSE ) %>%
  select(
    sputumCulture1_MTBDST,
    sputumCulture2_MTBDST,
    sputumCulture3_MTBDST,
    MBC1_MTBDST,
    MBC2_MTBDST,
    MBC3_MTBDST,
    uMTBculture.MTBDST,
    otherCul1_MTBDST,
    otherCul2_MTBDST) -> foo

foo <- t(foo)
dimnames(foo)[[2]] <- c("1", "2", "3", "4", "5", "6", "7", "8")
kable(foo,
     "latex", booktabs=T)
```

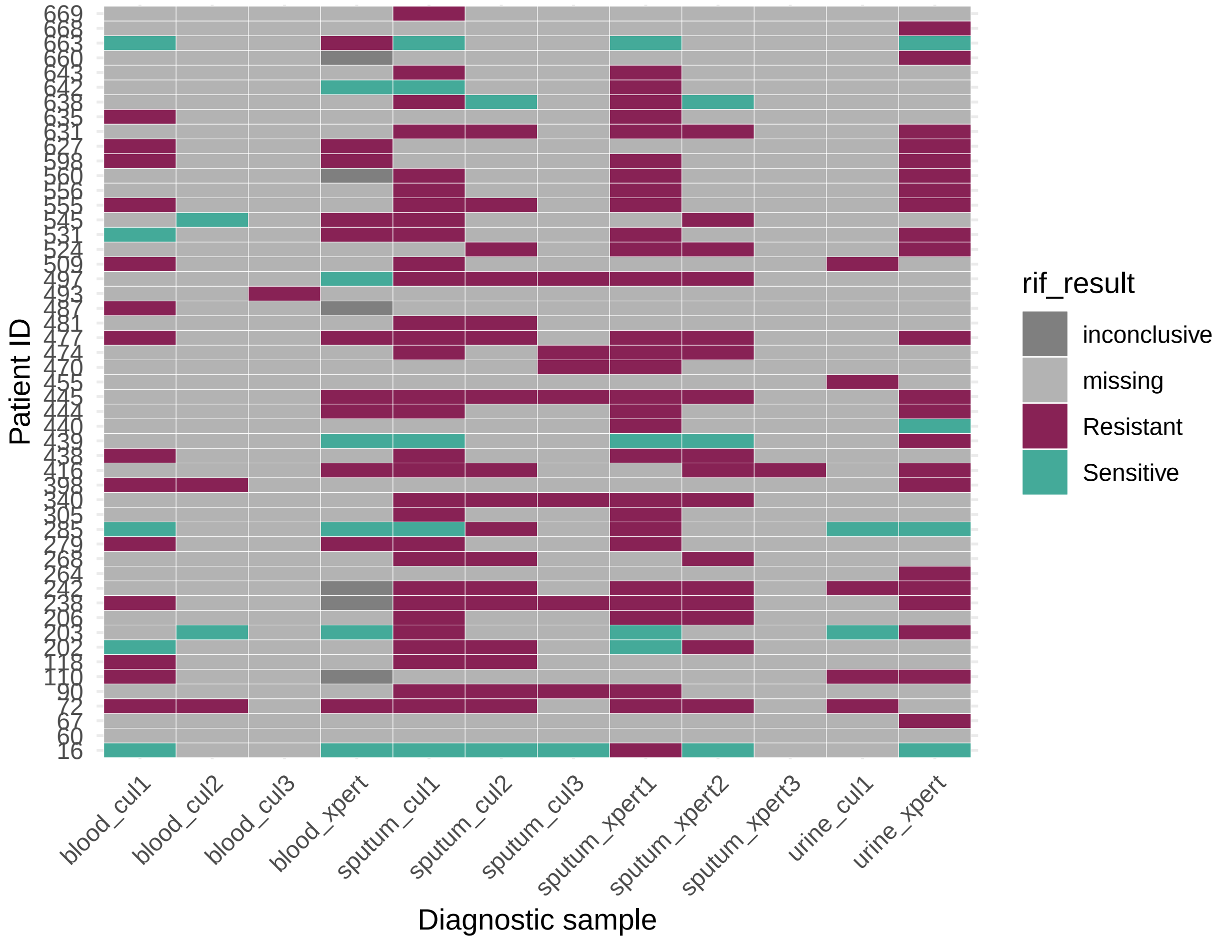
	1	2	3	4	5	6	7	8
sputumCulture1_MTBDST	RH.sensitive	NA	RH.sensitive	NA	RH.sensitive	NA	RH.sensitive	NA
sputumCulture2_MTBDST	RH.sensitive	NA	NA	NA	NA	NA	NA	NA
sputumCulture3_MTBDST	RH.sensitive	NA	NA	NA	NA	NA	NA	NA
MBC1_MTBDST	RH.sensitive	NA	NA	NA	NA	NA	RH.sensitive	NA
MBC2_MTBDST	NA	NA	NA	NA	NA	NA	NA	NA
MBC3_MTBDST	NA	NA	NA	NA	NA	NA	NA	NA
uMTBculture.MTBDST	NA	NA	NA	NA	NA	NA	NA	NA
otherCul1_MTBDST	NA	NA	NA	NA	NA	NA	NA	NA
otherCul2_MTBDST	NA	NA	NA	NA	NA	NA	NA	NA

6.1.2 Discordant sensitivity results

As stated above, in this data set (n=447 with any TB test positive) there are 51 patients with any rif resistance (*any* DST or Xpert probe = rif resistance). In the plot below, these 51 patients are shown, each in a row. All the culture DST are recoded to rif sensitive (inc INH monoresistant) or rif resistant (mono, MDR, XDR) and Rif sensitivity result from diagnostic samples shown in 12 columns.

```
# a bit of messy coding here to harmonise how the results are recorded in each diagnostic
sputum_xpert1 <- tbdf$sputumGXP1_RifDST
sputum_xpert2 <- tbdf$sputumGXP2_RifDST
sputum_xpert3 <- tbdf$sputumGXP3_RifDST

blood_xpert <- tbdf$blood_Xpert_rif
blood_xpert[blood_xpert=="Indeterminate" & !is.na(blood_xpert)] <- "inconclusive"
blood_xpert[blood_xpert=="Not detected" & !is.na(blood_xpert)] <- "Sensitive"
blood_xpert[blood_xpert=="Resistance detected" & !is.na(blood_xpert)] <- "Resistant"

urine_xpert <- tbdf$uGXP.Rifprobe

sputum_cul1 <- tbdf$sputumCulture1_MTBDST
sputum_cul1[(sputum_cul1=="INH.mono" |
              sputum_cul1=="RH.sensitive") &
              !is.na(sputum_cul1)] <- "Sensitive"
sputum_cul1[(sputum_cul1=="MDR" |
              sputum_cul1=="XDR" |
               sputum_cul1=="Rif.mono") &
              !is.na(sputum_cul1)] <- "Resistant"
sputum_cul2 <- tbdf$sputumCulture2_MTBDST
sputum_cul2[(sputum_cul2=="INH.mono" |
              sputum_cul2=="RH.sensitive") &
              !is.na(sputum_cul2)] <- "Sensitive"
sputum_cul2[(sputum_cul2=="MDR" |
              sputum_cul2=="XDR" |
               sputum_cul2=="Rif.mono") &
              !is.na(sputum_cul2)] <- "Resistant"
sputum_cul3 <- tbdf$sputumCulture3_MTBDST
sputum_cul3[(sputum_cul3=="INH.mono" |
              sputum_cul3=="RH.sensitive") &
              !is.na(sputum_cul3)] <- "Sensitive"
sputum_cul3[(sputum_cul3=="MDR" |
              sputum_cul3=="XDR" |
               sputum_cul3=="Rif.mono") &
              !is.na(sputum_cul3)] <- "Resistant"
sputum_cul3[sputum_cul3=="RIF.inconclusive" &
              !is.na(sputum_cul3)] <- NA


blood_cul1 <- tbdf$MBC1_MTBDST
blood_cul1[(blood_cul1=="INH.mono" |
              blood_cul1=="RH.sensitive") &
              !is.na(blood_cul1)] <- "Sensitive"
blood_cul1[(blood_cul1=="MDR" |
              blood_cul1=="XDR" |
               blood_cul1=="Rif.mono") &
              !is.na(blood_cul1)] <- "Resistant"

blood_cul2 <- tbdf$MBC2_MTBDST
blood_cul2[(blood_cul2=="INH.mono" |
              blood_cul2=="RH.sensitive") &
              !is.na(blood_cul2)] <- "Sensitive"
blood_cul2[(blood_cul2=="MDR" |
              blood_cul2=="XDR" |
               blood_cul2=="Rif.mono") &
              !is.na(blood_cul2)] <- "Resistant"

blood_cul3 <- tbdf$MBC3_MTBDST
blood_cul3[(blood_cul3=="INH.mono" |
              blood_cul3=="RH.sensitive") &
              !is.na(blood_cul3)] <- "Sensitive"
blood_cul3[(blood_cul3=="MDR" |
              blood_cul3=="XDR" |
               blood_cul3=="Rif.mono") &
              !is.na(blood_cul3)] <- "Resistant"

other_cul1 <- tbdf$otherCul1_MTBDST
other_cul1[(other_cul1=="INH.mono" |
              other_cul1=="RH.sensitive") &
              !is.na(other_cul1)] <- "Sensitive"
other_cul1[(other_cul1=="MDR" |
              other_cul1=="XDR" |
               other_cul1=="Rif.mono") &
              !is.na(other_cul1)] <- "Resistant"

other_cul2 <- tbdf$otherCul2_MTBDST
other_cul2[(other_cul2=="INH.mono" |
              other_cul2=="RH.sensitive") &
              !is.na(other_cul2)] <- "Sensitive"
other_cul2[(other_cul2=="MDR" |
              other_cul2=="XDR" |
               other_cul2=="Rif.mono") &
              !is.na(other_cul2)] <- "Resistant"

urine_cul1 <- tbdf$uMTBculture.MTBDST
urine_cul1[(urine_cul1=="INH.mono" |
              urine_cul1=="RH.sensitive") &
              !is.na(urine_cul1)] <- "Sensitive"
urine_cul1[(urine_cul1=="MDR" |
              urine_cul1=="XDR" |
               urine_cul1=="Rif.mono") &
              !is.na(urine_cul1)] <- "Resistant"

other_xpert1 <- tbdf$otherGXP.refprobe

# combine them in adata frame, filter out patients with no resistant results, then plot
data.frame(UID = tbdf$UID,
           sputum_cul1, sputum_cul2, sputum_cul3,
           sputum_xpert1, sputum_xpert2, sputum_xpert3,
           urine_xpert, urine_cul1,
           blood_cul1, blood_cul2, blood_cul3, blood_xpert,
           other_cul1, other_cul2, other_xpert1) %>%
  filter_all(any_vars(str_detect(., pattern = "Resistant"))) %>%
  gather(key="Sample", value="rif_result", 2:13) %>%
  replace_na(list(rif_result = "missing")) %>%
  ggplot(aes(Sample, as.factor(UID))) +
  geom_tile(aes(fill=rif_result), colour="white") +
  scale_fill_manual(
    values = c("grey50", "grey70", "#882255", "#44AA99")) +
  labs(x = "Diagnostic sample", y = "Patient ID") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle=45, hjust = 1))
```

```
## Warning: attributes are not identical across measure variables;
## they will be dropped
```



```
n_atleast2 <-  sum(rowSums(!is.na(data.frame(sputum_cul1, sputum_cul2, sputum_cul3,
           sputum_xpert1, sputum_xpert2, sputum_xpert3,
           urine_xpert, urine_cul1,
           blood_cul1, blood_cul2, blood_cul3, blood_xpert,
           other_cul1, other_cul2, other_xpert1)))>1)
```

There are 12 patients who have at least one rif sensitive and at least one rif resistant result (ie a discordant result). This is 12 / 51 (24%) of those with any rif resistance detected, and 12 / 368 (3.3%) patients who have at least 2 test results.

Checking the dates of samples for these 12 patients in case some are from samples long before or after the other samples, it looks like they are from same episodes of care:

```r
data.frame(UID = tbdf$UID,
           sputum_cul1, sputum_cul2, sputum_cul3,
           sputum_xpert1, sputum_xpert2, sputum_xpert3,
           urine_xpert, urine_cul1,
           blood_cul1, blood_cul2, blood_cul3, blood_xpert,
           other_cul1, other_cul2, other_xpert1) %>%
  filter_all(any_vars(str_detect(., pattern = "Resistant"))) %>%
  filter_all(any_vars(str_detect(., pattern = "Sensitive"))) %>%
  select(UID) -> discords

as.numeric(discords$UID) -> discords

kable(tbdf[tbdf$UID %in% discords,
     c("UID", "StudyDate", "MBC1_Date",
       "otherCul1_Date",
       "sputumGXP1_Date", "sputumGXP2_Date",
       "sputumCulture1_Date", "sputumCulture2_Date",

       "uMTBculture.Date")],
     "latex", booktabs=T) %>%
  kable_styling(latex_options = c("scale_down"))
```

UID	StudyDate	MBC1_Date	otherCul1_Date	sputumGXP1_Date	sputumGXP2_Date	sputumCulture1_Date	sputumCulture2_Date	uMTBculture.Date
16	2014-03-01	2014-03-02	NA	2014-03-04	2014-03-07	2014-03-12	2014-03-04	2014-03-03
202	2014-10-21	2014-10-21	2014-10-20	2014-10-22	2014-10-21	2014-10-22	2014-10-21	2014-10-22
203	2014-10-22	2014-10-23	NA	2014-10-24	NA	2014-10-18	NA	2014-10-24
285	2015-03-09	2015-03-08	NA	2015-03-09	NA	2015-03-11	2015-03-09	2015-03-09
439	2015-10-20	2015-10-20	NA	2015-10-20	2015-10-20	2015-10-20	NA	2015-10-29
440	2015-10-20	2015-10-20	NA	2015-10-15	NA	NA	NA	NA
497	2016-02-08	2016-01-29	NA	2016-02-04	2016-02-08	2016-02-04	2016-01-30	NA
531	2016-03-22	2016-03-22	NA	2016-03-22	NA	2016-03-22	NA	NA
545	2016-04-06	2016-04-06	NA	2016-04-06	2016-04-06	2016-04-06	NA	NA
638	2016-08-02	2016-08-02	2016-08-02	2016-07-18	2016-08-02	2016-07-18	2016-08-02	NA
642	2016-08-04	2016-08-06	NA	2016-08-04	NA	2016-08-15	2016-08-04	NA
663	2016-09-19	2016-09-19	NA	2016-09-19	NA	2016-09-19	NA	NA

Is there any obvious pattern to the discordance across sample types? Maybe. In the 12 discordant sets of samples, 7 have a single test which is the "odd one out" (1 test giving different result to all the others), and 4/7 of these are from blood compartment:

```r
data.frame(UID = tbdf$UID,
           sputum_cul1, sputum_cul2, sputum_cul3,
           sputum_xpert1, sputum_xpert2, sputum_xpert3,
           urine_xpert, urine_cul1,
           blood_cul1, blood_cul2, blood_cul3, blood_xpert,
           other_cul1, other_cul2, other_xpert1) %>%
  mutate_all(funs(str_replace(., "Rif_sensitive", "Sensitive"))) %>%
  mutate_all(funs(str_replace(., "Rif_resistant", "Resistant"))) %>%
  filter_all(any_vars(str_detect(., pattern = "Resistant"))) %>%
  filter_all(any_vars(str_detect(., pattern = "Sensitive"))) -> discords_df

discords_df$sensitive_n <- rowSums(discords_df == "Sensitive", na.rm = TRUE)
discords_df$resistant_n <- rowSums(discords_df == "Resistant", na.rm = TRUE)
```

```
discords_df %>%
  mutate(sens_outlier = (sensitive_n==1 & resistant_n>1),
         resi_outlier = (sensitive_n>1 & resistant_n==1)) -> foo

gather(foo, key="test", value="result", 2:13) -> foo2

UID <- c(foo2$UID[foo2$resi_outlier & foo2$result=="Resistant" & !is.na(foo2$result)],
  foo2$UID[foo2$sens_outlier & foo2$result=="Sensitive" & !is.na(foo2$result)])

odd_one_out <- c(foo2$test[foo2$resi_outlier & foo2$result=="Resistant" & !is.na(foo2$result)],
  foo2$test[foo2$sens_outlier & foo2$result=="Sensitive" & !is.na(foo2$result)])

kable(
  left_join(foo, data.frame(UID, odd_one_out), by="UID") %>%
  select(UID, sensitive_n, resistant_n, odd_one_out),
     "latex", booktabs=T)
```

UID	sensitive_n	resistant_n	odd_one_out
16	7	1	sputum_xpert1
202	3	3	NA
203	4	2	NA
285	5	2	NA
439	5	1	urine_xpert
440	1	1	NA
497	1	5	blood_xpert
531	1	4	blood_cul1
545	1	3	blood_cul2
638	4	2	NA
642	2	1	sputum_xpert1
663	4	1	blood_xpert

What is the probability that two TB tests from the same patient will have a discordant rif sensitivity result? As stated above 12 / 368 (3.3%) patients *who have at least 2 test results available* have a discordant result.

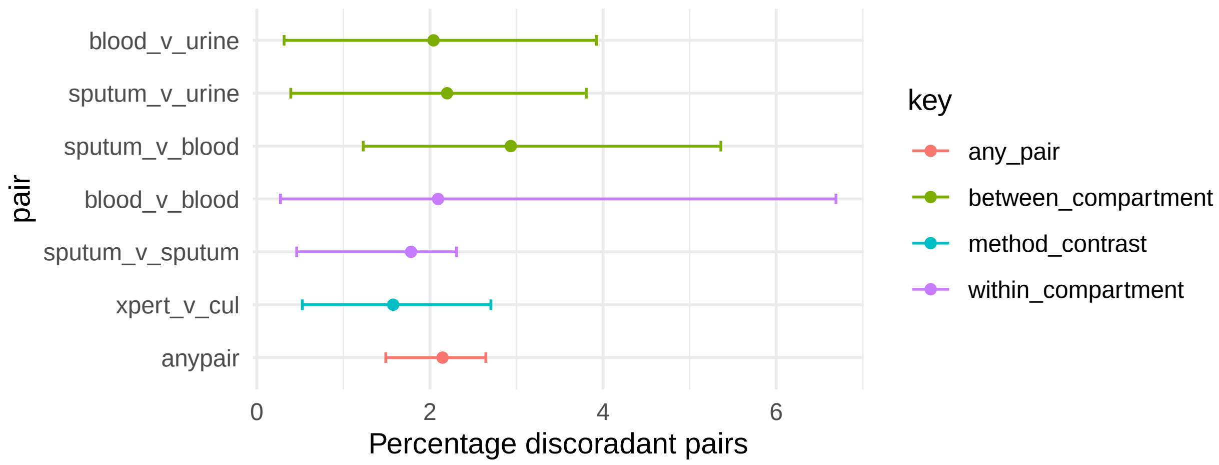
Patients have different numbers of test results, and those with multiple tests presumably have higher opportunity to have a discoradant result. So if we randomly sample a pair of tests from a random patient, and do this repeatedly (say 1000 times, with replacement), what proportion are discordant (ignoring "NA" and "inconclusive" missing test results)? NB: these pairwise comparisons are in most cases based on very sparse data (hence the wide CIs).

```
# data frame with just the harmonised test rif sens results
# replacing any "incolclusive" results with NA
data.frame(sputum_cul1, sputum_cul2, sputum_cul3,
           sputum_xpert1, sputum_xpert2, sputum_xpert3,
           urine_xpert, urine_cul1,
           blood_cul1, blood_cul2, blood_cul3, blood_xpert,
           other_cul1, other_cul2, other_xpert1,
           stringsAsFactors = FALSE) %>%
    mutate_all(funs(str_replace(., "inconclusive",
                                 replacement = NA_character_))) -> foo

# Remove cases with <2 positive tests
foo[rowSums(!is.na(foo))>1, ] -> foo
```

```
fx <- function(data=foo, indicies, n=1000){
  data <- data[indicies,] # boot indicies
  temp <- rep(NA, n)  # temp vector to contain output
  for(i in 1:n){
    x <- data[sample(1:nrow(data), size=1), ]
    pair <- sample(x[!is.na(x)], replace = FALSE, size = 2)
    temp[i] <- pair[1]==pair[2]
  }
  return(sum(!temp)/sum(temp))
}


# a function that does same but allows to selevct contrasts for comparison : a & b
fx2 <- function(data=foo, indicies, n=1000, a="cul", b="xpert"){
  data[indicies, ] -> data                        # the boot indicies
  select(data, matches(paste0(a, "|", b))) %>%    # only want the columsn related to contrasts
    mutate(
      include =                               # only rows with data for each contrast
        rowSums(!is.na(data[, grepl(names(data), pattern=a, fixed = T)]))>0 &
        rowSums(!is.na(data[, grepl(names(data), pattern=b, fixed = T)]))>0) %>%
    filter(include) %>%
    select(-include) -> data

  temp <- rep(NA, n)
  for(i in 1:n){
    x <- data[sample(1:nrow(data), size=1), ]

    pair1 <- sample(x[!is.na(x) & grepl(names(x), pattern=a, fixed = T)],
                    replace = FALSE, size = 1)
    pair2 <- sample(x[!is.na(x) & grepl(names(x), pattern=b, fixed = T)],
                    replace = FALSE, size = 1)

    temp[i] <- pair1==pair2
  }
  return(sum(!temp)/sum(temp))
}


# a function that does same but checks discordance *within* compartment
fx3 <- function(data=foo, indicies, n=1000, a="sputum"){
  data[indicies, ] -> data                        # the boot indicies
  select(data, matches(a)) %>%    # only want the columsn related to compartment
    mutate(
      include =                               # only rows with >1 test of compartment
        rowSums(!is.na(data[, grepl(names(data), pattern=a, fixed = T)]))>1) %>%
    filter(include) %>%
    select(-include) -> data

  temp <- rep(NA, n)
  for(i in 1:n){
    x <- data[sample(1:nrow(data), size=1), ]
    pair <- sample(x[!is.na(x)], replace = FALSE, size = 2)
    temp[i] <- pair[1]==pair[2]  }
  return(sum(!temp)/sum(temp))
}


anypair <- sumBoot(boot(data=foo, R=10,
                        statistic= fx)$t)

xpert_v_cul <- sumBoot(boot(data=foo, R=10,
                            statistic= fx2, a="cul", b="xpert")$t)

sputum_v_blood <- sumBoot(boot(data=foo, R=10,
                            statistic= fx2, a="sputum", b="blood")$t)
sputum_v_urine <- sumBoot(boot(data=foo, R=10,
                            statistic= fx2, a="sputum", b="urine")$t)
blood_v_urine <- sumBoot(boot(data=foo, R=10,
                            statistic= fx2, a="blood", b="urine")$t)

sputum_v_sputum <- sumBoot(boot(data=foo, R=10,
                            statistic= fx3, a="sputum")$t)
blood_v_blood <- sumBoot(boot(data=foo, R=10,
                            statistic= fx3, a="blood")$t)
urine_v_urine <- sumBoot(boot(data=foo, R=10,
                            statistic= fx3, a="urine")$t)
```

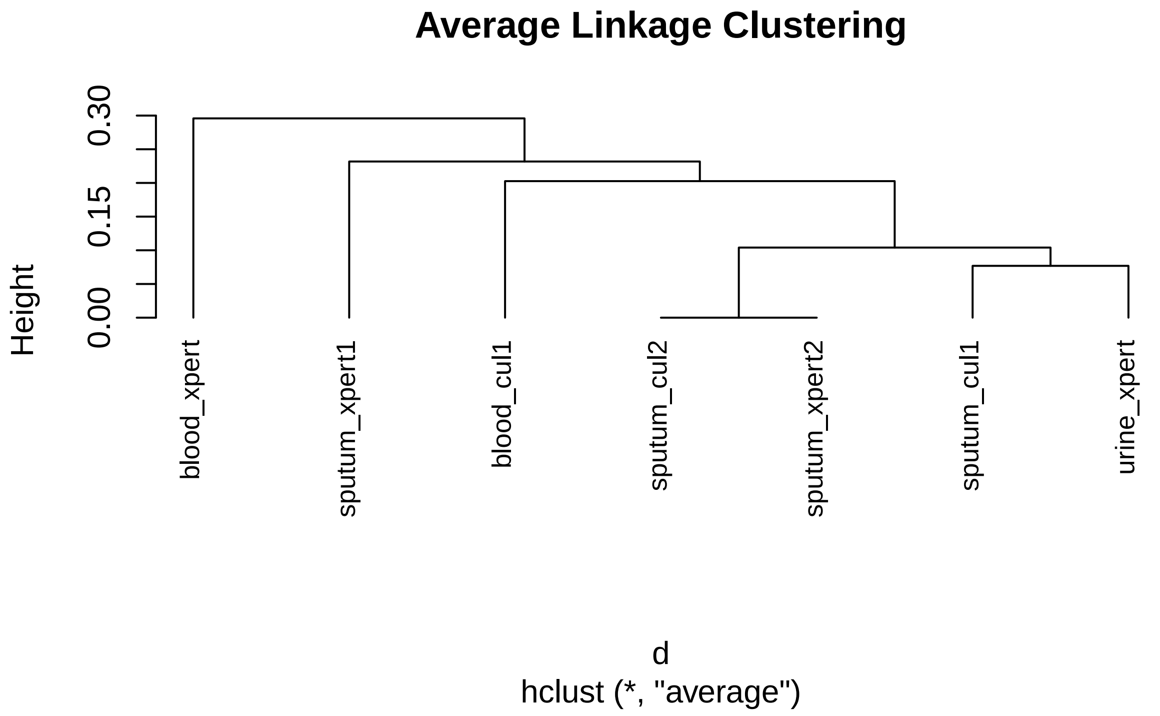
The result is 2.1%. We can also use the same procedure to test more specific sample type pairs. E.g. we can sample any blood and any sputum rif resistance result (within any patient that has at least one result for each), and see if discordance between blood and sputum is on average higher than other pairings (any sputum sample versus any other sputum sample; any Xpert versus any culture etc.). The results are shown below with 95% CI from bootstrapping the procedure 1000 times.

```
bind_rows(anypair, xpert_v_cul,
          sputum_v_blood, sputum_v_urine, blood_v_urine,
          sputum_v_sputum, blood_v_blood) %>%
  mutate(pair = factor(
    c("anypair", "xpert_v_cul",
      "sputum_v_blood", "sputum_v_urine", "blood_v_urine",
      "sputum_v_sputum", "blood_v_blood"),
    levels =
      c("anypair", "xpert_v_cul",
        "sputum_v_sputum", "blood_v_blood",
        "sputum_v_blood", "sputum_v_urine", "blood_v_urine")),
         key = c("any_pair", "method_contrast",
                 rep("between_compartment", 3),
                 rep("within_compartment", 2))) %>%
  ggplot(aes(pair, fit*100, colour=key)) +
  geom_point() +
  geom_errorbar(aes(ymin=lwr*100, ymax=upr*100), width=0.2) +
  theme_minimal() +
  ylab("Percentage discoradant pairs") +
  coord_flip()
```



Clustering the same data shows blood Xpert results furtherest from the rest:

```r
data.frame(sputum_cul1, sputum_cul2,
           sputum_xpert1, sputum_xpert2,
           urine_xpert,
           blood_cul1, blood_xpert,
           stringsAsFactors = FALSE) %>%
  mutate_all(funs(str_replace(., "Sensitive", "0"))) %>%
  mutate_all(funs(str_replace(., "Resistant", "1"))) %>%
  mutate_all(funs(as.numeric(.))) -> m

m <- t(m)
d <- dist(m, method = "binary")
fit.average <- hclust(d, method="average")
plot(fit.average, hang=-1, cex=.8, main="Average Linkage Clustering")
```

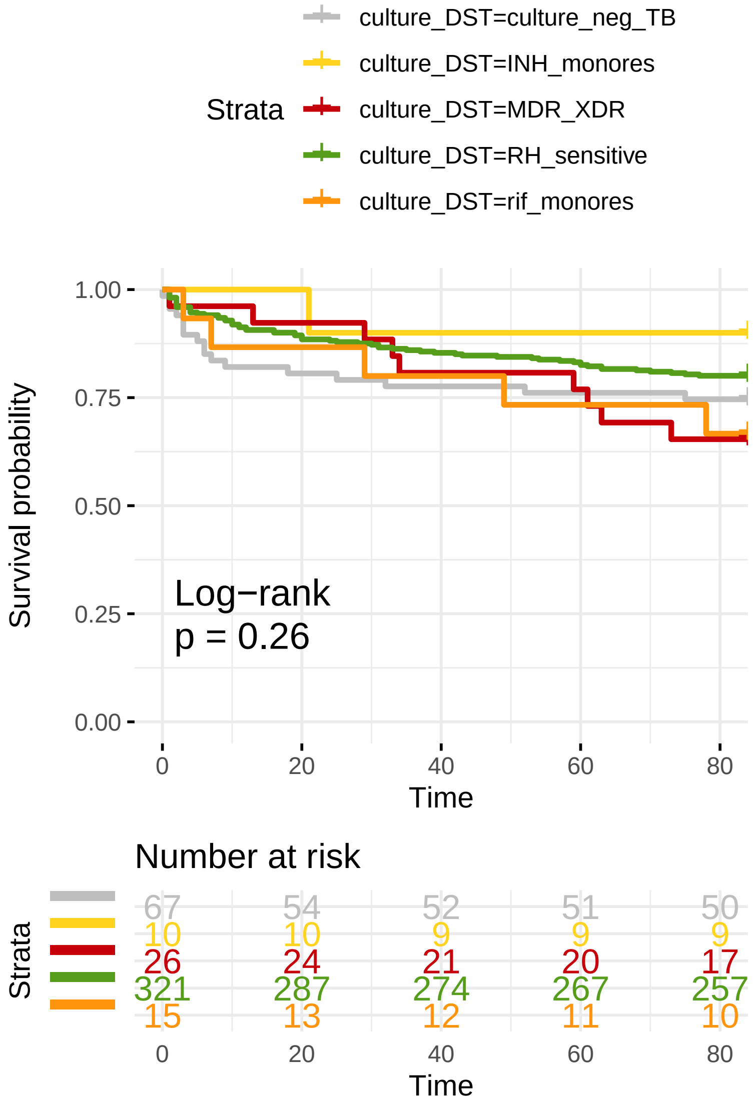


6.1.3 Mortality and DR TB

6.1.3.1 By culture DST results

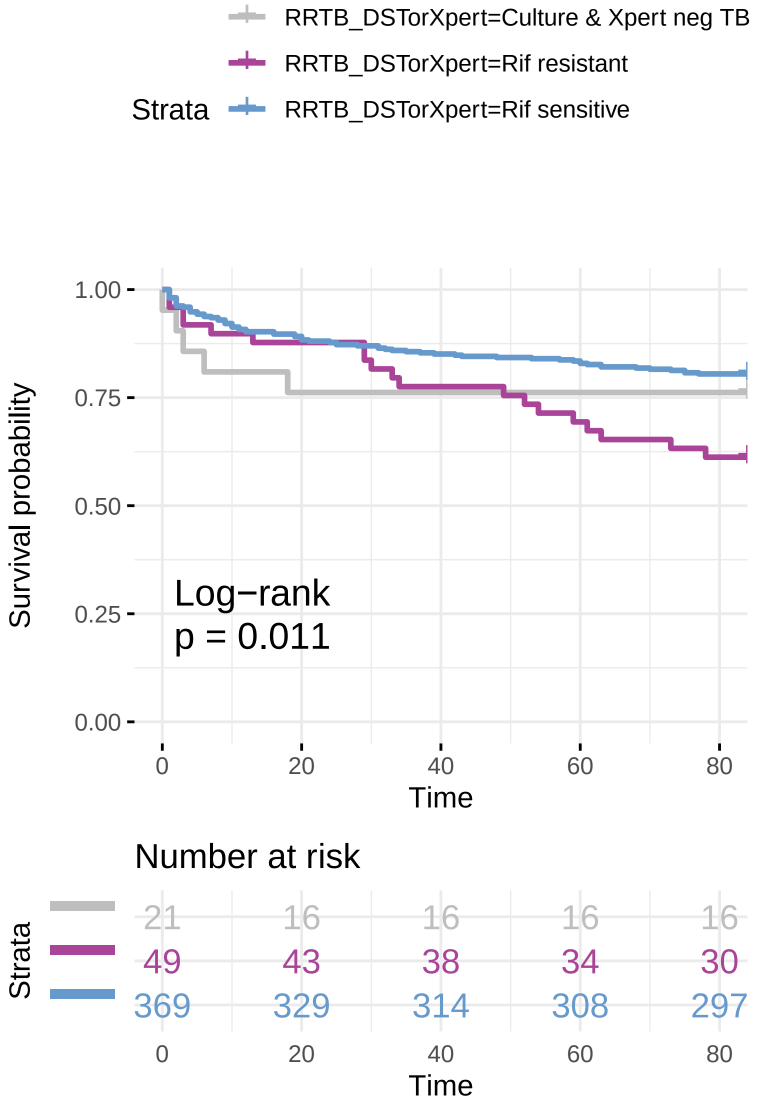
Here any resistant result classifies a case as drug resistant.

```
# these are applied in sequence so as to be in a hierarchy eg any MDR result means case classified as M
tbdf$culture_DST <- NA
tbdf$culture_DST[tbdf$RH_sensitive] <- "RH_sensitive"
tbdf$culture_DST[tbdf$INH_monoDR] <- "INH_monores"
tbdf$culture_DST[tbdf$rif_monoDR] <- "rif_monores"
tbdf$culture_DST[tbdf$MDR_XDR_TB] <- "MDR_XDR"
tbdf$culture_DST[is.na(tbdf$culture_DST)] <- "culture_neg_TB"


y <- Surv(tbdf$time, tbdf$day84death)
km <- survfit(y ~ tbdf$culture_DST)

ggsurvplot(km, data = tbdf,
           risk.table = TRUE,
           palette = c("grey", "#ffd320", "#c5000b",
                       "#579d1c", "#ff950e"),
           pval = TRUE, pval.method = TRUE,
           ggtheme = theme_minimal(),
           risk.table.col="strata",
           risk.table.y.text=FALSE) +
  guides(colour = guide_legend(nrow = 5))
```



6.1.3.2 By any rif resistant result including Xpert

```
tbdf$RRTB_DSTorXpert[
  is.na(tbdf$sputumGXP1_RifDST) &
  is.na(tbdf$sputumGXP2_RifDST) &
  is.na(tbdf$sputumGXP3_RifDST) &

    is.na(tbdf$sputumCulture1_MTBDST) &
    is.na(tbdf$sputumCulture2_MTBDST) &
    is.na(tbdf$sputumCulture3_MTBDST) &

    is.na(tbdf$MBC1_MTBDST) &
    is.na(tbdf$MBC2_MTBDST) &
    is.na(tbdf$MBC3_MTBDST) &

    is.na(tbdf$uMTBculture.MTBDST) &
    is.na(tbdf$otherCul1_MTBDST) &
    is.na(tbdf$otherCul2_MTBDST) &

    is.na(tbdf$uGXP.Rifprobe) &
    is.na(tbdf$otherGXP.refprobe) &
    is.na(tbdf$blood_Xpert_rif)] <- "Culture & Xpert neg TB"

tbdf$RRTB_DSTorXpert[tbdf$RRTB_DSTorXpert=="TRUE"] <- "Rif resistant"
tbdf$RRTB_DSTorXpert[tbdf$RRTB_DSTorXpert=="FALSE"] <- "Rif sensitive"

y <- Surv(tbdf$time, tbdf$day84death)
km <- survfit(y ~ tbdf$RRTB_DSTorXpert)

ggsurvplot(km, data = tbdf,
           risk.table = TRUE,
           palette = c("grey", "#AA4499", "#6699CC"),
           pval = TRUE, pval.method = TRUE,
           ggtheme = theme_minimal(),
           risk.table.col="strata",
           risk.table.y.text=FALSE) +
  guides(colour = guide_legend(nrow = 5))
```



6.2 Time to detection of rif resistance

Have defined *rif resistance* here as culture confirmed rif resistance (mono or M/XDR) on any culture sample (n=43), rather than any culture or Xpert result = rif resistant (n=51). (As shown above the difference of 8 patients is a mix of culture negative and culture DST - Xpert rif probe discoradant cases.)

Have defined *'time to detection of rif resitance by culture'* as the lowest TTP of any culture result, +1 day.

Will compare blood Xpert to a single sputum Xpert, using the same method to select a single sputum Xpert as was used above in section 4 sensitivity and diagnostic yield analysis.

```
# a data frame of the culture confirmed rif resistance cases
tbdf %>%
  filter(!is.na(culture_DST) &
           (culture_DST=="MDR_XDR" | culture_DST=="rif_monores")) -> foo

# add the minimum time to detection to this data frame
```

```r
foo %>%
  select(contains("TTP")) %>%
  rowwise() %>%
  mutate(min_ttp =
           min(
             sputumCulture1_TTP, sputumCulture2_TTP, sputumCulture3_TTP,
             MBC1_TTP, MBC2_TTP, MBC3_TTP,
             uMTBculture.TTP, otherCul1_TTP, otherCul2_TTP,
             na.rm = TRUE)) -> min_ttp
```

```
## Warning in min(sputumCulture1_TTP, sputumCulture2_TTP,
## sputumCulture3_TTP, : no non-missing arguments to min; returning Inf

## Warning in min(sputumCulture1_TTP, sputumCulture2_TTP,
## sputumCulture3_TTP, : no non-missing arguments to min; returning Inf
```

```r
foo$min_ttd <- min_ttp$min_ttp + 1
foo$min_ttd[!is.finite(foo$min_ttd)] <- median(foo$min_ttd)

#### SELECTING  A SPUTUM XPERT RIF RESULTS

# create a new variable which will be our final sputum Xpert result
foo$sputum_xpert_rif <- rep("foo", nrow(foo))

# This for loop now populates that new sputum variable so that it is:
## NA if all 3 sputum Xperts are NA
## gets result of single Xpert result if only one available
## picks closest to recruitment date or 'samples' one at random if 2 or 3 are available on same day

for(i in 1:nrow(foo)){

  if(is.na(foo$sputumGXP1[i]) &
     is.na(foo$sputumGXP2[i]) &
     is.na(foo$sputumGXP3[i])){
    foo$sputum_xpert_rif[i] <- NA  # If all 3 NA then result is NA
     } else

  if(!is.na(foo$sputumGXP1[i]) &
     is.na(foo$sputumGXP2[i]) &
     is.na(foo$sputumGXP3[i])){
    foo$sputum_xpert_rif[i] <- foo$sputumGXP1_RifDST[i]
     } else

  if(is.na(foo$sputumGXP1[i]) &
     !is.na(foo$sputumGXP2[i]) &
     is.na(foo$sputumGXP3[i])){
    foo$sputum_xpert_rif[i] <- foo$sputumGXP2_RifDST[i]
     } else

  if(is.na(foo$sputumGXP1[i]) &
     is.na(foo$sputumGXP2[i]) &
     !is.na(foo$sputumGXP3[i])){
    foo$sputum_xpert_rif[i] <- foo$sputumGXP3_RifDST[i]
     } else
```

```
                  # If only 1/3 recorded then result is that one
if(!is.na(foo$sputumGXP1[i]) &
   !is.na(foo$sputumGXP2[i]) &
   is.na(foo$sputumGXP3[i])){
  if(
    (abs(foo$sptmxpert1_day[i])<abs(foo$sptmxpert2_day[i]) &
       !is.na(abs(foo$sptmxpert1_day[i])<abs(foo$sptmxpert2_day[i])))
  ){
    foo$sputum_xpert_rif[i] <- foo$sputumGXP1_RifDST[i]
    }else
      if(
    (abs(foo$sptmxpert1_day[i])>abs(foo$sptmxpert2_day[i]) &
       !is.na(abs(foo$sptmxpert1_day[i])>abs(foo$sptmxpert2_day[i])))
  ){
    foo$sputum_xpert_rif[i] <- foo$sputumGXP2_RifDST[i]
    } else
      if(
        (abs(foo$sptmxpert1_day[i])==abs(foo$sptmxpert2_day[i]) &
       !is.na(abs(foo$sptmxpert1_day[i])==abs(foo$sptmxpert2_day[i])))
      ){
        foo$sputum_xpert_rif[i] <-
    sample(c(foo$sputumGXP1_RifDST[i],
             foo$sputumGXP2_RifDST[i]), 1)
      }
           } else
                  # if 2 result available sample 1 closest to recruitment and if both same day select

  if(is.na(foo$sputumGXP1[i]) &
   !is.na(foo$sputumGXP2[i]) &
   !is.na(foo$sputumGXP3[i])){
  if(
    (abs(foo$sptmxpert2_day[i])<abs(foo$sptmxpert3_day[i]) &
       !is.na(abs(foo$sptmxpert2_day[i])<abs(foo$sptmxpert3_day[i])))
  ){
    foo$sputum_xpert_rif[i] <- foo$sputumGXP2_RifDST[i]
    } else
      if(
    (abs(foo$sptmxpert2_day[i])>abs(foo$sptmxpert3_day[i]) &
       !is.na(abs(foo$sptmxpert2_day[i])>abs(foo$sptmxpert3_day[i])))
  ){
    foo$sputum_xpert_rif[i] <- foo$sputumGXP3_RifDST[i]
    } else
      if(
        (abs(foo$sptmxpert2_day[i])==abs(foo$sptmxpert3_day[i]) &
       !is.na(abs(foo$sptmxpert2_day[i])==abs(foo$sptmxpert3_day[i])))
      ){
        foo$sputum_xpert_rif[i] <-
    sample(c(foo$sputumGXP2_RifDST[i],
             foo$sputumGXP3_RifDST[i]), 1)
      }
           } else

  if(!is.na(foo$sputumGXP1[i]) &
     is.na(foo$sputumGXP2[i]) &
     !is.na(foo$sputumGXP3[i])){
    if(
      (abs(foo$sptmxpert1_day[i])<abs(foo$sptmxpert3_day[i]) &
         !is.na(abs(foo$sptmxpert1_day[i])<abs(foo$sptmxpert3_day[i])))
    ){
      foo$sputum_xpert_rif[i] <- foo$sputumGXP1_RifDST[i]
      } else
        if(
      (abs(foo$sptmxpert1_day[i])>abs(foo$sptmxpert3_day[i]) &
         !is.na(abs(foo$sptmxpert1_day[i])>abs(foo$sptmxpert3_day[i])))
    ){
      foo$sputum_xpert_rif[i] <- foo$sputumGXP3_RifDST[i]
      } else
        if(
          (abs(foo$sptmxpert1_day[i])==abs(foo$sptmxpert3_day[i]) &
         !is.na(abs(foo$sptmxpert1_day[i])==abs(foo$sptmxpert3_day[i])))
        ){
          foo$sputum_xpert_rif[i] <-
      sample(c(foo$sputumGXP1_RifDST[i],
               foo$sputumGXP3_RifDST[i]), 1)
        }
             } else
       # now for the times when all 3 results are available...
  if(!is.na(foo$sputumGXP1[i]) &
     !is.na(foo$sputumGXP2[i]) &
     !is.na(foo$sputumGXP3[i])){

# one sample of 3 is closest to recruitment:
      if(
        (abs(foo$sptmxpert1_day[i])<abs(foo$sptmxpert2_day[i])) &
        (abs(foo$sptmxpert1_day[i])<abs(foo$sptmxpert3_day[i]))){
          foo$sputum_xpert_rif[i] <- foo$sputumGXP1_RifDST[i]}else
      if(
        (abs(foo$sptmxpert2_day[i])<abs(foo$sptmxpert1_day[i])) &
        (abs(foo$sptmxpert2_day[i])<abs(foo$sptmxpert3_day[i]))){
          foo$sputum_xpert_rif[i] <- foo$sputumGXP2_RifDST[i]}else
      if(
        (abs(foo$sptmxpert3_day[i])<abs(foo$sptmxpert2_day[i])) &
        (abs(foo$sptmxpert3_day[i])<abs(foo$sptmxpert1_day[i]))){
          foo$sputum_xpert_rif[i] <- foo$sputumGXP3_RifDST[i]}else

# now cases where 2 of 3 available are same day

      if(
        (abs(foo$sptmxpert1_day[i])<abs(foo$sptmxpert2_day[i])) &
        (abs(foo$sptmxpert1_day[i])==abs(foo$sptmxpert3_day[i]))){
          foo$sputum_xpert_rif[i] <- sample(
            c(foo$sputumGXP1_RifDST[i], foo$sputumGXP3_RifDST[i]), 1)}else
      if(
        (abs(foo$sptmxpert1_day[i])<abs(foo$sptmxpert3_day[i])) &
        (abs(foo$sptmxpert1_day[i])==abs(foo$sptmxpert2_day[i]))){
          foo$sputum_xpert_rif[i] <- sample(
            c(foo$sputumGXP1_RifDST[i], foo$sputumGXP2_RifDST[i]), 1)}else
      if(
        (abs(foo$sptmxpert2_day[i])<abs(foo$sptmxpert1_day[i])) &
        (abs(foo$sptmxpert2_day[i])==abs(foo$sptmxpert3_day[i]))){
          foo$sputum_xpert_rif[i] <- sample(
            c(foo$sputumGXP2_RifDST[i], foo$sputumGXP3_RifDST[i]), 1)}else

# all 3 are same day, sample one at random
      if(
        (abs(foo$sptmxpert1_day[i])==abs(foo$sptmxpert2_day[i])) &
        (abs(foo$sptmxpert2_day[i])==abs(foo$sptmxpert3_day[i]))
        ){
        foo$sputum_xpert_rif[i] <- sample(
          c(foo$sputumGXP1_RifDST[i], foo$sputumGXP2_RifDST[i], foo$sputumGXP3_RifDST[i]), 1)}
  }
}
```

Rif resistance results for asingle sputum Xpert (rows) versus blood Xpert, amongst patients with rif resistnace diagnosed by culture:

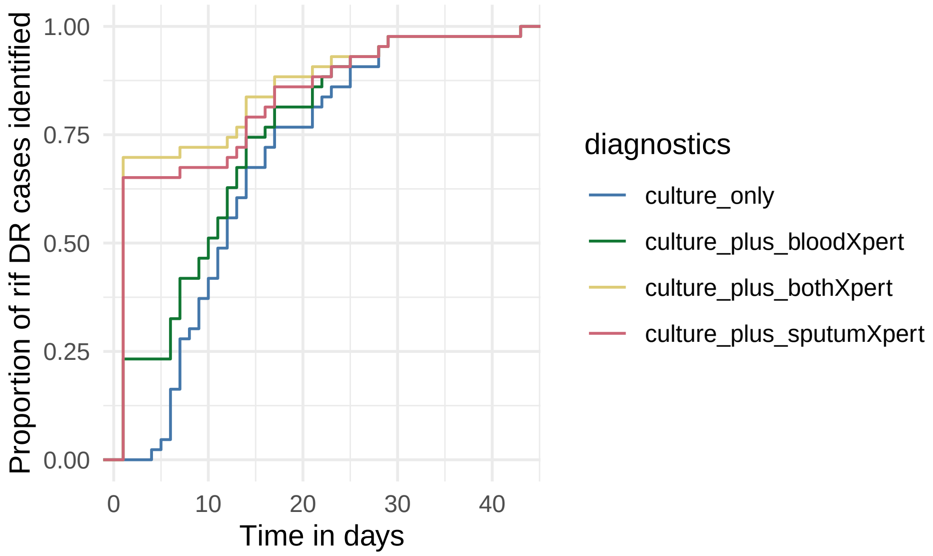
```
kable(table(foo$sputum_xpert_rif, foo$blood_Xpert_rif, useNA = "always"),
      "latex", booktabs=T)
```

	Indeterminate	Not detected	Resistance detected	NA
Resistant	3	2	8	15
Sensitive	0	1	0	1
NA	2	0	2	9

Meaning a single Xpert would have picked up 28/43 cases; a single blood xpert picked up 10/43 cases, including 2 not detected by the sputum Xpert.

Time to detection of rifampicin drug resistance by culture alone, culture with a sputum Xpert, culture with a blood xpert, or culture with both xperts (assuming Xpert result takes 1 day):

```
ttd_cul <- foo$min_ttd
ttd_with_sputum_xpert <- ttd_cul
ttd_with_sputum_xpert[!is.na(foo$sputum_xpert_rif) &
                        foo$sputum_xpert_rif=="Resistant"] <- 1
ttd_with_blood_xpert <- ttd_cul
ttd_with_blood_xpert[!is.na(foo$blood_Xpert_rif) &
                       foo$blood_Xpert_rif=="Resistance detected"] <- 1
ttd_with_both_xpert <- ttd_cul
ttd_with_both_xpert[(!is.na(foo$sputum_xpert_rif) &
                        foo$sputum_xpert_rif=="Resistant") |
                      (!is.na(foo$blood_Xpert_rif) &
                       foo$blood_Xpert_rif=="Resistance detected")] <- 1

data.frame(culture_only = ttd_cul,
           culture_plus_sputumXpert = ttd_with_sputum_xpert,
           culture_plus_bloodXpert = ttd_with_blood_xpert,
           culture_plus_bothXpert = ttd_with_both_xpert) %>%
  gather(key = "diagnostics", value = "ttd_rdr", 1:4) %>%
  ggplot(aes(x=ttd_rdr, colour=diagnostics)) +
  geom_step(aes(y=..y..), stat="ecdf") +
  xlab("Time in days") +
  ylab("Proportion of rif DR cases identified") +
  scale_colour_ptol() +
  theme_minimal()
```



Added value of blood Xpert for diagnosing rif resistance looks pretty marginal here. This might be partly because a lot of the drug resistant TB is picked up by sputum culture?

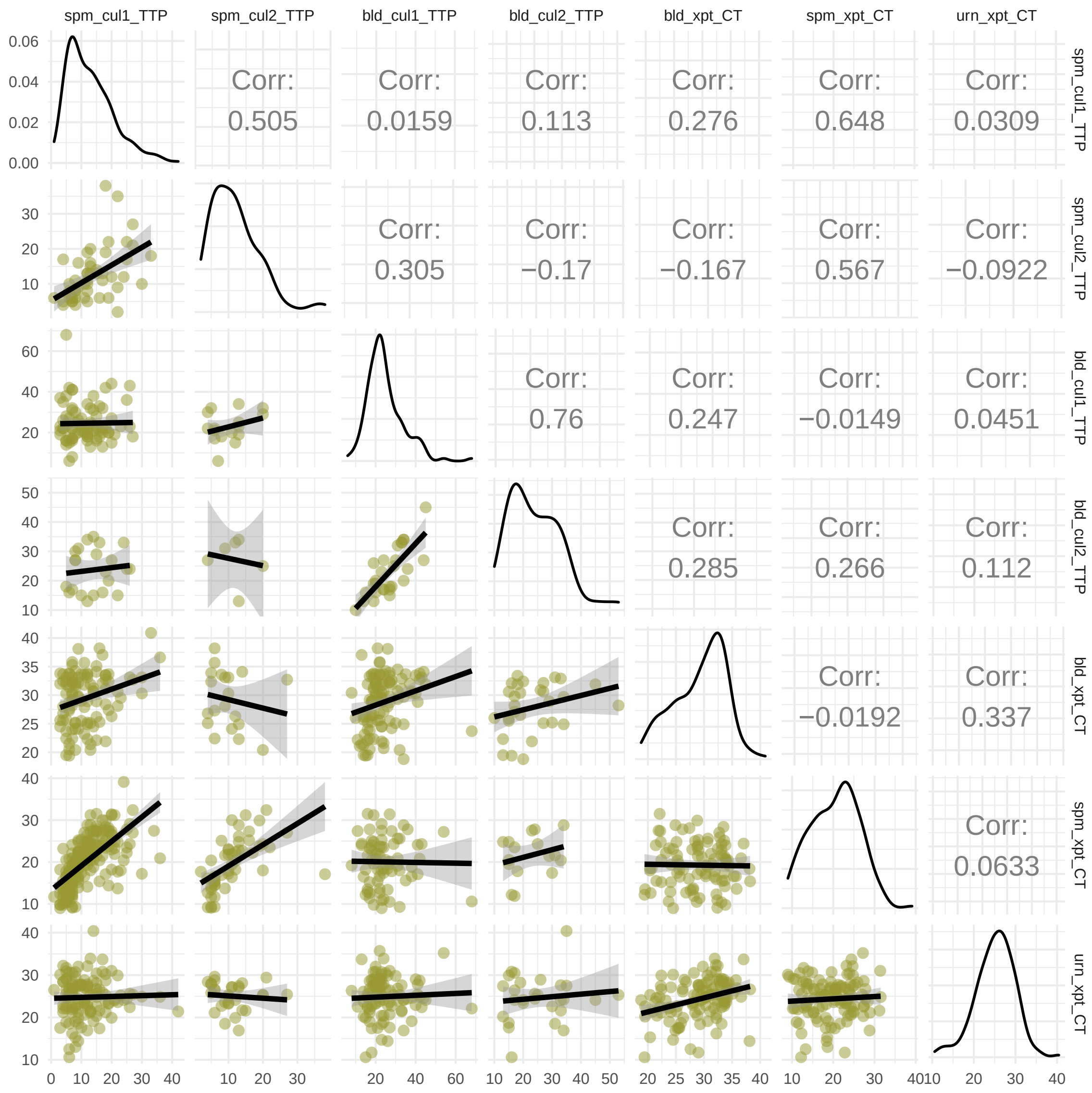
7 Correlation between (semi) quantitative measures of bacilli number

7.1 Pairwise comparisons, all samples we have readouts for

7.1.1 pairwise scatter plots

With Pearson's correlation coefficients

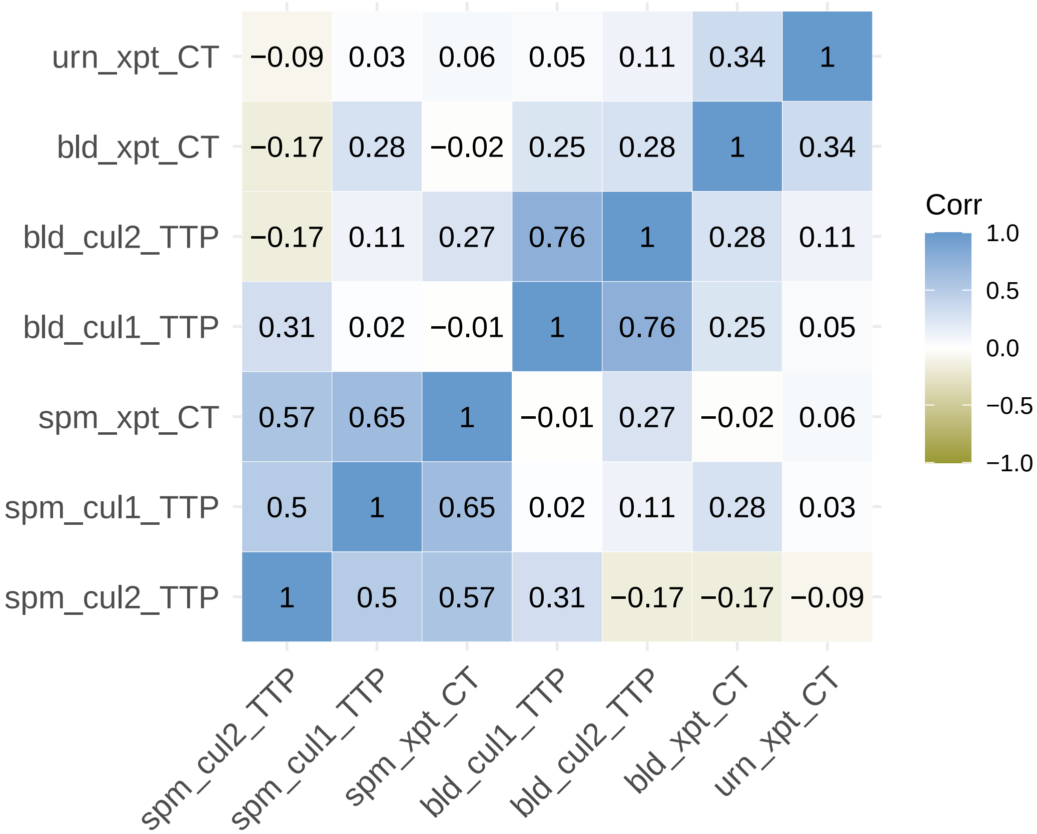
```
tbdf %>%
  select(spm_cul1_TTP=sputumCulture1_TTP,
         spm_cul2_TTP=sputumCulture2_TTP,
         bld_cul1_TTP=MBC1_TTP,
         bld_cul2_TTP=MBC2_TTP,
         bld_xpt_CT=blood_Xpert_CT,
         spm_xpt_CT=min.ct_sptmGXP,
         urn_xpt_CT=min.ct_urineGXP) %>%
  ggpairs(lower=list(continuous = wrap("smooth", alpha=0.5, colour="#999933"))) + theme_minimal(base_si
```



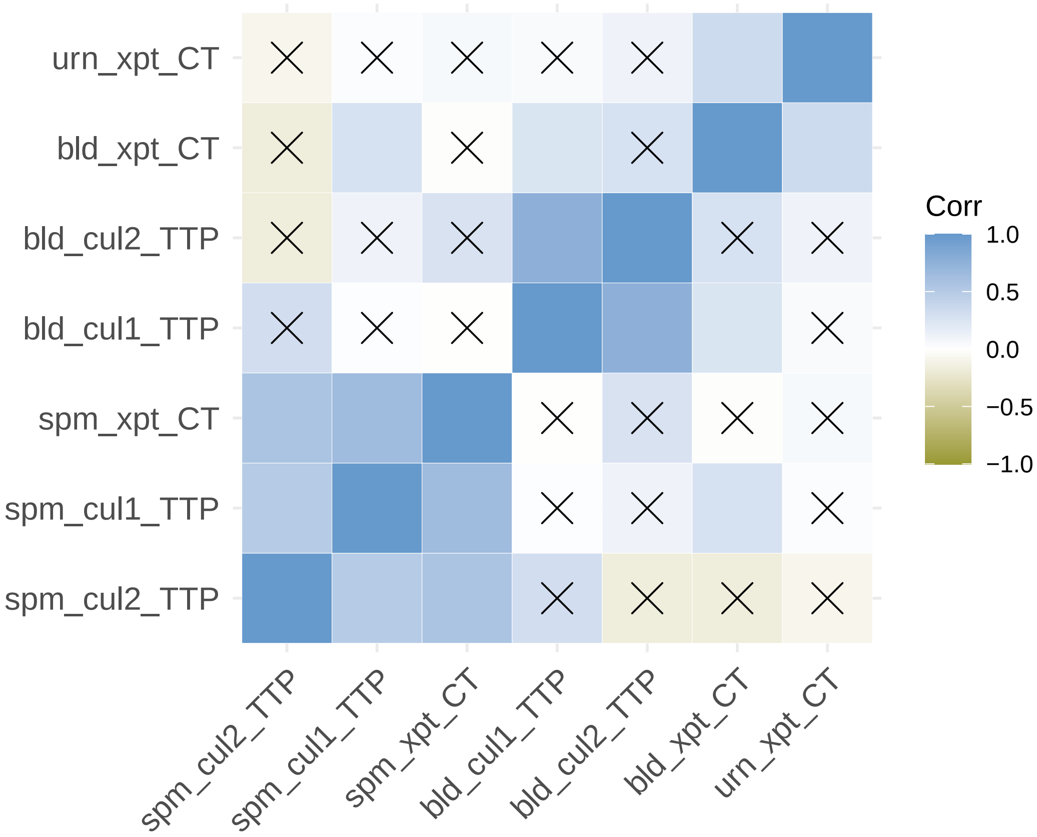
7.1.2 Same correlations but in a correlation matrix

- A : with Pearson's r coefficients shown by numbers in the squares
- B : same plot, but with "non-significant" (p>0.05) correlations indicated by an X in the squares

```
m <- cor(tbdf %>%
  select(spm_cul1_TTP=sputumCulture1_TTP,
         spm_cul2_TTP=sputumCulture2_TTP,
         bld_cul1_TTP=MBC1_TTP,
         bld_cul2_TTP=MBC2_TTP,
         bld_xpt_CT=blood_Xpert_CT,
         spm_xpt_CT=min.ct_sptmGXP,
         urn_xpt_CT=min.ct_urineGXP),
  use = "pairwise")
p.mat <- cor_pmat(tbdf %>%
  select(spm_cul1_TTP=sputumCulture1_TTP,
         spm_cul2_TTP=sputumCulture2_TTP,
         bld_cul1_TTP=MBC1_TTP,
         bld_cul2_TTP=MBC2_TTP,
         bld_xpt_CT=blood_Xpert_CT,
         spm_xpt_CT=min.ct_sptmGXP,
         urn_xpt_CT=min.ct_urineGXP),
  use = "pairwise")

ggcorrplot(m,
  hc.order = TRUE, outline.color = "white", lab = TRUE,
  colors = c("#999933", "white", "#6699CC"))
```

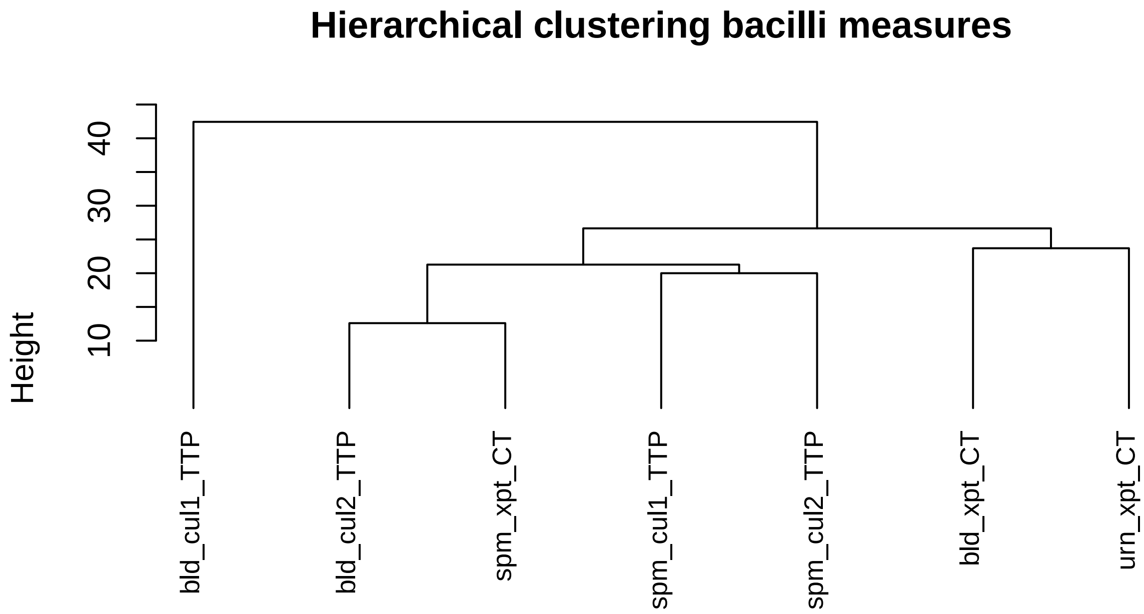


```
ggcorrplot(m,
  hc.order = TRUE, outline.color = "white", p.mat=p.mat,
  colors = c("#999933", "white", "#6699CC"))
```



7.1.3 Clustering the same variables

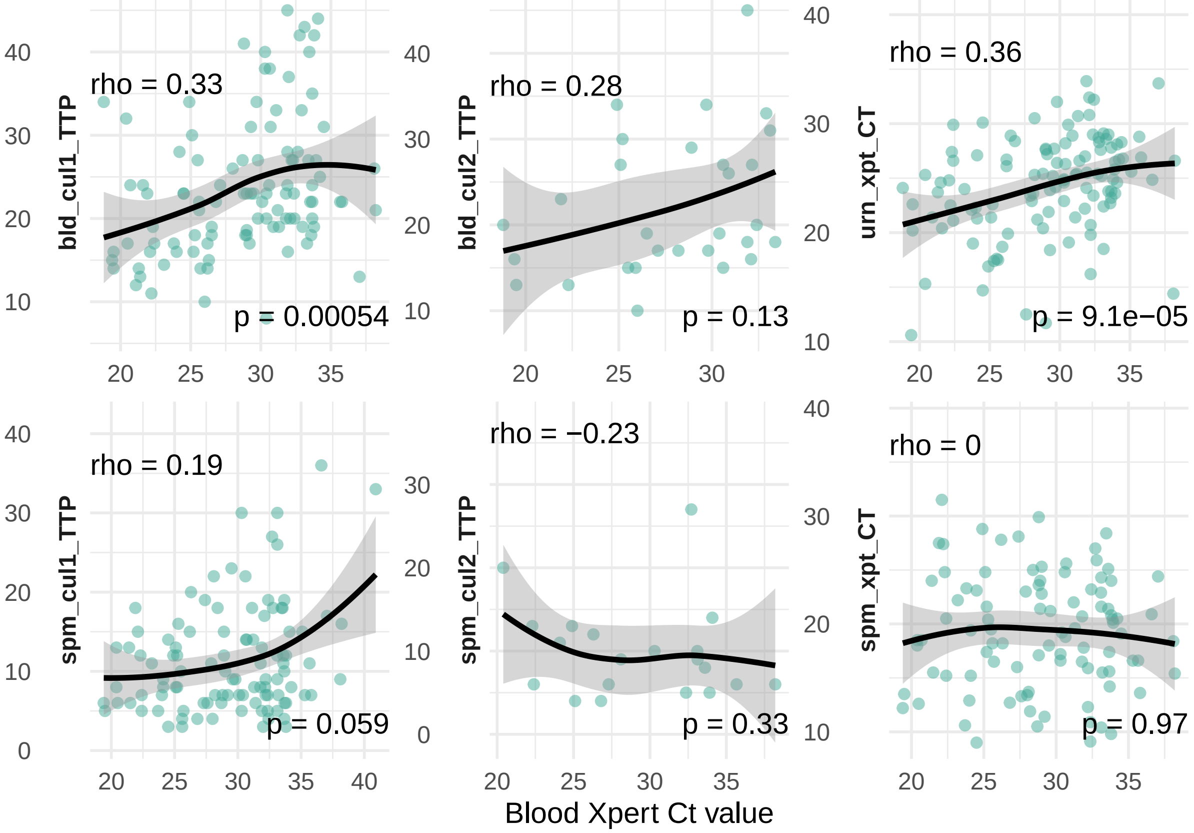
```
m <- tbdf %>%
  select(spm_cul1_TTP=sputumCulture1_TTP,
         spm_cul2_TTP=sputumCulture2_TTP,
         bld_cul1_TTP=MBC1_TTP,
         bld_cul2_TTP=MBC2_TTP,
         bld_xpt_CT=blood_Xpert_CT,
         spm_xpt_CT=min.ct_sptmGXP,
         urn_xpt_CT=min.ct_urineGXP)

d <- dist(t(as.matrix(m)), method = "maximum")
fit.average <- hclust(d, method="average")
plot(fit.average, hang=-1, cex=.8, main="Hierarchical clustering bacilli measures", xlab="", sub="")
```



7.2 Plot focusing on blood Xpert Ct values

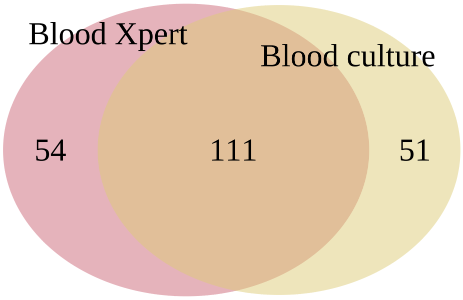
Have excluded outlier values - TTPs greater than 48 days.
Spearman's Rho with p value shown.

```r
df.cor <- function(x, y) {
  round(cor(x[y<50], y[y<50], use = "complete.obs", method = "spear"), 2)
}

df.p <- function(x, y){
  formatC(cor.test(x[y<50], y[y<50],
                   method = "spear",
                   use="complete.cases")$p.value,
          format="e", digits=1)
}


tbdf %>%
  select(
    bld_xpt_CT=blood_Xpert_CT,
    spm_cul1_TTP=sputumCulture1_TTP,
    spm_cul2_TTP=sputumCulture2_TTP,
    bld_cul1_TTP=MBC1_TTP,
    bld_cul2_TTP=MBC2_TTP,
    spm_xpt_CT=min.ct_sptmGXP,
    urn_xpt_CT=min.ct_urineGXP) -> foo

foo %>% map(df.cor, y=foo$bld_xpt_CT) -> rdf
names(rdf) -> var
data.frame(var,
           rho = as.numeric(rdf)) %>%
  filter(var!="bld_xpt_CT") -> rdf

foo %>% map(df.p, y=foo$bld_xpt_CT) -> pdf
names(pdf) -> var
data.frame(var,
           p = as.numeric(pdf)) %>%
  filter(var!="bld_xpt_CT") -> pdf

foo %>%
  gather(key=var, value = value, 2:7) %>%
  filter(value<50) %>%
  mutate(var = factor(var, levels =
                        c("bld_cul1_TTP", "bld_cul2_TTP",
                          "urn_xpt_CT", "spm_cul1_TTP",
                          "spm_cul2_TTP", "spm_xpt_CT"))) %>%
  ggplot(aes(bld_xpt_CT, value)) +
  geom_point(colour="#44AA99", alpha=0.5) +
  geom_smooth(span=2, colour="black") +
  geom_text(data=rdf,
            aes(label = paste0("rho = ", rho)),
            x=-Inf, y=38, hjust=0, vjust=1.2) +
  geom_text(data=pdf,
            aes(label = paste0("p = ", p)),
            x=Inf, y=-Inf, hjust=1, vjust=-1.2) +
  facet_wrap(~var, scales = "free", strip.position = "left") +
  theme_minimal() +
  theme(strip.text = element_text(face = "bold")) +
  ylab("") + xlab("Blood Xpert Ct value")
```



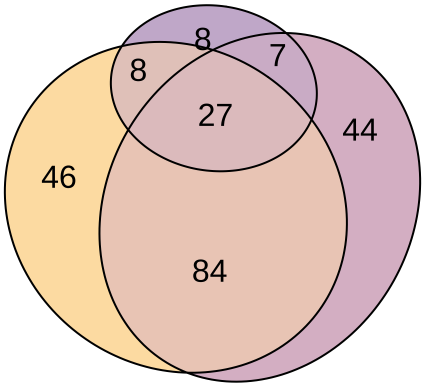
8 Venn/Euler and Venn type figures

```
area1 = sum(tbdf$bld_xpert_diagnosed)
area2 = sum(tbdf$MBC1_cultureID=="MTB" &
              !is.na(tbdf$MBC1_cultureID))
n12 = sum(tbdf$bld_xpert_diagnosed &
            tbdf$MBC1_cultureID=="MTB" &
            !is.na(tbdf$MBC1_cultureID))

draw.pairwise.venn(area1 = area1, area2 = area2, cross.area = n12,
                   category = c("Blood Xpert", "Blood culture"),
                   lty="blank",
                   fill = c("#CC6677", "#DDCC77"),
                   cat.just = list(c(0,0), c(1,1)))
```



```
## (polygon[GRID.polygon.4475], polygon[GRID.polygon.4476], polygon[GRID.polygon.4477], polygon[GRID.po
```

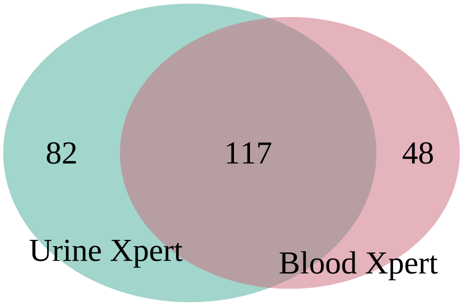
```
Blood_Xpt <- tbdf$bld_xpert_diagnosed
Blood_Cul_1 <- tbdf$MBC1_cultureID=="MTB" & !is.na(tbdf$MBC1_cultureID)
Blood_Cul_2 <- tbdf$MBC2_cultureID=="MTB" & !is.na(tbdf$MBC2_cultureID)
Blood_Cul_1_f <- tbdf$MBC1_cultureID=="MTB"
Blood_Cul_2_f <- tbdf$MBC2_cultureID=="MTB"

euldf <- data.frame(Blood_Xpt, Blood_Cul_1, Blood_Cul_2, Blood_Cul_1_f, Blood_Cul_2_f)


plot( euler(euldf[,c(1:3)], shape="ellipse"),
      quantities = TRUE,
      fills = list(fill = c("#f9b641ff", "#a65c85ff", "#7e4e90ff"), alpha=0.5),
      labels = list(alpha=0))
```



```
c1 <- cohen.kappa(table(Blood_Xpt, Blood_Cul_1_f))
c2 <- cohen.kappa(table(Blood_Xpt, Blood_Cul_2_f))
c3 <- cohen.kappa(table(Blood_Cul_1_f, Blood_Cul_2_f))

ck <- data.frame(
  contrast = c("Xpt v Cul1", "Xpt v Cul2", "Cul1 v Cul2"),
  n = c(c1$n.obs, c2$n.obs, c3$n.obs),
  kappa = round(c(c1$kappa, c2$kappa, c3$kappa), 2),
  CI = c(
    paste0(round(c1$confid[1,1],2), "-", round(c1$confid[1,3],2)),
    paste0(round(c2$confid[1,1],2), "-", round(c2$confid[1,3],2)),
    paste0(round(c3$confid[1,1],2), "-", round(c3$confid[1,3],2))
  )
)

kable(ck,"latex", booktabs=T)
```

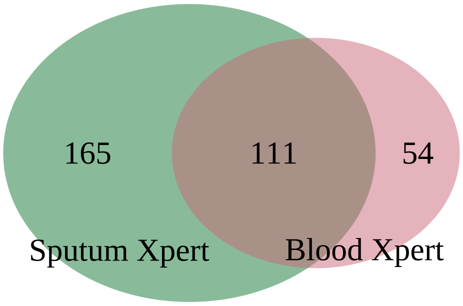
contrast	n	kappa	CI
Xpt v Cul1	438	0.49	0.41-0.58
Xpt v Cul2	113	0.38	0.21-0.55
Cul1 v Cul2	113	0.46	0.29-0.62

```
area1 = sum(tbdf$bld_xpert_diagnosed)
area2 = sum(tbdf$uGXP=="MTB" &
              !is.na(tbdf$uGXP))
n12 = sum(tbdf$bld_xpert_diagnosed &
            tbdf$uGXP=="MTB" &
            !is.na(tbdf$uGXP))

draw.pairwise.venn(area1 = area1, area2 = area2, cross.area = n12,
                   category = c("Blood Xpert", "Urine Xpert"),
                   lty="blank",
                   fill = c("#CC6677", "#44AA99"),
                   cat.just = list(c(1,1), c(0,0)))
```



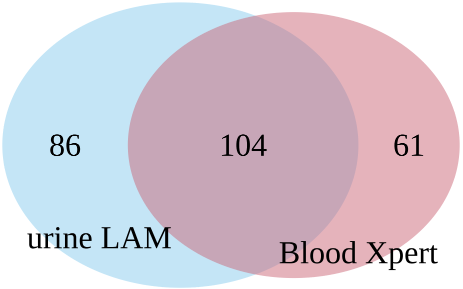
```
## (polygon[GRID.polygon.4484], polygon[GRID.polygon.4485], polygon[GRID.polygon.4486], polygon[GRID.po
```

```
area1 = sum(tbdf$bld_xpert_diagnosed)
area2 = sum(tbdf$sputumGXP1_GeneXpert=="MTB" &
              !is.na(tbdf$sputumGXP1_GeneXpert))
n12 = sum(tbdf$bld_xpert_diagnosed &
            tbdf$sputumGXP1_GeneXpert=="MTB" &
            !is.na(tbdf$sputumGXP1_GeneXpert))

draw.pairwise.venn(area1 = area1, area2 = area2, cross.area = n12,
                   category = c("Blood Xpert", "Sputum Xpert"),
                   lty="blank",
                   fill = c("#CC6677", "#117733"),
                   cat.just = list(c(1,1), c(0,0)))
```



```
## (polygon[GRID.polygon.4493], polygon[GRID.polygon.4494], polygon[GRID.polygon.4495], polygon[GRID.po
```

```
area1 = sum(tbdf$bld_xpert_diagnosed)
area2 = sum(tbdf$ALERE_FC=="MTB" &
              !is.na(tbdf$ALERE_FC))
n12 = sum(tbdf$bld_xpert_diagnosed &
            tbdf$ALERE_FC=="MTB" &
            !is.na(tbdf$ALERE_FC))

draw.pairwise.venn(area1 = area1, area2 = area2, cross.area = n12,
                   category = c("Blood Xpert", "urine LAM"),
                   lty="blank",
                   fill = c("#CC6677", "#88CCEE"),
                   cat.just = list(c(1,1), c(0,0)))
```



```
## (polygon[GRID.polygon.4502], polygon[GRID.polygon.4503], polygon[GRID.polygon.4504], polygon[GRID.po
```

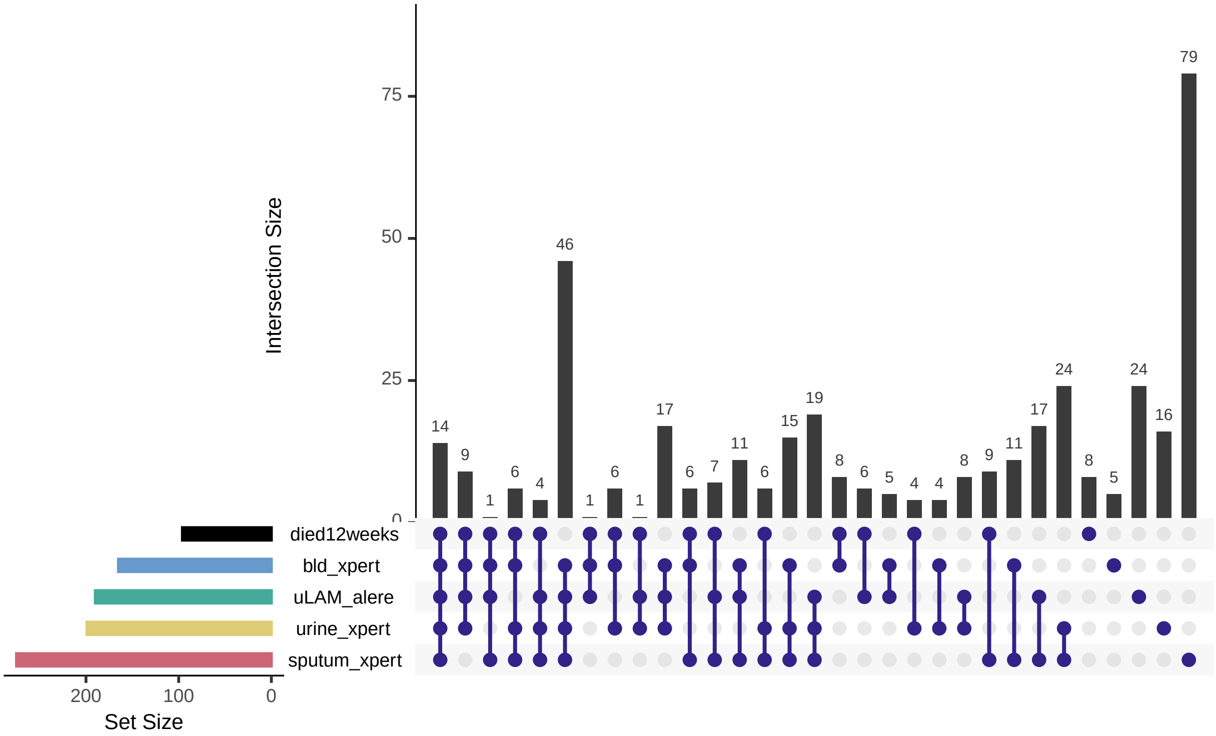
8.1 “UpSet” plot

As described by Lex and Gehlenborg in http://www.nature.com/nmeth/journal/v11/n8/abs/nmeth.3033.html

Five variables are shown: 4 rapid diagnostics and 12 week mortality. The horizontal coloured bars show the number positive for each of these 5 variables. The vertical bars show the size of the intersections between these variables indicated by the dots below the bar, e.g. 14 patients are postive for all 5 variables (the first bar) and 79 patients were positive by sputum xpert only and didn’t die (the last bar).

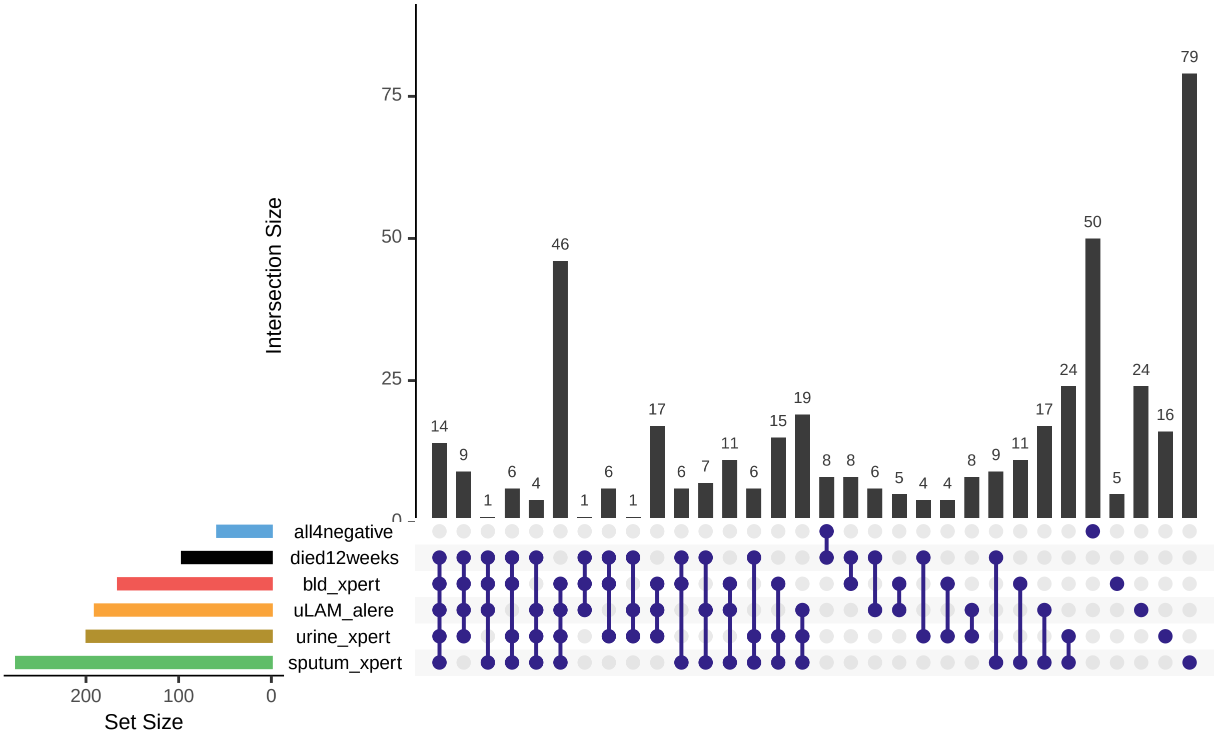
```
bld_xpert <- as.numeric(tbdf$bld_xpert_diagnosed)
urine_xpert <- as.numeric(tbdf$uGXP=="MTB" & !is.na(tbdf$uGXP))
sputum_xpert <- as.numeric(tbdf$sputum_xpert=="MTB" & !is.na(tbdf$sputum_xpert))
```

```r
uLAM_alere <- as.numeric(tbdf$ALERE_FC=="MTB" & !is.na(tbdf$ALERE_FC))
died12weeks <- as.numeric(tbdf$survival.12weeks=="Died" & !is.na(tbdf$survival.12weeks))


updf <- data.frame(bld_xpert, urine_xpert, sputum_xpert, uLAM_alere, died12weeks)

UpSetR::upset(updf,
              sets = c("bld_xpert", "urine_xpert",
                       "sputum_xpert", "uLAM_alere",
                       "died12weeks"),
              order.by = "degree", matrix.color = "#332288",
              sets.bar.color = c("#CC6677", "#DDCC77",
                                 "#44AA99", "#6699CC",
                                 "black"))
```



Unlike Venn this is scalable - we could add blood culture for example. Or can add set “no rapid diagnostic test positive” - cases missed by the 4 rapid diagnostics (but diagnosed TB by another test eg culture):

```r
all4negative <- as.numeric(bld_xpert==0 & urine_xpert==0 & sputum_xpert==0 & uLAM_alere==0)

updf <- data.frame(bld_xpert, urine_xpert, sputum_xpert, uLAM_alere, died12weeks, all4negative)

UpSetR::upset(updf,
              sets = c("bld_xpert", "urine_xpert",
                       "sputum_xpert", "uLAM_alere",
                       "died12weeks", "all4negative"),
              order.by = "degree", matrix.color = "#332288",
              sets.bar.color = c("#60BD68", "#B2912F",
                                 "#FAA43A", "#F15854",
                                   "black", "#5DA5DA"))
```

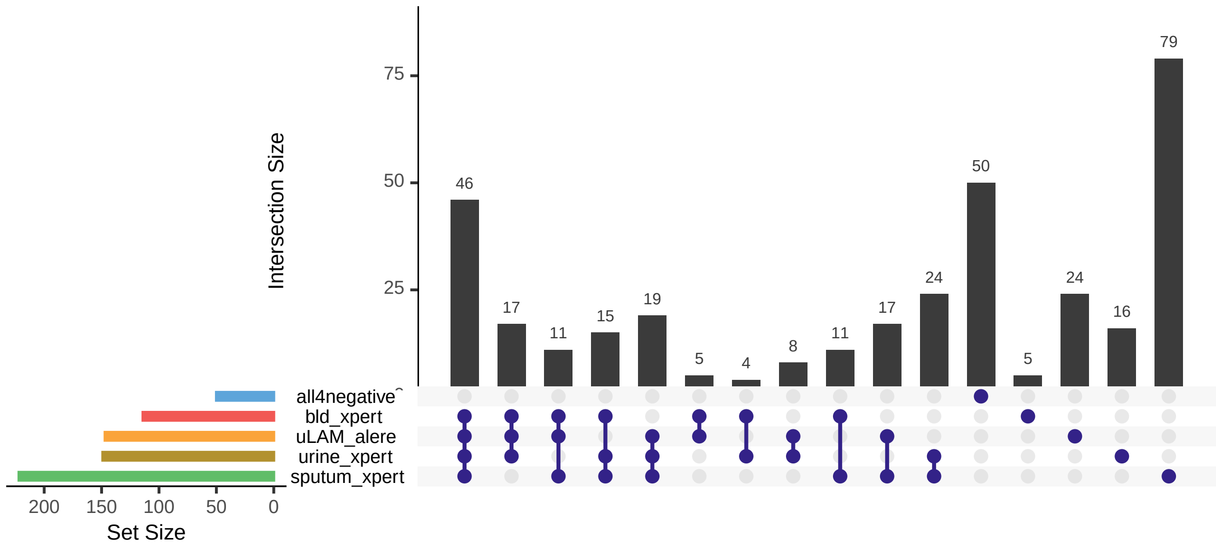


58 patients were missed by the 4 rapid diagnostics, of these 8 died before 12 weeks.

Another way to show this would be to split into patients who survived and those that died:

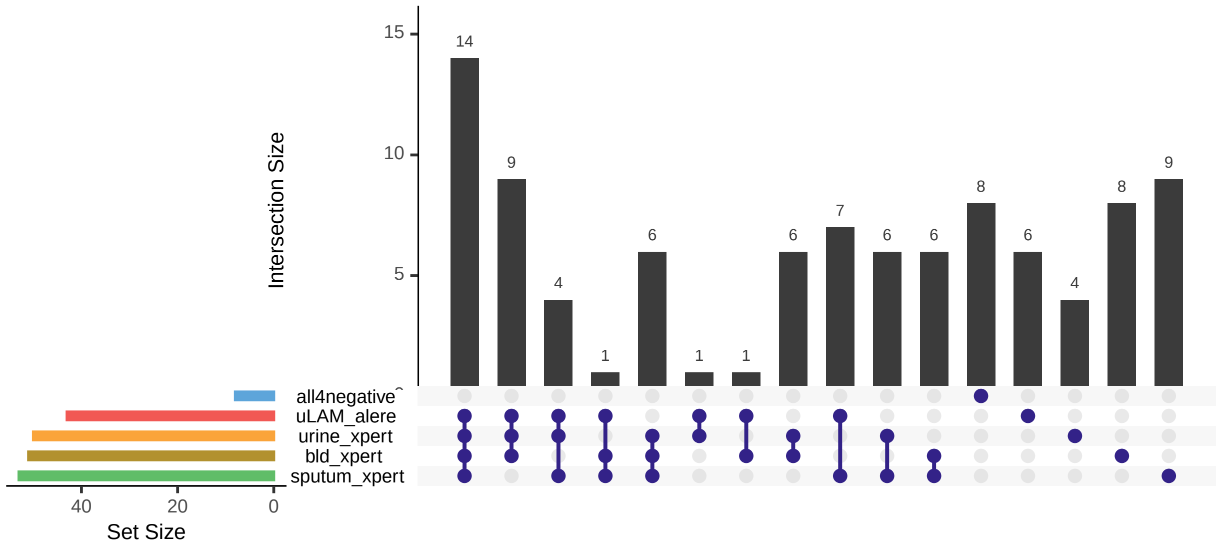
Survived

```
updf_s <- updf[updf$died12weeks==0, ]
updf_d <- updf[updf$died12weeks==1, ]

UpSetR::upset(updf_s,
              sets = c("bld_xpert", "urine_xpert",
                       "sputum_xpert", "uLAM_alere",
                       "all4negative"),
              order.by = "degree", matrix.color = "#332288",
              sets.bar.color = c("#60BD68", "#B2912F",
                                 "#FAA43A", "#F15854",
                                 "#5DA5DA"))
```



Died

```
UpSetR::upset(updf_d,
              sets = c("bld_xpert", "urine_xpert",
                       "sputum_xpert", "uLAM_alere",
                       "all4negative"),
              order.by = "degree", matrix.color = "#332288",
              sets.bar.color = c("#60BD68", "#B2912F",
                                 "#FAA43A", "#F15854",
                                 "#5DA5DA"))
```



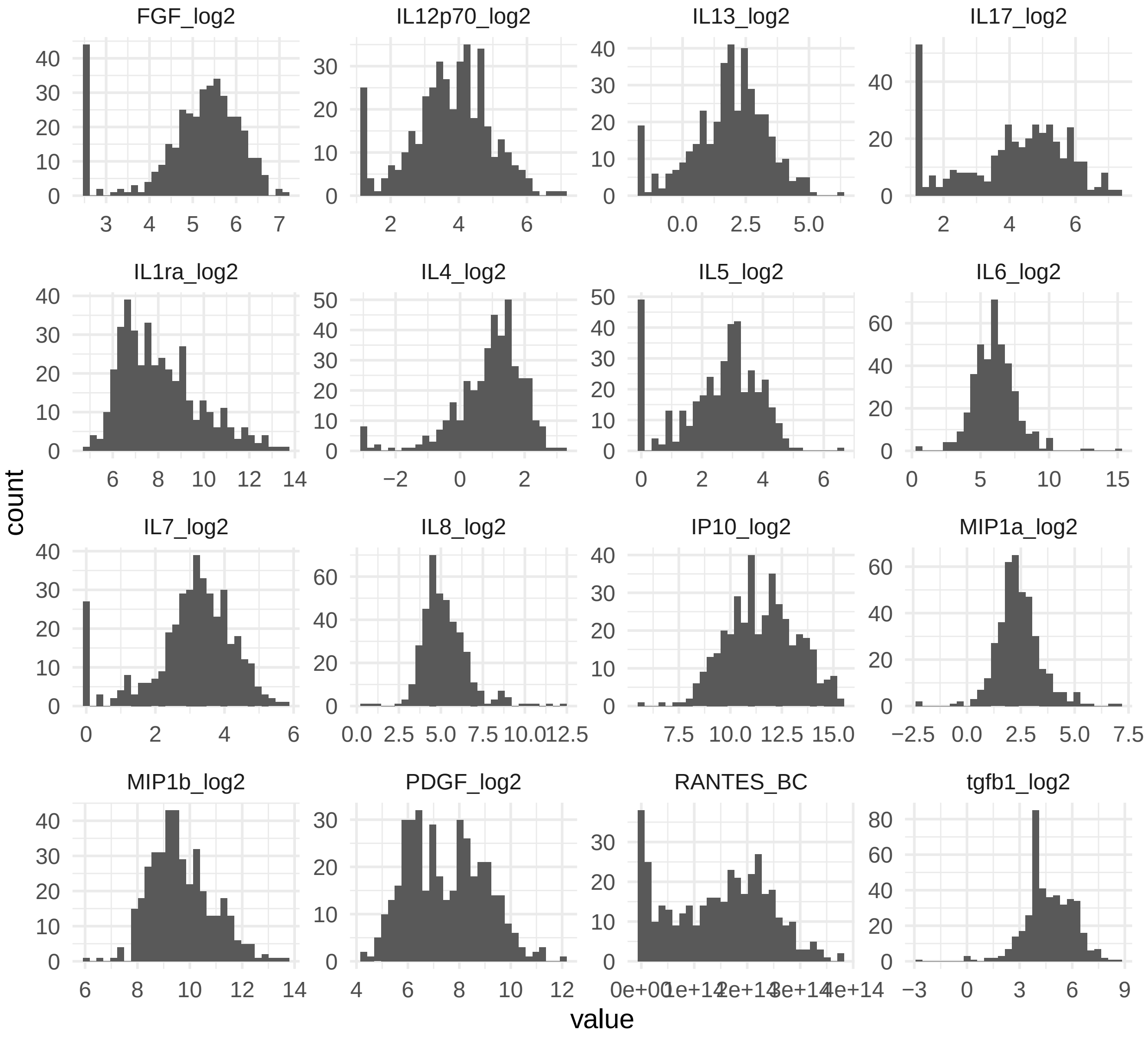
9 Clinical phenotype correlation with blood Xpert Ct values

Aim here is to assess for “dose-response” relationship between blood bacilli burden as measured by blood Xper Ct value, and markers of clinical and immunological phenotype, in particular variables we know to be associated with mortality.

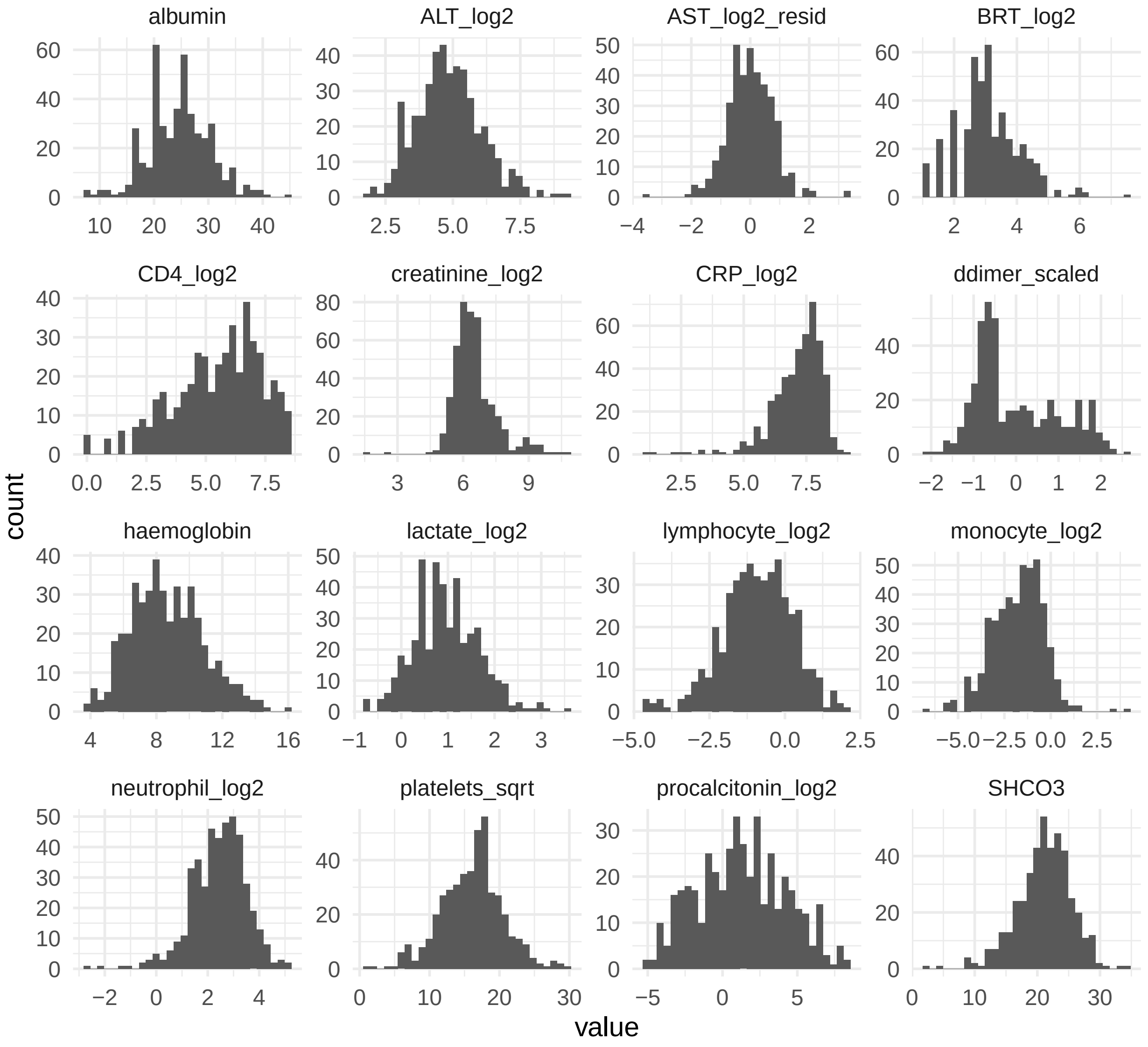
9.1 Immune markers

The 16 soluble immune mediators Charlotte identified as being most strongly associated with mortality are considered. They are transformed to be approximately normally distributed (in most cases with log transformation). q-values are given where p values are “corrected” for multiple comparison by Benjamini-Hochberg procedure for limiting false discovery rate.

```
tbdf %>%
  dplyr::transmute(blood_Xpert_CT,
       IL8_log2 = log2(Hu.IL_8), #innate and chemotaxis
       MIP1a_log2 = log2(Hu.MIP_1a),
       MIP1b_log2 = log2(Hu.MIP_1b_FI),
       IP10_log2 = log2(Hu.IP_10),

       IL6_log2 = log2(Hu.IL_6), # pro & anti inflam
       IL1ra_log2 = log2(Hu.IL_1ra_FI),

       IL17_log2 = log2(Hu.IL_17), # t cell
       IL4_log2 = log2(Hu.IL_4),
       RANTES_BC = Hu.RANTES_FI^3.4,
       IL7_log2 = log2(Hu.IL_7),
       IL12p70_log2 = log2(Hu.IL_12.p70),
       IL5_log2 = log2(Hu.IL_5),
       IL13_log2 = log2(Hu.IL_13),

       FGF_log2 = log2(Hu.FGF.basic), # growth factors
       PDGF_log2 = log2(Hu.PDGF_bb_FI),
       tgfb1_log2 = log2(tgfb1.pg.ml)
       ) -> immune_markers

# histograms
immune_markers %>%
  gather(key = var, value = value, 2:17) %>%
  ggplot(aes(value)) +
  geom_histogram() +
  facet_wrap(~var, scales = "free") +
  theme_minimal()
```

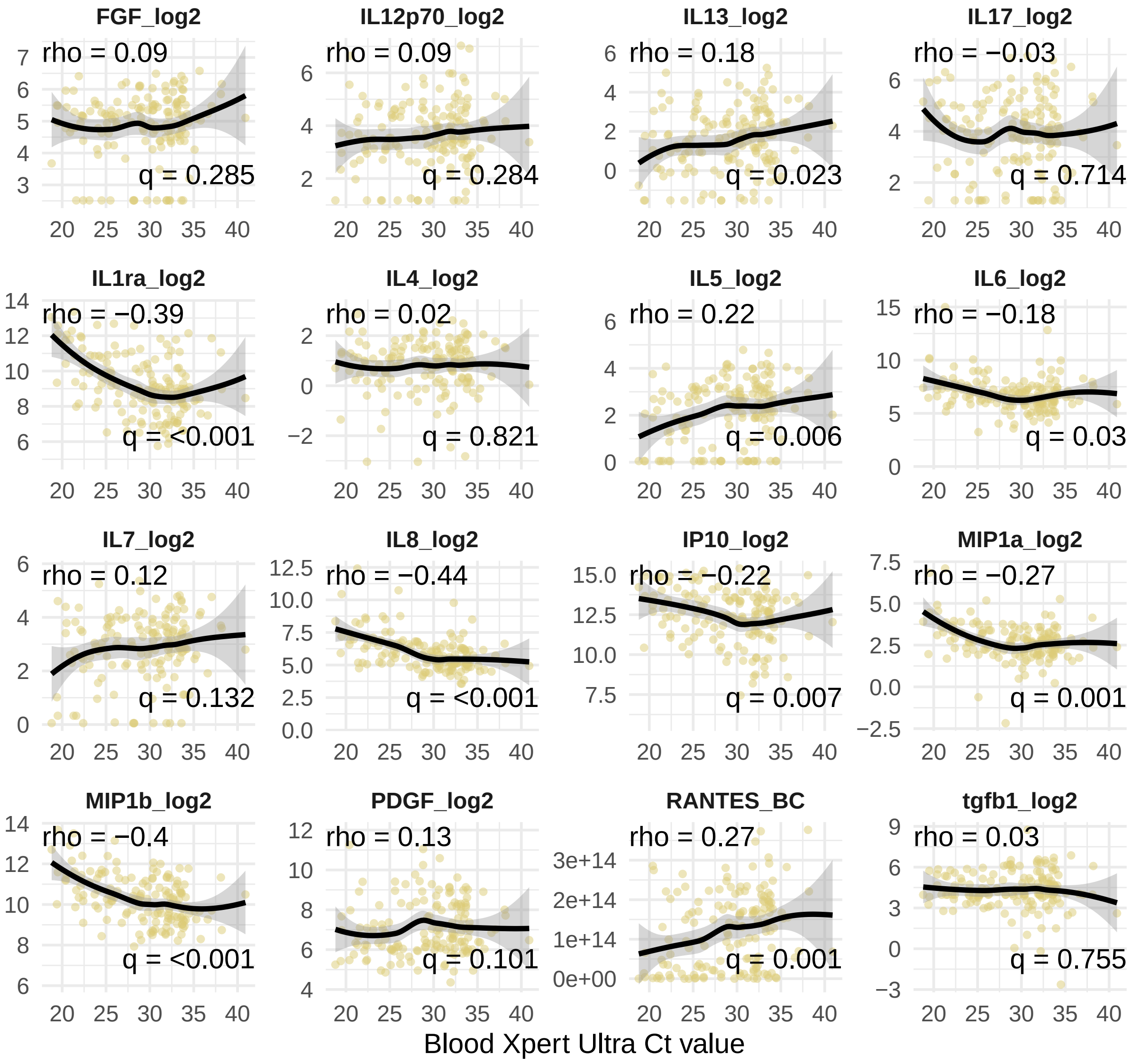


```
tbdf %>%
  mutate(
    ddimer.cpt.sqrt = sqrt(ddimer.cpt),
    ddimer.amst.sqrt = sqrt(ddimer.amst)) -> tbdf

u_cpt = mean(tbdf$ddimer.cpt.sqrt, na.rm=T)
sd_cpt = sd(tbdf$ddimer.cpt.sqrt, na.rm=T)
u_amst = mean(tbdf$ddimer.amst.sqrt, na.rm=T)
sd_amst = sd(tbdf$ddimer.amst.sqrt, na.rm=T)
ddmcpt <- (tbdf$ddimer.cpt.sqrt - u_cpt)/sd_cpt
ddmamst <- (tbdf$ddimer.amst.sqrt - u_amst)/sd_amst
tbdf$ddimer <- ifelse(!is.na(ddmcpt), ddmcpt, ddmamst)

tbdf %>%
  dplyr::transmute(blood_Xpert_CT,
                   albumin = Albumin,
                   ALT_log2 = log2(ALT),
                   AST_log2 = log2(AST),
                   BRT_log2 = log2(BRT),
                   CD4_log2 = log2(CD4+1),
                   creatinine_log2 = log2(creatinine),
                   CRP_log2 = log2(CRP),
                   ddimer_scaled = ddimer,
                   haemoglobin = Haemoglobin,
                   lactate_log2 = log2(lactate),
                   lymphocyte_log2 = log2(AbsLymphocyte),
                   monocyte_log2 = log2(AbsMonocyte),
                   neutrophil_log2 = log2(AbsNeutrophil),
                   platelets_sqrt = sqrt(Platelets),
                   procalcitonin_log2 = log2(ProCalcitonin),
                   SHCO3 = SHCO3) -> clin_markers

ast_m <- lm(AST_log2 ~ ALT_log2, data=clin_markers)

clin_markers %>%
  add_residuals(ast_m) %>%
  mutate(AST_log2_resid = resid) %>%
  select(-AST_log2, -resid) -> clin_markers


# histograms
clin_markers %>%
  gather(key = var, value = value, 2:17) %>%
  ggplot(aes(value)) +
  geom_histogram() +
  facet_wrap(~var, scales = "free") +
  theme_minimal()
```

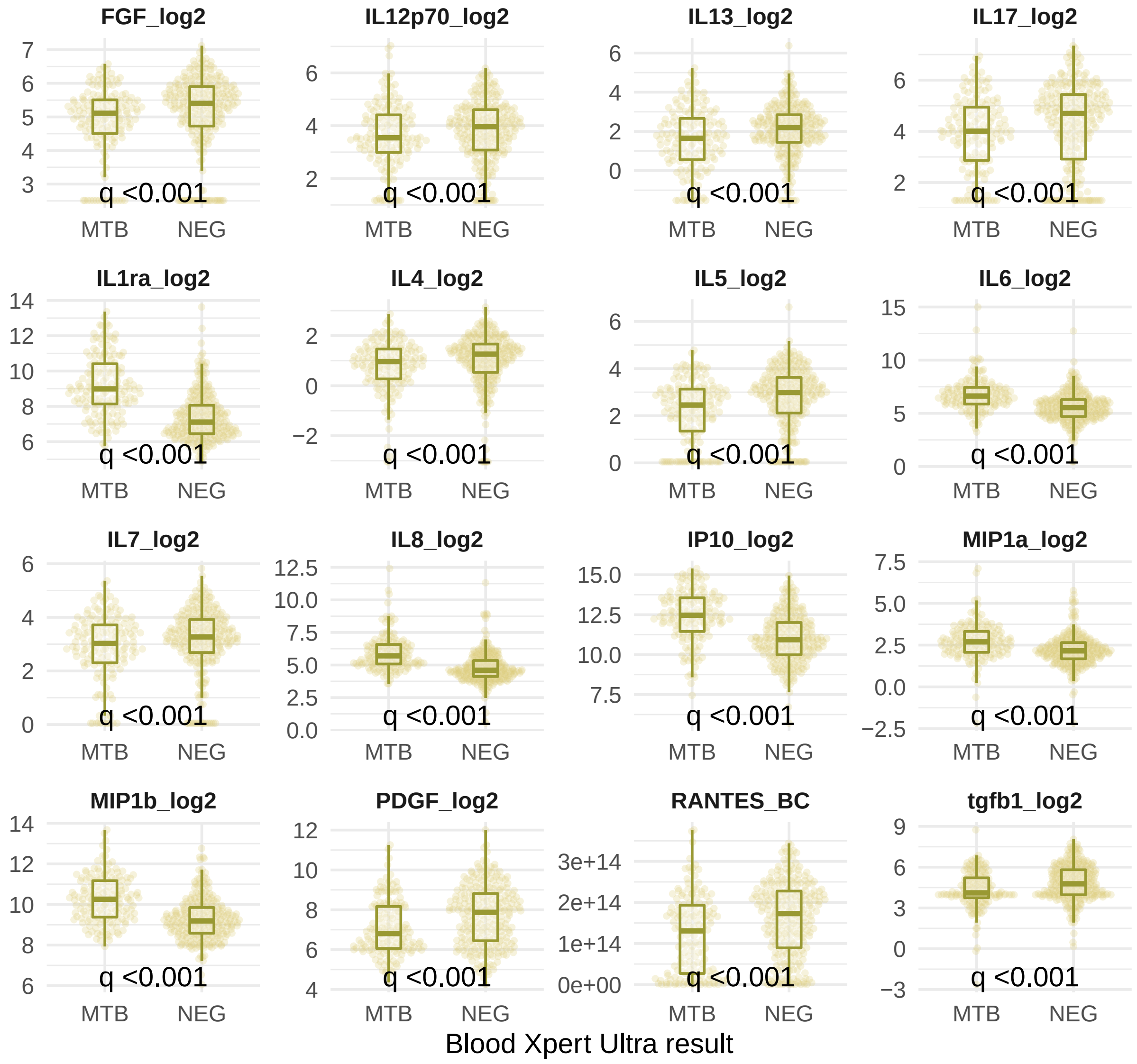


Correlation with blood Xpert CT values for each of the 16 immune markers:

```
immune_markers %>%
  filter(blood_Xpert_CT<50) %>%
  pivot_longer(-blood_Xpert_CT) %>%
  group_by(name) %>%
  do(tidy(cor.test(.$blood_Xpert_CT, .$value,
                   use="complete.cases", method = "spear"))) %>%
  select(-method, -alternative) %>%
  mutate(q = round(p.adjust(p.value, method="BH"), digits=3),
         q_value = ifelse(q<0.001, "<0.001", as.character(q)),
         estimate = round(estimate, 2)) %>%
  mutate(
    cytokine_type =
      case_when(
        name %in% c("IL8_log2", "IP10_log2",
                    "MIP1a_log2", "MIP1b_log2") ~ "innate/chemotax",
        name %in% "IL6_log2" ~ "acute inflam",
        name %in% "IL1ra_log2" ~ "anti-inflam",
        name %in% c("FGF_log2", "IL12p70_log2", "IL13_log2",
                    "IL17_log2", "IL4_log2", "IL5_log2", "IL7_log2",
                    "PDGF_log2", "RANTES_BC", "tgfb1_log2") ~ "t-cell")) %>%
  rename(var = name) -> rdf

rdf_imm <- rdf

# scatter
immune_markers %>%
  gather(key = var, value = value, 2:17) %>%
  ggplot(aes(blood_Xpert_CT, value)) +
  geom_point(colour="#DDCC77", alpha=0.5, size=0.9) +
  geom_smooth(colour="black") +
  facet_wrap(~var, scales = "free") +
  theme_minimal() +
  geom_text(data=rdf,
            aes(label = paste0("rho = ", estimate)),
            x=-Inf, y=Inf, hjust=0, vjust=1.2) +
  geom_text(data=rdf,
            aes(label = paste0("q = ", q_value)),
            x=Inf, y=-Inf, hjust=1, vjust=-1.2) +
  theme(strip.text = element_text(face = "bold")) +
  ylab("") + xlab("Blood Xpert Ultra Ct value")
```

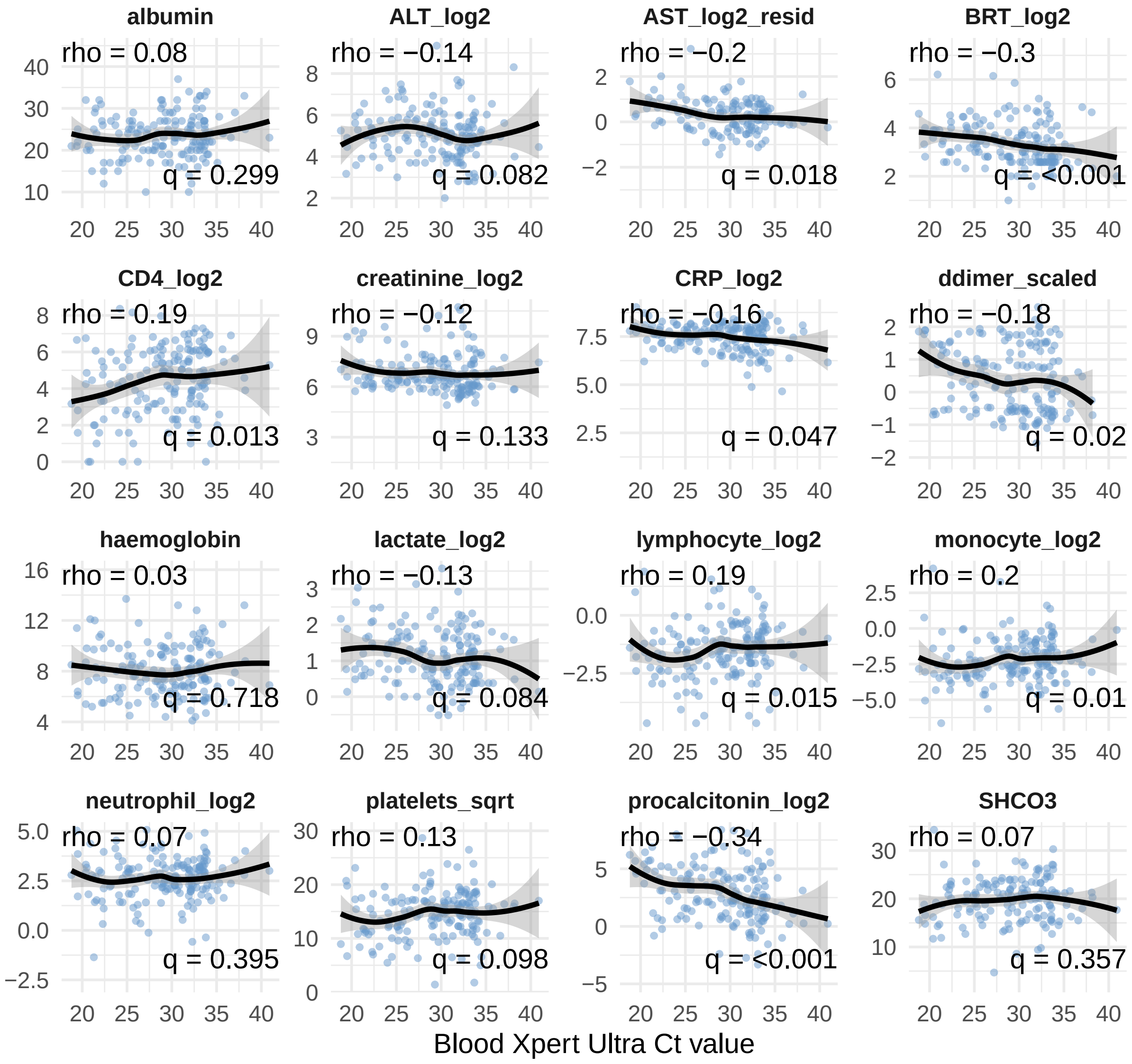


```
immune_markers %>%
  mutate(xpt = ifelse(!is.na(blood_Xpert_CT),
                      "MTB", "NEG")) %>%
  select(- blood_Xpert_CT) %>% pivot_longer(-xpt) %>%
  group_by(name) %>%
  do(tidy(t.test(.$value, .$xpt=="MTB"))) %>%
  select(-method, -alternative) %>%
  mutate(q = round(p.adjust(p.value, method="BH"), digits=3),
         q_value = ifelse(q<0.001, "<0.001", as.character(q)),
         estimate = round(estimate, 2)) %>%
  mutate(
    cytokine_type =
      case_when(
        name %in% c("IL8_log2", "IP10_log2",
                    "MIP1a_log2", "MIP1b_log2") ~ "innate/chemotax",
        name %in% "IL6_log2" ~ "acute inflam",
        name %in% "IL1ra_log2" ~ "anti-inflam",
        name %in% c("FGF_log2", "IL12p70_log2", "IL13_log2",
                    "IL17_log2", "IL4_log2", "IL5_log2", "IL7_log2",
                    "PDGF_log2", "RANTES_BC", "tgfb1_log2") ~ "t-cell")) -> tdf

immune_markers %>%
    mutate(xpt = ifelse(!is.na(blood_Xpert_CT),
                        "MTB", "NEG")) %>%
    select(- blood_Xpert_CT) %>% pivot_longer(-xpt) %>%
    ggplot(aes(xpt, value)) +
    geom_quasirandom(colour="#DDCC77", alpha=0.25, size=0.7) +
    geom_boxplot(width=0.25, colour="#999933", fill="white", alpha=0.3,
                 outlier.alpha = 0) +
    facet_wrap(~name, scales = "free") +
    theme_minimal() +
    geom_text(data=tdf,
              aes(label = paste0("q ", q_value)),
              x=1.5, y=-Inf, hjust=0.5, vjust=-0.3) +
  theme(strip.text = element_text(face = "bold")) +
  xlab("Blood Xpert Ultra result") + ylab("")
```

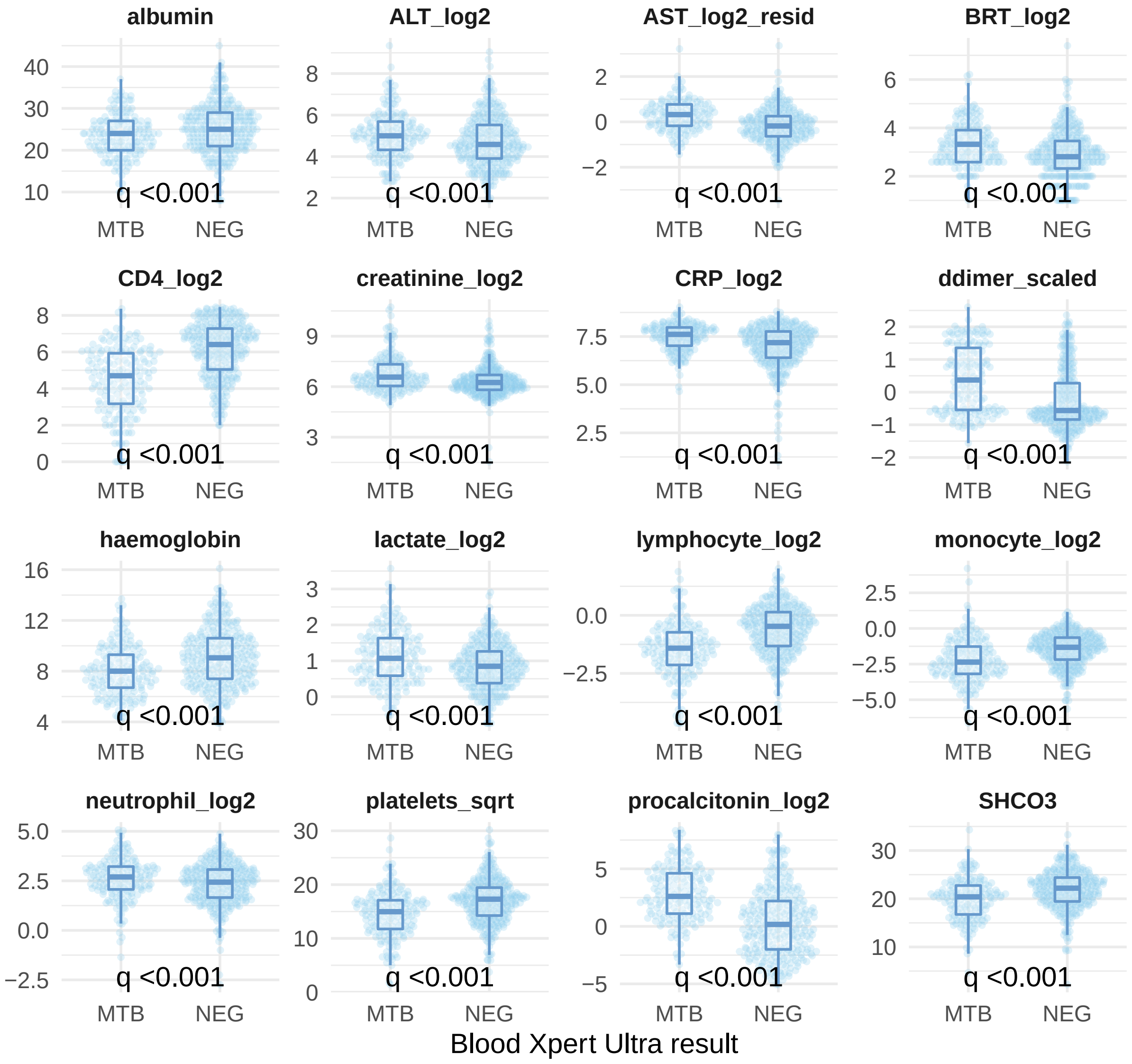


Correlation with blood Xpert CT values for each of the 16 clinical markers:

```
clin_markers %>%
  filter(blood_Xpert_CT<50) %>%
  pivot_longer(-blood_Xpert_CT) %>%
  group_by(name) %>%
  do(tidy(cor.test(.$blood_Xpert_CT, .$value,
                   use="complete.cases", method = "spear"))) %>%
  select(-method, -alternative) %>%
  mutate(q = round(p.adjust(p.value, method="BH"), digits=3),
         q_value = ifelse(q<0.001, "<0.001", as.character(q)),
         estimate = round(estimate, 2)) %>%
  mutate(
    marker_type =
      case_when(
        name %in% c("CRP_log2", "lactate_log2",
                    "procalcitonin_log2") ~ "acute inflam",
        name %in% c("albumin", "haemoglobin") ~ "chronic inflam",
        name %in% c("CD4_log2", "lymphocyte_log2", "monocyte_log2", "neutrophil_log2") ~ "WBC",
        name %in% c("ALT_log2", "BRT_log2") ~ "liver",
        name %in% c("ddimer_scaled", "platelets_sqrt") ~ "coagulation",
        name %in% c("creatinine_log2","SHCO3") ~ "renal",
        name %in% "AST_log2_resid" ~ "mitochondrial")) -> rdf

rdf_clin <- rdf

# scatter
clin_markers %>%
  gather(key = name, value = value, 2:17) %>%
  ggplot(aes(blood_Xpert_CT, value)) +
  geom_point(colour="#6699CC", alpha=0.5, size=0.8) +
  geom_smooth(colour="black") +
  facet_wrap(~name, scales = "free") +
  theme_minimal() +
  geom_text(data=rdf,
            aes(label = paste0("rho = ", estimate)),
            x=-Inf, y=Inf, hjust=0, vjust=1.2) +
  geom_text(data=rdf,
            aes(label = paste0("q = ", q_value)),
            x=Inf, y=-Inf, hjust=1, vjust=-1.2) +
  theme(strip.text = element_text(face = "bold")) +
  ylab("") + xlab("Blood Xpert Ultra Ct value")
```

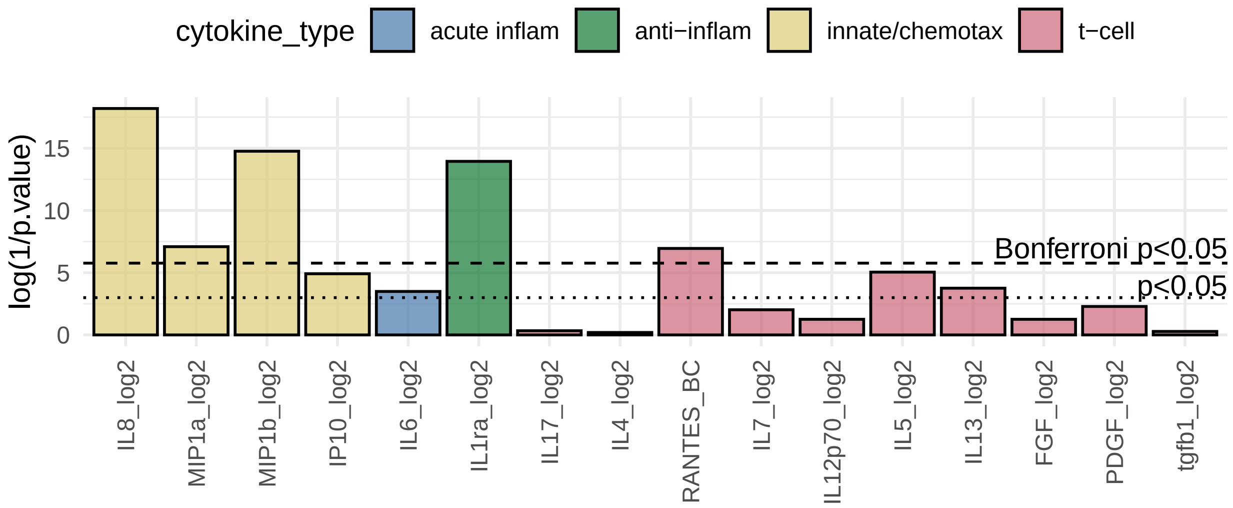


```
clin_markers %>%
  mutate(xpt = ifelse(!is.na(blood_Xpert_CT),
                      "MTB", "NEG")) %>%
  select(- blood_Xpert_CT) %>% pivot_longer(-xpt) %>%
  group_by(name) %>%
  do(tidy(t.test(.$value, .$xpt=="MTB"))) %>%
  select(-method, -alternative) %>%
  mutate(q = round(p.adjust(p.value, method="BH"), digits=3),
         q_value = ifelse(q<0.001, "<0.001", as.character(q)),
         estimate = round(estimate, 2)) %>%
  mutate(
    marker_type =
      case_when(
        name %in% c("CRP_log2", "lactate_log2",
                    "procalcitonin_log2") ~ "acute inflam",
        name %in% c("albumin", "haemoglobin") ~ "chronic inflam",
        name %in% c("CD4_log2", "lymphocyte_log2", "monocyte_log2", "neutrophil_log2") ~ "WBC",
        name %in% c("ALT_log2", "BRT_log2") ~ "liver",
        name %in% c("ddimer_scaled", "platelets_sqrt") ~ "coagulation",
        name %in% c("creatinine_log2","SHCO3") ~ "renal",
        name %in% "AST_log2_resid" ~ "mitochondrial")) -> tdf

clin_markers %>%
    mutate(xpt = ifelse(!is.na(blood_Xpert_CT),
                        "MTB", "NEG")) %>%
    select(- blood_Xpert_CT) %>% pivot_longer(-xpt) %>%
    ggplot(aes(xpt, value)) +
    geom_quasirandom(colour="#88CCEE", alpha=0.25, size=0.7) +
    geom_boxplot(width=0.25, colour="#6699CC", fill="white", alpha=0.3,
                 outlier.alpha = 0) +
    facet_wrap(~name, scales = "free") +
    theme_minimal() +
    geom_text(data=tdf,
              aes(label = paste0("q ", q_value)),
              x=1.5, y=-Inf, hjust=0.5, vjust=-0.3) +
  xlab("Blood Xpert Ultra result") + ylab("") +
  theme(strip.text = element_text(face = "bold"))
```

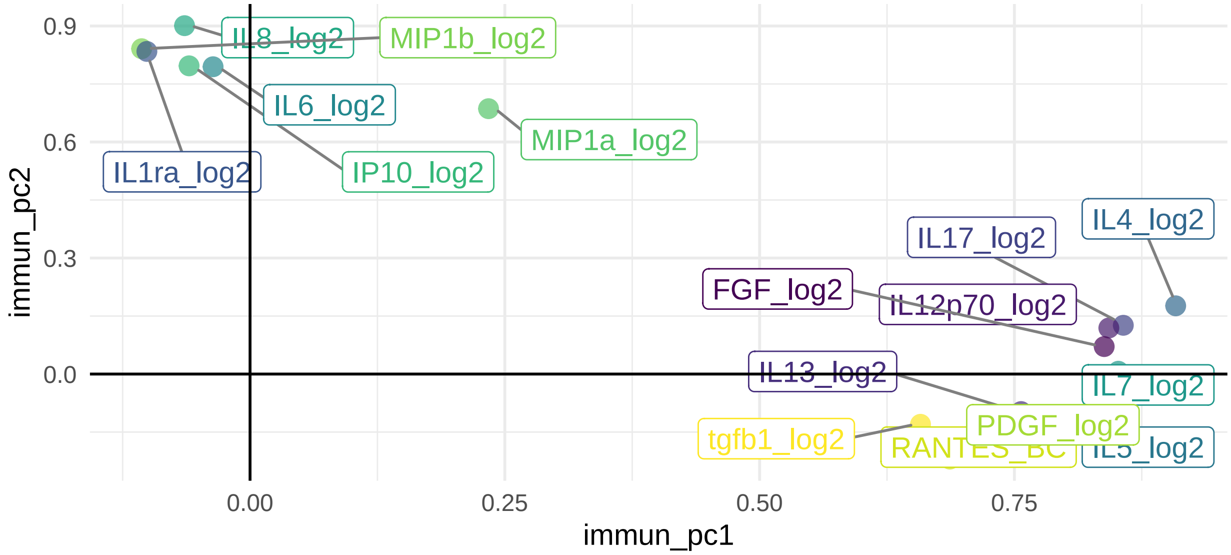


Manhattan plot showing strength of association with blood Xpert CT values, cytokines grouped by approxiamte function.

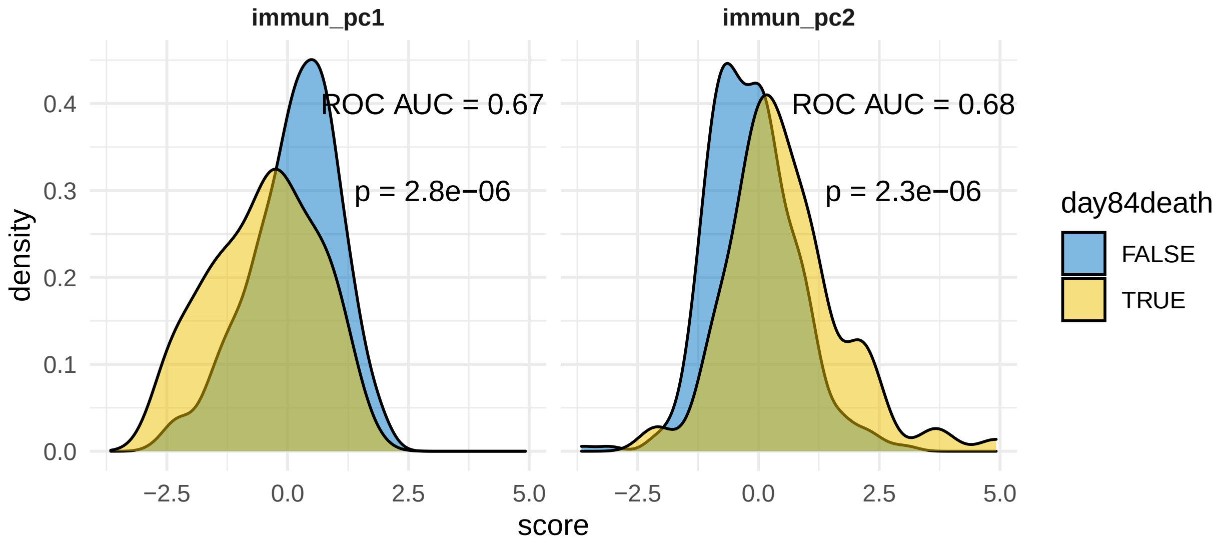
```r
# manhattan plot
rdf_imm %>% ungroup() %>%
  mutate(
    var = factor(
      var,
      levels = c(
       "IL8_log2", "MIP1a_log2", "MIP1b_log2", "IP10_log2",
       "IL6_log2", "IL1ra_log2",
       "IL17_log2", "IL4_log2", "RANTES_BC", "IL7_log2",
       "IL12p70_log2", "IL5_log2", "IL13_log2",
       "FGF_log2", "PDGF_log2", "tgfb1_log2"))) %>%
  ggplot(aes(var, log(1/p.value), fill=cytokine_type)) +
  geom_bar(colour="black", alpha=0.7, stat = "identity") +
  theme_minimal() +
  xlab("") + ylab("log(1/p.value)") +
  theme(axis.text.x =
          element_text(angle=90, hjust = 1, vjust=0.5),
        legend.position = "top") +
  scale_fill_ptol() +
  geom_hline(yintercept = log(1/(0.05/16)), linetype=2) +
  geom_hline(yintercept = log(1/(0.05)), linetype=3) +
  annotate("text", x =Inf, y=7, hjust=1,
           label="Bonferroni p<0.05") +
  annotate("text", x =Inf, y=4, hjust=1,
           label="p<0.05")
```



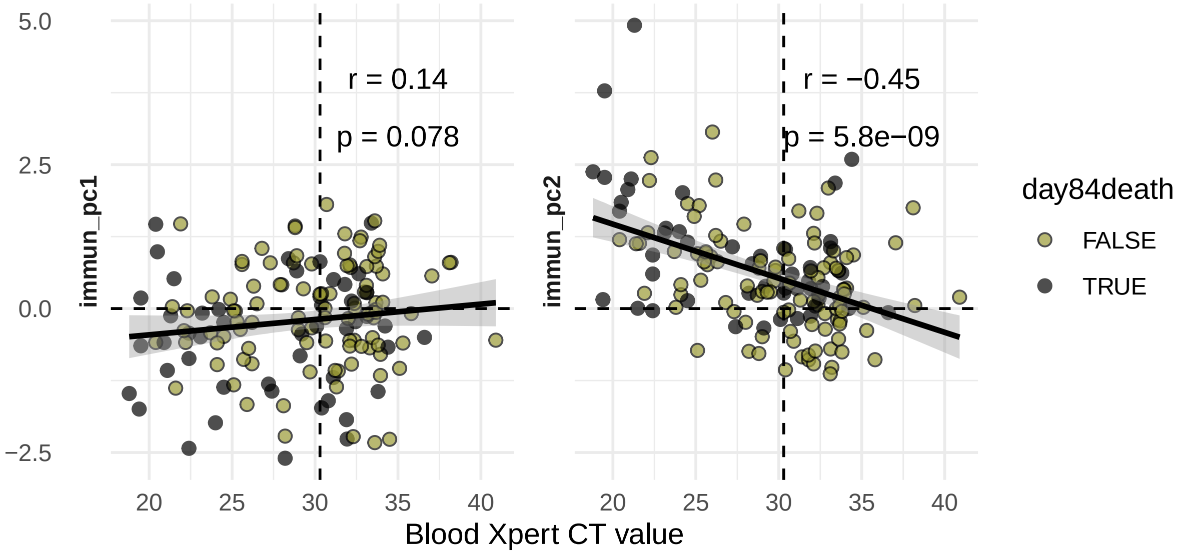
```
#PCA

pc <- principal(immune_markers[,-1], nfactors = 2, rotate="varimax")

immune_pc <- data.frame(
  PC1 = pc$loadings[,1],
  PC2 =  pc$loadings[,2],
                   assay = names(pc$loadings[,1]))


ggplot(immune_pc,
       aes(PC1, PC2,
           colour=assay)) +
  geom_point(size=3, alpha=0.7) +
  geom_label_repel(
        aes(PC1, PC2,
            label = assay),
        box.padding = 0.35, point.padding = 0.5,
        segment.color = 'grey50') +
  theme_minimal() +
  scale_color_viridis_d() +
  theme(legend.position = "none") +
  geom_vline(xintercept = 0) +
  geom_hline(yintercept = 0) +
  xlab("immun_pc1") +
  ylab("immun_pc2")
```



```
cbind(tbdf,
      data.frame(immun_pc1 = pc$scores[,1],
                 immun_pc2 = pc$scores[,2])) -> tbdf

tdf <- data.frame(
  pc = c("immun_pc1", "immun_pc2"),
  auc =
  c(round(auc(roc(tbdf$day84death, tbdf$immun_pc1)), 2),
  round(auc(roc(tbdf$day84death, tbdf$immun_pc2)), 2)),
  p =
    c(signif(t.test(tbdf$immun_pc1 ~ tbdf$day84death)$p.value, 2),
      signif(t.test(tbdf$immun_pc2 ~ tbdf$day84death)$p.value, 2))
  )

tbdf[!is.na(tbdf$day84death), ] %>%
    select(day84death, blood_Xpert_CT, immun_pc1, immun_pc2) %>%
    gather(key = pc, value = score, 3:4) %>%
  ggplot(aes(score)) +
  geom_density(alpha=0.5,
               aes(fill=day84death)) +
  theme_minimal() +
  scale_fill_jco() +
  geom_text(data=tdf,
              aes(label = paste0("ROC AUC = ", auc)),
              x=3, y=0.4) +
  geom_text(data=tdf,
              aes(label = paste0("p = ", p)),
              x=3, y=0.3) +
  theme(strip.text = element_text(face = "bold")) +
  facet_wrap(~pc)
```



```
rdf <- data.frame(
  pc = c("immun_pc1", "immun_pc2"),
  r = c(round(cor.test(tbdf$blood_Xpert_CT, tbdf$immun_pc1)$estimate, 2),
      round(cor.test(tbdf$blood_Xpert_CT, tbdf$immun_pc2)$estimate, 2)),
  p = c(signif(cor.test(tbdf$blood_Xpert_CT, tbdf$immun_pc1)$p.value, 2),
      signif(cor.test(tbdf$blood_Xpert_CT, tbdf$immun_pc2)$p.value, 2))
  )


tbdf[!is.na(tbdf$day84death) & !is.na(tbdf$blood_Xpert_CT), ] %>%
    select(day84death, blood_Xpert_CT, immun_pc1, immun_pc2) %>%
    gather(key = pc, value = score, 3:4) %>%
    ggplot(aes(blood_Xpert_CT, score)) +
    geom_point(size=2, alpha=0.7, shape=21,
               aes(fill=day84death)) +
    geom_smooth(colour="black", method="lm") +
    theme_minimal() +
    geom_vline(xintercept = median(tbdf$blood_Xpert_CT, na.rm = TRUE), linetype=2) +
    geom_hline(yintercept = 0, linetype=2) +
    scale_fill_manual(values = c("#999933", "black")) +
    xlab("Blood Xpert CT value") +
    ylab("") +
    geom_text(data=rdf,
              aes(label = paste0("r = ", r)),
              x=35, y=4) +
    geom_text(data=rdf,
              aes(label = paste0("p = ", p)),
              x=35, y=3) +
    theme(strip.text = element_text(face = "bold")) +
    facet_wrap(~pc, strip.position = "left")
```


5.0
2.5
0.0
−2.5
immun_pc1
immun_pc2
r = 0.14
p = 0.078
r = −0.45
p = 5.8e−09
20
25
30
35
40
20
25
30
35
40
Blood Xpert CT value
day84death
FALSE
TRUE