

# MTBBSI detection & quantification serial sub-study KDHTB: analysis for manuscript

dauidadambarr

23/07/2020

## Contents

<b>1 Packages and functions</b>	<b>1</b>
<b>2 Raw data read in</b>	<b>1</b>
<b>3 Initial data wrangling</b>	<b>2</b>
<b>4 Raw qualitative headline results</b>	<b>4</b>
<b>5 Raw quantitative headline results</b>	<b>7</b>
<b>6 Summary measures from quantitative data</b>	<b>10</b>
<b>7 Clinical description cohort</b>	<b>16</b>
<b>8 Agreement between methods</b>	<b>28</b>
<b>9 Ordinal scaling data</b>	<b>35</b>
<b>10 Extracting data from KDHTB cohort to use as priors in ordinal regression model</b>	<b>40</b>
<b>11 Ordinal regression model relating blood bacilliary load dynamics to mortality</b>	<b>43</b>
11.1 Summary of model . . . . .	47
11.2 Formal hypothesis testing and parameter posterior distributions . . . . .	48
11.3 Expected probabilities of specific ordinal values by time and assay . . . . .	50
11.4 Prediction intervals for new patient . . . . .	56
11.5 Model diagnostics . . . . .	57
11.6 Checking sensitivity to priors . . . . .	60
<b>12 Serial observations</b>	<b>67</b>
<b>13 Bacilli lengths</b>	<b>67</b>

## 1 Packages and functions

## 2 Raw data read in

```

gxp <- bind_rows(read_csv("tidy_gxp.csv"), read_csv("tidy_gxp2.csv")) # two batches
dmn <- read_csv("DMN.csv")
mfl <- read_csv("MFL.csv")
clin <- read_csv("log.csv")
nhls <- read_csv("nhls.csv")
blds <- read_csv("baseline_blds.csv")

# Assign new PIDs which are easier to understand (specific to substudy, not contig with KDHTB)
pids <- data.frame(
  study_id = unique(gxp$study_id),
  pid = paste0("pid",
    c(9,10,11,12,13,14,1,2,15,16,17,3,18,19,
      20,21,4,5,6,22,7,8,23,24,25,26,27,28))
)

oc <- data.frame(
  study_id = c("pid9", "pid10", "pid11", "pid12", "pid13", "pid14",
    "pid1", "pid2", "pid15", "pid16", "pid17", "pid3",
    "pid18", "pid19", "pid20", "pid21", "pid4", "pid5",
    "pid6", "pid22", "pid7", "pid8", "pid23", "pid24",
    "pid25", "pid26", "pid27", "pid28"),
  died = c(0,0,0,0,0,0,
    1,1,0,0,0,1,
    0,0,0,0,1,1,
    1,0,1,1,0,0,
    0,0,0,0))

```

### 3 Initial data wrangling

```

#### GXP
gxp %>% select(-X1) %>%
  mutate(
    rpoB1_CT = na_if(rpoB1_CT, 0), # misssing from negative are coded 0 - fix
    rpoB2_CT = na_if(rpoB2_CT, 0),
    rpoB3_CT = na_if(rpoB3_CT, 0),
    rpoB4_CT = na_if(rpoB4_CT, 0),
    IS1081_IS6110_CT = na_if(IS1081_IS6110_CT, 0),
    result = factor(result,
      levels = c("ERROR", "MTB NOT DETECTED", "MTB DETECTED TRACE",
        "MTB DETECTED VERY LOW", "MTB DETECTED LOW",
        "MTB DETECTED MEDIUM", "MTB DETECTED HIGH")),
    timepoint = round(timepoint/24, 1)) %>%
  left_join(pids, by = "study_id") %>%
  mutate(study_id = pid) %>% select(-pid) -> gxp

#### DMN
dmn %>%
  dplyr::mutate(DMN_ml =
    round((2850/3) * (bacilli/random_fields), 0), # FOV calculation for /ml
    timepoint = round(timepoint/24, 1)
  ) %>%

```

```

left_join(pids, by = "study_id") %>%
mutate(study_id = pid) %>%
dplyr::select(study_id, timepoint, qualitative_result, DMN_ml) -> dmn

#### MFL
# mfl$MFL[mfl$TTP<10 & !is.na(mfl$TTP)] # these are mixed (contam) growth but MTB was recovered...
mfl %>%
  mutate(
    TTP = ifelse(TTP<10, NA, TTP),
    timepoint = round(timepoint/24, 1)
  ) %>% left_join(pids, by = "study_id") %>%
  mutate(study_id = pid) %>% select(-pid, -X10) -> mfl

#### clinical log data
clin %>%
  left_join(pids, by="study_id") %>%
  left_join(blds, by="study_id") %>%
  mutate(outcome = ifelse(outcome_12wk=="died", "Died", "Survived"),
    age =
      as.numeric(as.Date(dateRecruited,
        format="%d/%m/%Y") -
        as.Date(dob,
          format="%d/%m/%Y"))/364.75
  ) %>%
  select(
    study_id = pid,
    age, sex, ARVstatus,

    cd4, creat = Cr_pp,
    pH, HCO3, BE, lactate, pO2, pCO2, glucose,

    HR, sBP, RR, sats, temp, GCS,
    walks.unaided, self.feed, wasting,
    flushed, sweating, cool.periph, cap.refil,
    ancil.muscles, nasal.flare, full.sentences,
    candida, oedema, periph.LN,
    splenomegaly, hepatomegaly, ascities, doughy.abdo, tender.abdo,

    CXR.pleff, CXR.inf.character, CXR.inf.lobes,
    CXR.inf.bilat, CXR.inf.symmetrical, CXR.LN,

    liver_enlarged, liver_echogen, liver_hypoecho_lesions,
    spleen_enlarged, spleen_hypoecho_lesions,
    kidney_size, kidney_hydronephrosis, kidney_echogen, kidney_CMD_loss,
    frank_ascities, pleural_effusion, pericardial_effusion,
    adenopathy, GB_oedema, GB_sludge, CBD, veins,

    T_4, HR_4, RR_4, BP_4, GCS_4, sweat_4, flush_4,
    cool_4, lethargy_4, ambulant_4, eating_4, improved_4,
    T_24, HR_24, RR_24, BP_24, GCS_24, sweat_24, flush_24,
    cool_24, lethargy_24, ambulant_24, eating_24, improved_24,

```

```

T_48, HR_48, RR_48, BP_48, GCS_48, sweat_48, flush_48,
cool_48, lethargy_48, ambulant_48, eating_48, improved_48,
T_72, HR_72, RR_72, BP_72, GCS_72, sweat_72, flush_72,
cool_72, lethargy_72, ambulant_72, eating_72, improved_72
) -> clin

```

## 4 Raw qualitative headline results

```

# mung out the qualitative results by pt-sample-timepoint
# to have same levels notation across methods
mfl %>%
  select(study_id, timepoint, type, MFL) %>%
  mutate(MFL = str_replace(MFL, "contam|lost_viability",
                           replacement = "Not available"),
         MFL = replace_na(MFL, "Not available"),
         MFL = str_replace(MFL, "MTB",
                           replacement = "M.tb detected")) %>%
  pivot_wider(id_cols = c("study_id", "timepoint"),
              names_from = "type",
              values_from = "MFL") %>%
  rename(mfl_pellet=pellet, mfl_lystate=lysate) -> qual_mfl

gxp %>%
  select(study_id, timepoint, sample_type, result) %>%
  filter( # these are duplicates which mess up the later pivot wider
         !(study_id==871 & timepoint==72 & result=="MTB NOT DETECTED"),
         !(study_id==872 & timepoint==24 & result=="MTB NOT DETECTED") ) %>%
  mutate(result = case_when(
    str_detect(result, "ERROR") ~ "Not available",
    str_detect(result, "MTB DETECTED") ~ "M.tb detected",
    str_detect(result, "MTB NOT DETECTED") ~ "negative")) %>%
  filter(sample_type=="bldextract" | sample_type=="urine") %>%
  pivot_wider(id_cols = c("study_id", "timepoint"),
              names_from = "sample_type",
              values_from = "result") %>%
  # mutate(bldextract = ifelse(
  #   is.na(bldextract) & !is.na(buffy), buffy, bldextract # a single case
  # )) %>%
  select(study_id, timepoint,
         bld_xpert = bldextract, urn_xpert = urine) -> qual_gxp

dmn %>%
  mutate(qualitative_result =
    str_replace(qualitative_result, "positive",
              replacement = "M.tb detected"),
         qualitative_result =
    str_replace(qualitative_result,
              "non-quantifiable M.tb detected",
              replacement = "negative"),
         qualitative_result =

```

```

      replace_na(qualitative_result, "Not available")) %>%
select(study_id, timepoint,
      dmn_micro = qualitative_result) -> qual_dmn

qual_mfl %>%
  full_join(qual_gxp, by=c("study_id", "timepoint")) %>%
  full_join(qual_dmn, by=c("study_id", "timepoint")) %>%
  filter(timepoint!=96) -> qual
rm(qual_dmn, qual_gxp, qual_mfl)

qual %>%
  mutate(
    study_id = factor(study_id, levels = paste0("pid", 1:28)),
    mfl_pellet = ifelse(is.na(mfl_pellet), "Not available", mfl_pellet),
    mfl_lysate = ifelse(is.na(mfl_lysate), "Not available", mfl_lysate),
    bld_xpert = ifelse(is.na(bld_xpert), "Not available", bld_xpert),
    urn_xpert = ifelse(is.na(urn_xpert), "Not available", urn_xpert),
    dmn_micro = ifelse(is.na(dmn_micro), "Not available", dmn_micro),
    mfl_culture = case_when(
      mfl_pellet=="M.tb detected" |
        mfl_lysate=="M.tb detected" ~ "M.tb detected",
      mfl_pellet=="Not available" &
        mfl_lysate=="Not available" ~ "Not available",
      mfl_pellet=="negative" &
        mfl_lysate=="negative" ~ "negative",
      mfl_pellet=="Not available" &
        mfl_lysate=="negative" ~ "negative",
      mfl_pellet=="negative" &
        mfl_lysate=="Not available" ~ "negative"
    )
  ) %>%
  select(-mfl_pellet, -mfl_lysate) %>%
  filter(timepoint < 4) -> qual

qual %>%
  pivot_longer(cols = 3:6,
    names_to = "assay",
    values_to = "Result") %>%
  mutate(assay = factor(assay,
    levels = c("mfl_culture", "dmn_micro",
      "bld_xpert", "urn_xpert"),
    labels = c("MFL blood culture", "DMN microscopy",
      "Blood Xpert-ultra", "Urine Xpert"))) %>%
  ggplot(aes(as.factor(timepoint), assay)) +
  geom_tile(aes(fill=Result), colour="white", alpha=0.8) +
  scale_fill_manual(values=c(viridis(8, option="E")[c(2,6)], "grey80")) +
  theme_minimal() +
  facet_wrap(~study_id, ncol = 7) +
  theme(legend.position = "top",

```

```

    panel.spacing = unit(0, "lines"),
    axis.text.x = element_text(size=7, angle=90, hjust=1, vjust=0.5)) +
  ylab("") + xlab("Time from start treatment (days)") -> g_tileplot

qual %>%
  pivot_longer(3:6, names_to = "assay", values_to = "result") %>%
  group_by(timepoint, assay) %>%
  count(result) %>%
  pivot_wider(names_from = result, values_from = n) %>%
  rename_all(list(~make.names(.))) %>%
  rename_with(~ tolower(gsub(".", "_", .x, fixed = TRUE))) %>%
  mutate(
    valid_tests = sum(m_tb_detected, negative, na.rm = T),
    proportion_positive = m_tb_detected / valid_tests
  ) %>%
  select(timepoint, assay, valid_tests, proportion_positive) -> qual_km_df

qual_km_df %>%
  select(timepoint, valid_tests) %>%
  mutate(valid_tests = ifelse(valid_tests==2, 0, valid_tests)) %>%
  pivot_wider(names_from = timepoint,
    values_from = valid_tests) %>%
  mutate(assay = factor(assay,
    levels = c("urn_xpert", "dmn_micro",
      "bld_xpert", "mfl_culture" ),
    labels = c("Urine Xpert", "DMN microscopy",
      "Blood Xpert-ultra", "MFL blood culture" ))) -> atrisk_table

atrisk_table <- as.data.frame(atrisk_table)
rownames(atrisk_table) <- atrisk_table$assay

ggtexttable(atrisk_table[,2:6], theme = ttheme("light")) %>%
  tab_add_title("Number valid tests at timepoint:") -> t1

qual_km_df %>%
  filter(!(timepoint==0.2 & assay=="urn_xpert")) %>%
  mutate(assay = factor(assay,
    levels = c("urn_xpert", "dmn_micro",
      "bld_xpert", "mfl_culture" ),
    labels = c("Urine Xpert", "DMN\nmicroscopy",
      "Blood\nXpert-ultra", "MFL\nblood culture" ))) %>%
  ggplot(aes(timepoint, proportion_positive, colour=assay)) +
  geom_point() +
  geom_line() +
  theme_dab() +
  ylim(0,1) +
  scale_colour_manual(
    values = c( "#f68f46ff", "#1F968BFF",
      "#453781FF", "#b8627dff" )) +
  ylab("Proportion\ntests positive") +

```

```

xlab("Time from start treatment (days)") -> g_km

# grid.arrange(g_km, t1, nrow=2)

g_tileplot + g_km +
  plot_annotation(tag_levels = "A") +
  plot_layout(ncol=2, widths = c(5,3))

```

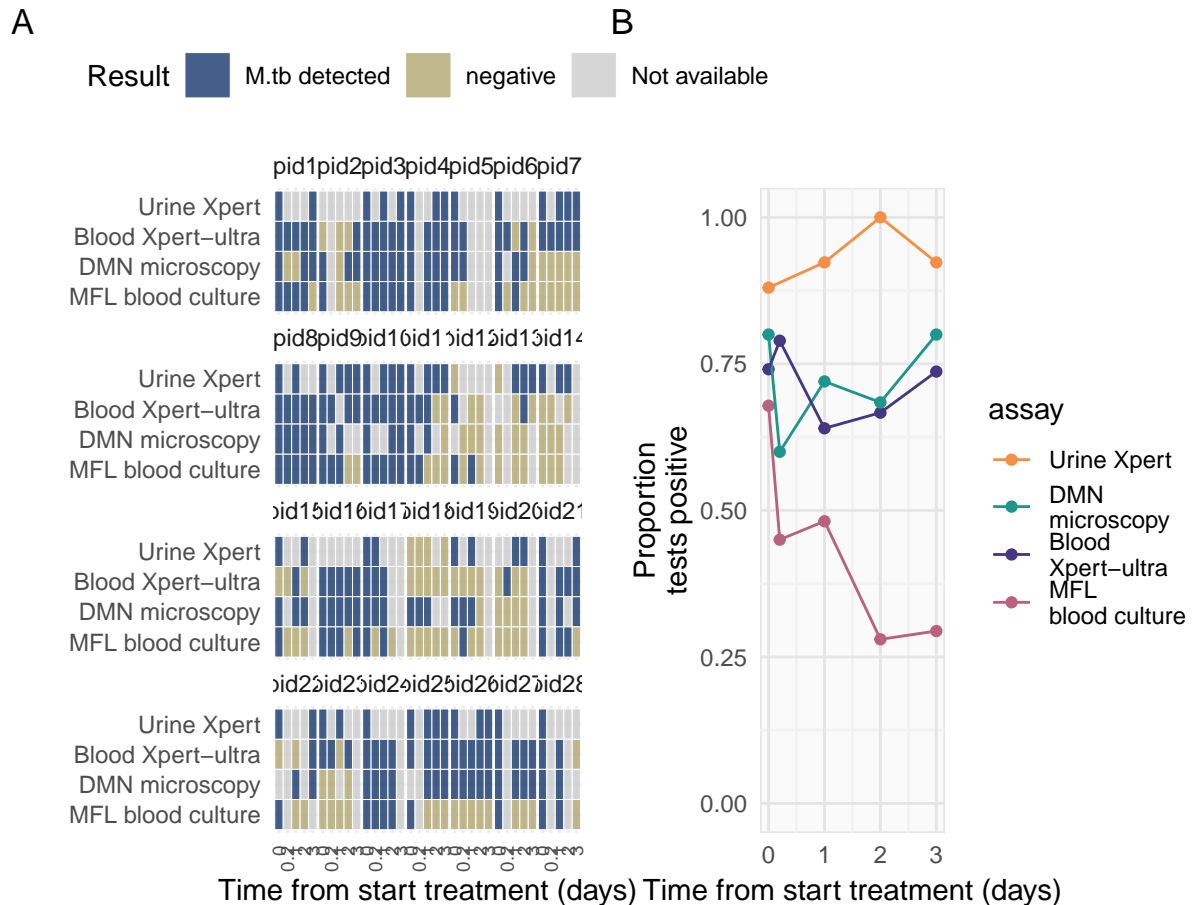


Figure 1: MTB identified by assay and time point

## 5 Raw quantitative headline results

```

# MFL quant results to longer format with preseved sample types pellet v lysate
mfl %>%
  select(study_id, timepoint, type, TTP, MFL) %>%
  mutate(mfl = str_replace(MFL, "contam|lost_viability", replacement = "Not available"),
         mfl = replace_na(mfl, "Not available"),
         mfl = str_replace(mfl, "MTB", replacement = "M.tb detected"),
         mfl = ifelse( is.na(TTP) & MFL == "MTB",

```

```

      "Not available", mfl)) %>% select(-MFL) %>%
pivot_wider(id_cols = c("study_id", "timepoint"),
            names_from = "type",
            values_from = c("mfl", "TTP")) %>%
rename_all(tolower) -> quant_mfl

# bld extract and urine CT values
gxp %>%
  select(study_id, timepoint, sample_type, result,
         matches("rpoB._CT"), IS1081-IS6110_CT) %>%
  filter(
    # these are accidental duplicates which mess up the later pivot wider
    !(study_id=="pid3" & timepoint==72 & result=="MTB NOT DETECTED"),
    !(study_id=="pid18" & timepoint==24 & result=="MTB NOT DETECTED"),
    # not needed for main quant analysis: plasma & buffy samples
    (sample_type == "bldextract" | sample_type == "urine"),
    # 2 pts with (unscheduled, not protocol) TP 96h samples - remove
    timepoint <= 3
  ) %>%
  mutate(result = case_when(
    str_detect(result, "ERROR") ~ "Not available",
    str_detect(result, "MTB DETECTED") ~ "M.tb detected",
    str_detect(result, "MTB NOT DETECTED") ~ "negative")) %>%
  rename(rpoB1=rpoB1_CT, rpoB2=rpoB2_CT, rpoB3=rpoB3_CT,
         rpoB4=rpoB4_CT, IS=IS1081-IS6110_CT) %>%
  pivot_longer(cols = c("rpoB1", "rpoB2", "rpoB3", "rpoB4", "IS"),
              names_to = "probe", values_to = "Ct" ) -> quant_gxp

### DMN
dmn %>%
  mutate(qualitative_result =
    str_replace(qualitative_result, "positive", replacement = "M.tb detected"),
    qualitative_result =
    replace_na(qualitative_result, "Not available")) %>%
  select(study_id, timepoint, dmn_micro = qualitative_result, dmn_ml=DMN_ml) -> quant_dmn
# negatives are 0, missing are NA

quant_gxp %>%
  mutate(study_id = factor(study_id,
                          levels = paste0("pid", 1:28)),
         sample = case_when(
           sample_type=="bldextract" ~ "Blood",
           sample_type=="urine" ~ "Urine"
         )) %>%
  drop_na() %>%
  ggplot(aes(timepoint, Ct)) +
  geom_point(aes(shape=probe, colour=sample)) +
  geom_line(aes(
    group = interaction(study_id, sample, probe),

```



```

    colour = sample)) +
  facet_wrap(~study_id, nrow=7, drop = FALSE) +
  theme_minimal() +
  scale_colour_manual(values = c("#b8627d", "grey")) +
#   theme(legend.position = "bottom",
#         legend.box="vertical", legend.margin=margin()) +
  ggtitle("Xpert-ultra") +
  ylab("Xpert-ultra Ct value") +
  xlab("Time from start treatment (days)") +
  guides(fill=guide_legend(nrow=2,byrow=TRUE)) -> g1

quant_mfl %>%
  select(-mfl_lystate, -mfl_pellet) %>%
  pivot_longer(cols = c("ttp_pellet", "ttp_lystate"),
               names_to = "sample_type", values_to = "TTP") %>%
  mutate(study_id = factor(study_id,
                           levels = paste0("pid", 1:28)),
         sample = str_replace(sample_type, pattern = "ttp_", replacement = "")) %>%
  drop_na() %>%
  ggplot(aes(timepoint, TTP)) +
  geom_point(aes(colour=sample)) +
  geom_line(aes(
    group = interaction(study_id, sample),
    colour = sample)) +
  facet_wrap(~study_id, nrow=7, drop = FALSE) +
  theme_minimal() +
  scale_colour_manual(values = c("#eb8055", "#7e4e90")) +
#   theme(legend.position = "bottom",
#         legend.box="vertical", legend.margin=margin()) +
  ggtitle("Myco/F lytic blood culture") +
  ylab("Time to positivity (TTP)") +
  xlab("Time from start treatment (days)") -> g2

quant_dmn %>%
  mutate(study_id = factor(study_id,
                           levels = paste0("pid", 1:28)),
         Result = case_when(
           dmn_micro=="M.tb detected" ~ "M.tb \ndetected",
           dmn_micro=="negative" ~ "zero \ncount",
           dmn_micro=="non-quantifiable M.tb detected" ~ "zero \ncount"
         )) %>%
  drop_na() %>%
  ggplot(aes(timepoint, log(dmn_ml))) +
  geom_point(aes(colour = Result)) +
  geom_line(colour = "#13306d") +
  facet_wrap(~study_id, nrow=7, drop = FALSE) +
  theme_minimal() +
  scale_colour_manual(values = c("#13306d", "grey")) +
  ggtitle("DMN-tre microscopy count") +
  ylab("Log count per ml") +
  xlab("Time from start treatment (days)") -> g3

```

```
#grid.arrange(g1,g2,g3, nrow=1)
```

g1

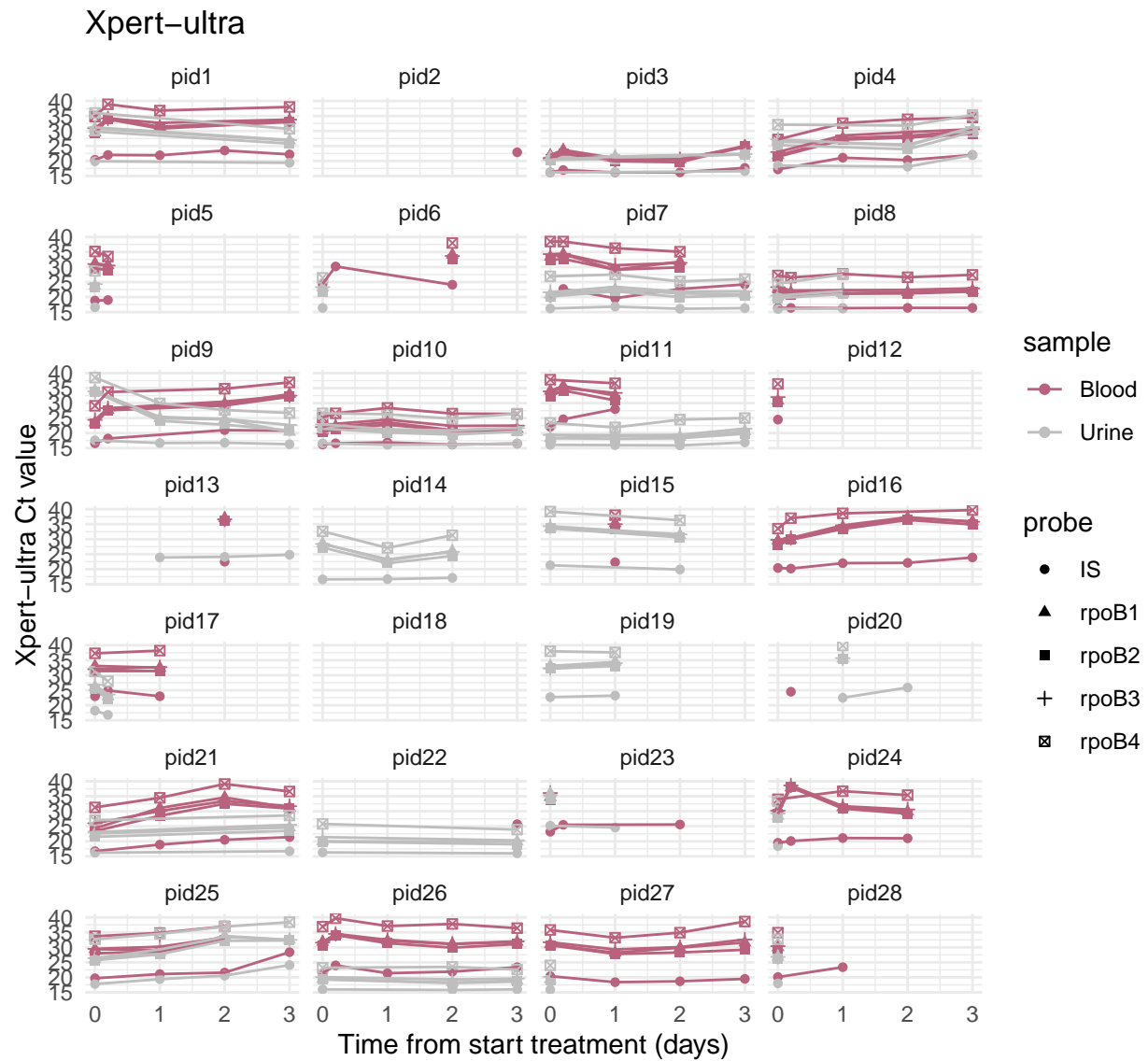


Figure 2: MTBBSI quantification by time and pt: blood Xpert-ultra raw Ct values

g2

g3

## 6 Summary measures from quantitative data

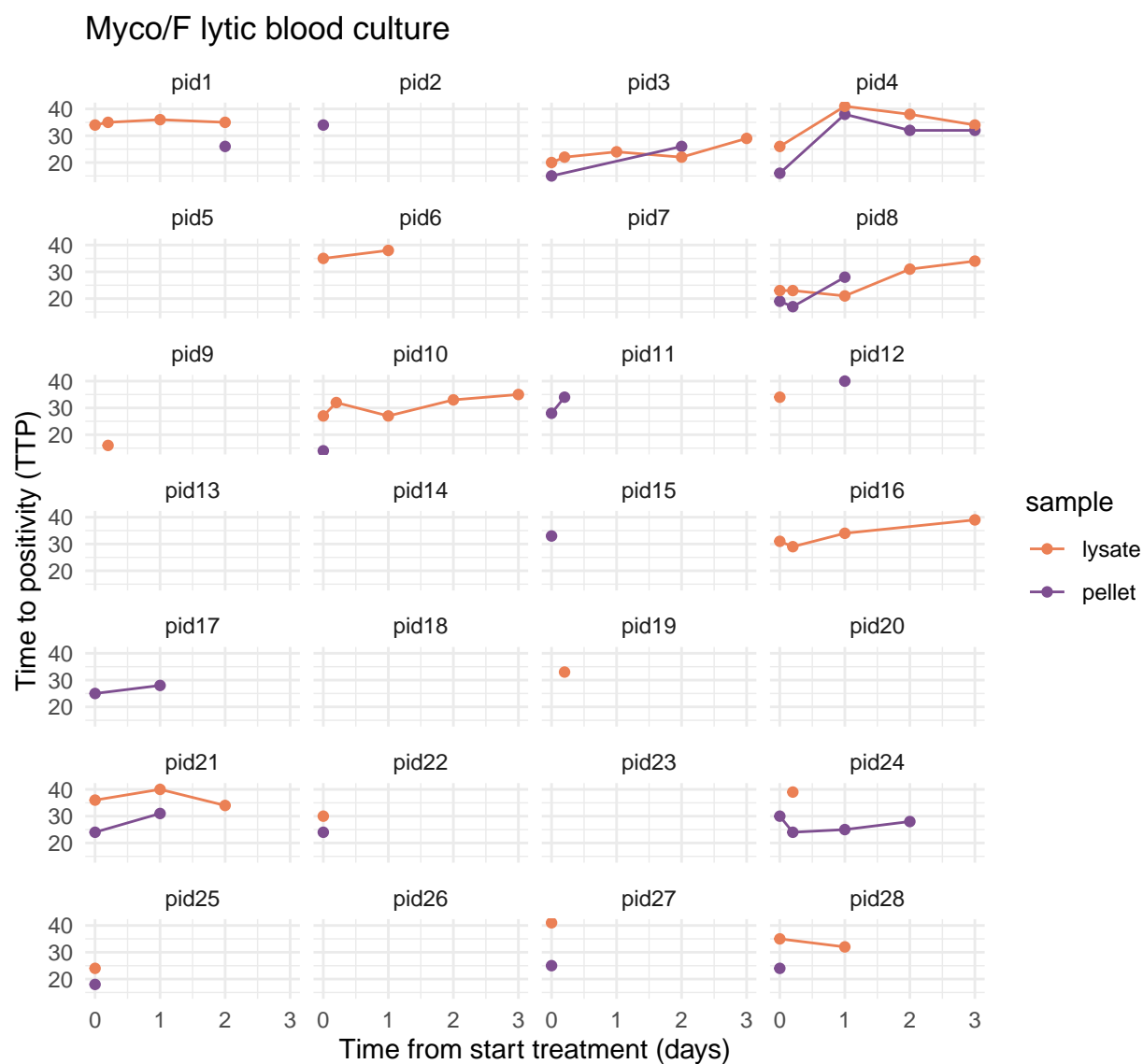


Figure 3: MTBBSI quantification by time and pt: MFL blood culture raw TTP values

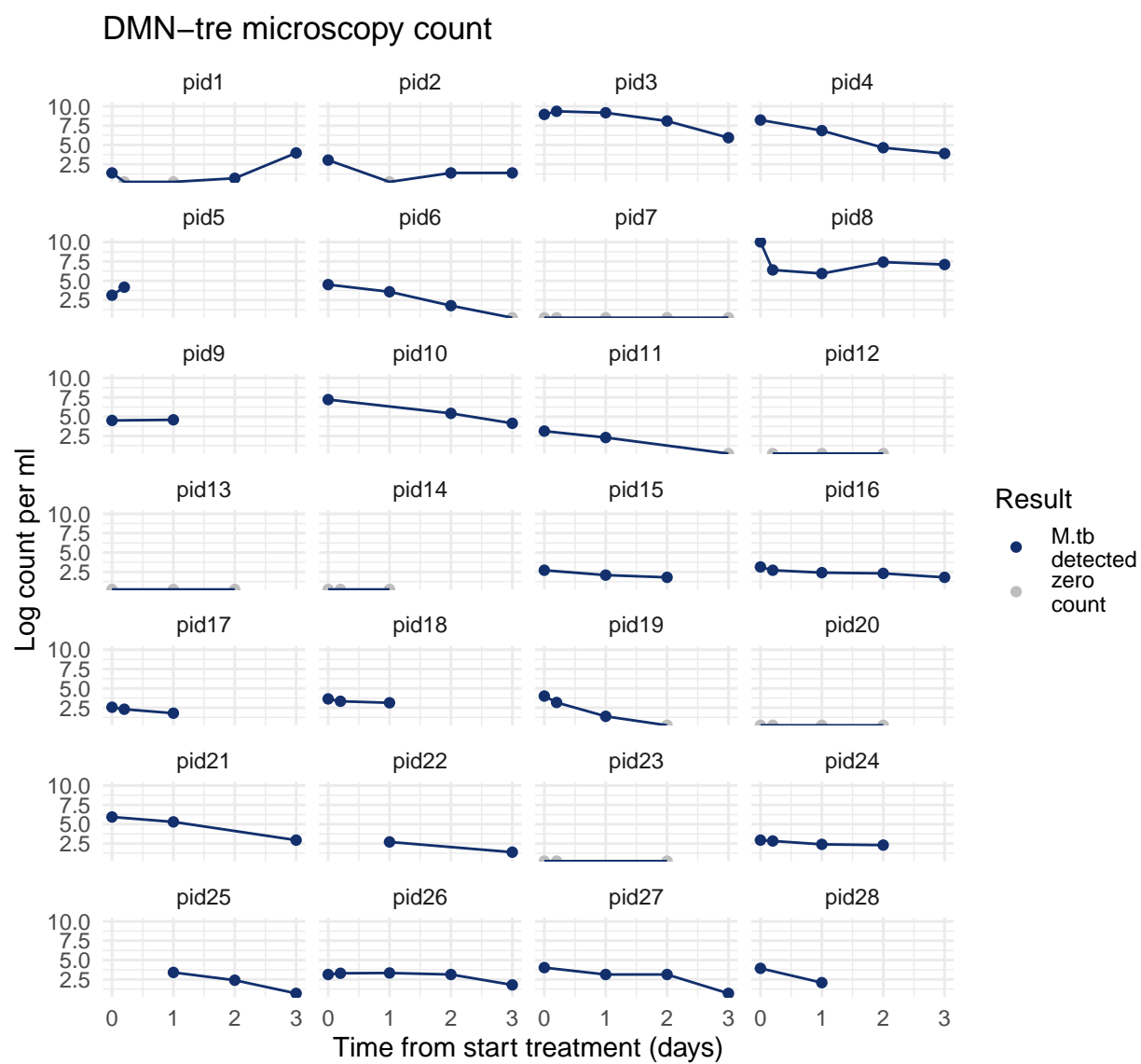


Figure 4: MTBBSI quantification by time and pt: raw log DMN-tre counts per mL

```

# filter and store urine results
quant_gxp %>%
  filter(sample_type == "urine") -> quant_gxp_urine

#### GXP - make summary measure across all probes - based on mean rpoB Ct
quant_gxp %>%
  filter(sample_type == "bldextract") %>%
  select(-sample_type, -result) %>%
  pivot_wider(id_cols = c("study_id", "timepoint"),
    names_from = "probe", values_from = "Ct") %>%
  mutate(
    # min CT from rpob probes as summary measure
    mean_rpob_ct = pmap_dbl(
      list(rpoB1, rpoB2, rpoB3, rpoB4),
      ~mean(c(...), na.rm=TRUE))) -> quant_gxp

# impute the trace positive samples' CT value
quant_gxp$IS_log = log(quant_gxp$IS)
fit <- lm(mean_rpob_ct ~ rcs(IS_log,3),
  # in data excluding a suspect high leverage outlier
  data=quant_gxp[quant_gxp$IS<26,])
ar2 = paste0(
  "RCS model\n aR2=",
  round(summary(fit)$adj.r.squared, 2)
)
quant_gxp %>%
  add_predictions(fit) %>%
  dplyr::mutate(
    pred_mean_rpoB_CT = pred,
    # imputed_ct is mean_rpob_ct, or predicted if trace, or NA if all probes negative
    imputed_ct = case_when(
      !is.na(mean_rpob_ct) ~ mean_rpob_ct,
      is.na(mean_rpob_ct) & !is.na(pred_mean_rpoB_CT) ~ pred_mean_rpoB_CT
    ) -> quant_gxp

quant_gxp %>%
  ggplot(aes(IS, imputed_ct)) +
  geom_point(aes(colour=is.na(mean_rpob_ct)), alpha=0.6) +
  geom_line(aes(IS, pred_mean_rpoB_CT)) +
  theme_minimal() + scale_colour_ptol() +
  theme(legend.position = "none") +
  ylab("Mean rpoB ct\n (imputed values shown)") +
  annotate("text", x = 25, y=22, label=ar2) -> g_imputed
# note non constant residual variance...

### MFL - make summary measure which deals with the difference
# between pellet and lysate, "scaling" pellet ttp to lysate ttp scale
fit <- lm(ttp_lystate ~ ttp_pellet, data=quant_mfl)
quant_mfl %>%
  add_predictions(fit) %>%
  ggplot(aes(ttp_pellet, ttp_lystate)) +

```

```

geom_point() +
geom_smooth() +
geom_abline(slope=1, intercept = 0) +
geom_line(aes(ttp_pellet, pred), colour="red") -> g_scaled_ttp

quant_mfl %>%
  add_predictions(fit) %>%
  dplyr::mutate(
    ttp_pellet = pred
  ) %>%
  select(study_id, timepoint, ttp_lystate, ttp_pellet) %>%
  filter(
    is.na(ttp_lystate) == FALSE | is.na(ttp_pellet) == FALSE) %>%
  mutate(
    ttp = pmap_dbl(
      list(ttp_lystate, ttp_pellet),
      ~mean(c(...), na.rm=TRUE))) -> quant_mfl

quant_dmn %>%
  mutate(dmn_ml_log = log(dmn_ml + 1)) %>%
  select(study_id, timepoint, dmn_ml_log) %>%

  left_join(
    quant_gxp %>%
      select(study_id, timepoint,
             mean_rpoB_Ct = imputed_ct),
    by = c("study_id", "timepoint")
  ) %>%

  left_join(
    quant_mfl %>%
      select(study_id, timepoint, ttp),
    by = c("study_id", "timepoint")
  ) -> quant_df

### urine ct values
quant_gxp_urine %>%
  select(-sample_type, -result) %>%
  pivot_wider(id_cols = c("study_id", "timepoint"),
              names_from = "probe", values_from = "Ct") %>%
  mutate(
    # mean CT from rpob probes as summary measure
    mean_rpob_ct = pmap_dbl(
      list(rpoB1, rpoB2, rpoB3, rpoB4),
      ~mean(c(...), na.rm=TRUE))) -> quant_gxp_urine

# impute the trace positive samples' CT value
quant_gxp_urine$IS_log = log(quant_gxp_urine$IS)
fit <- lm(mean_rpob_ct ~ rcs(IS_log,3),
          data=quant_gxp_urine)

```

```

ar2 = paste0(
  "RCS model\n aR2=",
  round(summary(fit)$adj.r.squared, 2)
)
quant_gxp_urine %>%
  add_predictions(fit) %>%
  dplyr::mutate(
    pred_mean_rpoB_CT = pred,
    # imputed_ct is mean_rprob_ct, or predicted if trace, or NA if all probes negative
    imputed_ct = case_when(
      !is.na(mean_rprob_ct) ~ mean_rprob_ct,
      is.na(mean_rprob_ct) & !is.na(pred_mean_rpoB_CT) ~ pred_mean_rpoB_CT
    ) -> quant_gxp_urine

quant_gxp_urine %>%
  ggplot(aes(IS, imputed_ct)) +
  geom_point(aes(colour=is.na(mean_rprob_ct)), alpha=0.6) +
  geom_line(aes(IS, pred_mean_rpoB_CT)) +
  theme_minimal() + scale_colour_ptol() +
  theme(legend.position = "none") +
  ylab("Mean rpoB ct\n (imputed values shown)") +
  annotate("text", x = 22, y=22, label=ar2) -> g_imputed_urine

quant_gxp_urine %>%
  select(study_id, timepoint,
         urine_mean_rpoB_Ct = imputed_ct) -> quant_gxp_urine

quant_df %>% select(study_id, timepoint,
                   dmn_ml_log,
                   blood_mean_rpoB_Ct = mean_rpoB_Ct,
                   ttp) %>%

left_join(
  quant_gxp_urine,
  by = c("study_id", "timepoint")
) %>%
pivot_longer(3:6, names_to = "assay") %>%
mutate(
  assay = factor(assay,
                 levels = c("ttp",
                           "dmn_ml_log",
                           "blood_mean_rpoB_Ct",
                           "urine_mean_rpoB_Ct"),
                 labels = c("MFL blood culture TTP",
                           "Log DMN-tre bacilli/ml",
                           "Blood Xpert-ultra mean rpoB Ct",
                           "Urine Xpert mean rpoB Ct"))
) %>%
drop_na(value) %>%
ggplot(
  aes(timepoint, value, group=study_id, colour=assay)
) +
geom_quasirandom(alpha=0.5) +

```

```
geom_line() +
facet_wrap(~assay, scales = "free") +
scale_colour_manual(
  values = c( "#b8627dff", "#1F968BFF",
              "#453781FF", "#f68f46ff")) +
theme_dab() +
theme(legend.position = "none") +
xlab("Time from start treatment (days)") +
ylab("") -> g_raw_quant
```

```
g_imputed + g_imputed_urine +
plot_layout(ncol=2, widths=c(3, 3)) +
plot_annotation(tag_levels = "A")
```

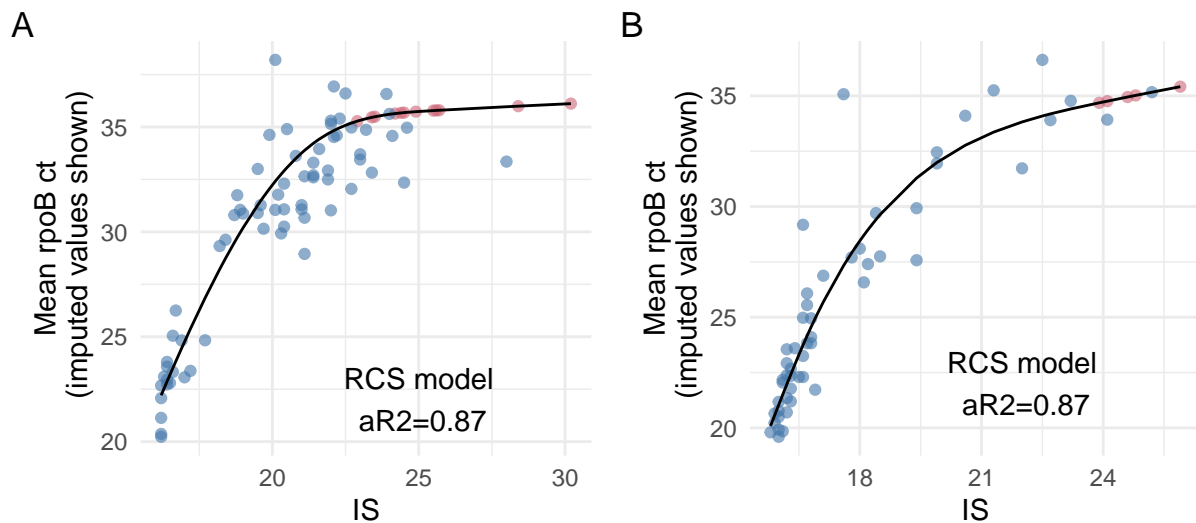


Figure 5: Imputation of trace positive mean rpoB Ct values: A blood; B urine

```
g_raw_quant
```

## 7 Clinical description cohort

With focus on relating clinical phenotype to MTBBSI bacilliary load and outcome.

*# make binary versions of clinical signs to facilitate cluster them*

```
clin %>%
transmute(
  study_id=study_id,
  tachycardia = as.numeric(HR>median(HR)),
  hypotension = as.numeric(sBP<90),
  tachypnea = as.numeric(RR>median(RR)),
  hypoxia = as.numeric(sats <94),
  pyrexia = as.numeric(temp>37.5),
  reduced_GCS = as.numeric(GCS<15),
```



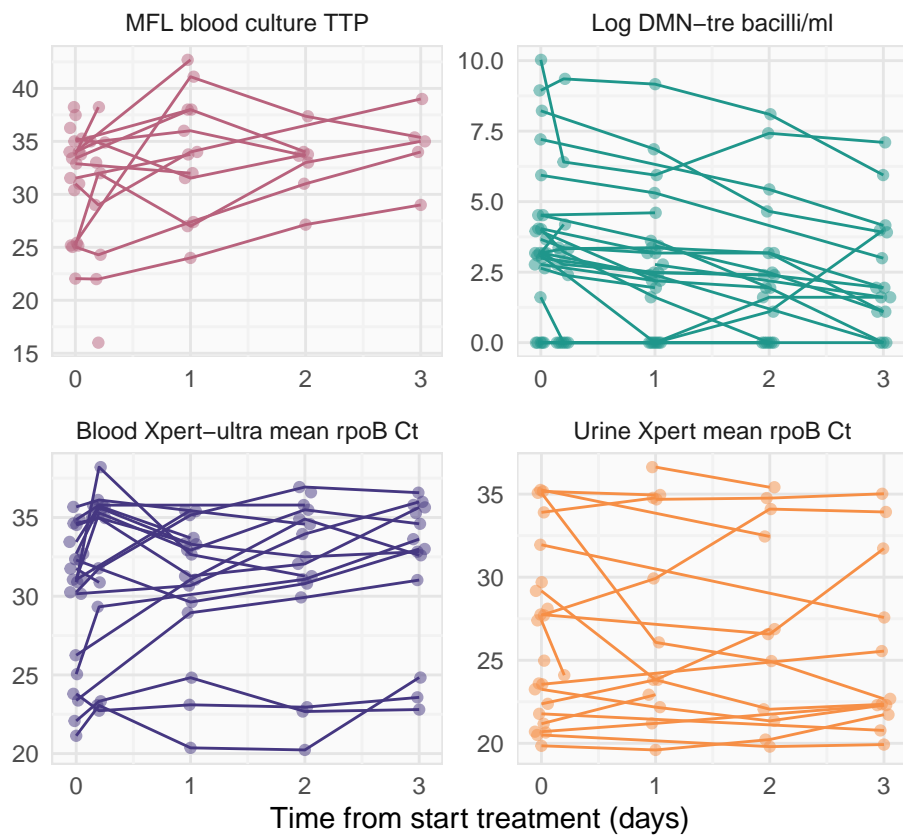


Figure 6: Quant results summarised to a single measure per assay

```

non_mobile = ifelse(walks.unaided==0, 1, 0),
feeding_dependent = ifelse(self.feed==0,1,0),
wasting = wasting,
flushed = flushed,
sweating = sweating,
cool_periph = cool.periph,
delayed_cap_refill = as.numeric(cap.refil>2),
ancil_muscles = ancil.muscles,
nasal_flare = nasal.flare,
not_full_senatnces = ifelse(full.sentences==0,1,0),
# candida = candida,
oedema = oedema,
adenopathy_periph = periph.LN,
doughy_abdo = doughy.abdo,
tender_abdo = tender.abdo,

bilat_infil_CXR = CXR.inf.bilat,
adenopathy_mediastinal = CXR.LN,

# liver_enlarged = liver_enlarged,
liver_echogen = liver_echogen,
# spleen_enlarged = spleen_enlarged,
spleen_hypoecho_lesions = spleen_hypoecho_lesions,
kidney_echogen = as.numeric(ifelse(kidney_echogen=="normal", 0, 1)),
adenopathy_abdo = ifelse(adenopathy=="no", 0, 1)
) -> baseline_signs

# clustering algorithm requires complete cases:
set.seed(1)
imp_blsigns <- mice(baseline_signs, m=1)

##
## iter imp variable
## 1 1 liver_echogen* spleen_hypoecho_lesions* kidney_echogen* adenopathy_abdo*
## 2 1 liver_echogen* spleen_hypoecho_lesions* kidney_echogen* adenopathy_abdo*
## 3 1 liver_echogen* spleen_hypoecho_lesions* kidney_echogen* adenopathy_abdo*
## 4 1 liver_echogen* spleen_hypoecho_lesions* kidney_echogen* adenopathy_abdo*
## 5 1 liver_echogen* spleen_hypoecho_lesions* kidney_echogen* adenopathy_abdo*
## * Please inspect the loggedEvents

baseline_signs <- mice::complete(imp_blsigns)

quant_df %>%
  pivot_wider(id_cols = "study_id",
              names_from = "timepoint",
              values_from = c("dmn_ml_log", "mean_rpoB_Ct", "ttp")) -> quant_df_wide

set.seed(1)
imp_quantdf <- mice(quant_df_wide, m=1)

##
## iter imp variable
## 1 1 dmn_ml_log_0* dmn_ml_log_0.2* dmn_ml_log_1* dmn_ml_log_2* dmn_ml_log_3* mean_rpoB_Ct_0
## 2 1 dmn_ml_log_0* dmn_ml_log_0.2* dmn_ml_log_1* dmn_ml_log_2* dmn_ml_log_3* mean_rpoB_Ct_0

```

```
## 3 1 dmn_ml_log_0* dmn_ml_log_0.2* dmn_ml_log_1* dmn_ml_log_2* dmn_ml_log_3* mean_rpoB_Ct_0
## 4 1 dmn_ml_log_0* dmn_ml_log_0.2* dmn_ml_log_1* dmn_ml_log_2* dmn_ml_log_3* mean_rpoB_Ct_0
## 5 1 dmn_ml_log_0* dmn_ml_log_0.2* dmn_ml_log_1* dmn_ml_log_2* dmn_ml_log_3* mean_rpoB_Ct_0
## * Please inspect the loggedEvents
```

```
quant_df_wide <- mice::complete(imp_quantdf)
quant_df_wide %>%
  select(study_id, ends_with("_0")) -> quant_df_wide

baseline_signs %>%
  left_join(
    quant_df_wide,
    by = "study_id"
  ) %>%
  left_join(
    oc,
    by="study_id"
  ) %>%
  mutate(outcome = ifelse(died==1, "Died", "Survived"))-> baseline_signs

blsdf <- baseline_signs[, 2:27]

# heat map with clustering:

bls_clusters <- data.frame(
  study_id = paste0(
    "pid",
    c(7,25,10,2,5,27,8,16,3,4,
      24,26,21,6,
      22,14,1,9,11,18,28,15,17,13,20,12,19,23)
  ),
  cluster = c(
    rep("A", 10),
    rep("B", 18) ) # added retrospectively so can easier add to figure
)

rownames(blsdf) <- baseline_signs$study_id
rowanno <- data.frame(
  study_id = baseline_signs$study_id,
  outcome = baseline_signs$outcome,
  mean_rpoB_Ct = baseline_signs$mean_rpoB_Ct_0,
  dmn_ml_log = baseline_signs$dmn_ml_log_0,
  MFL_TTP = baseline_signs$ttp_0
) %>%
  left_join(bls_clusters, by = "study_id") %>%
  select(-study_id)

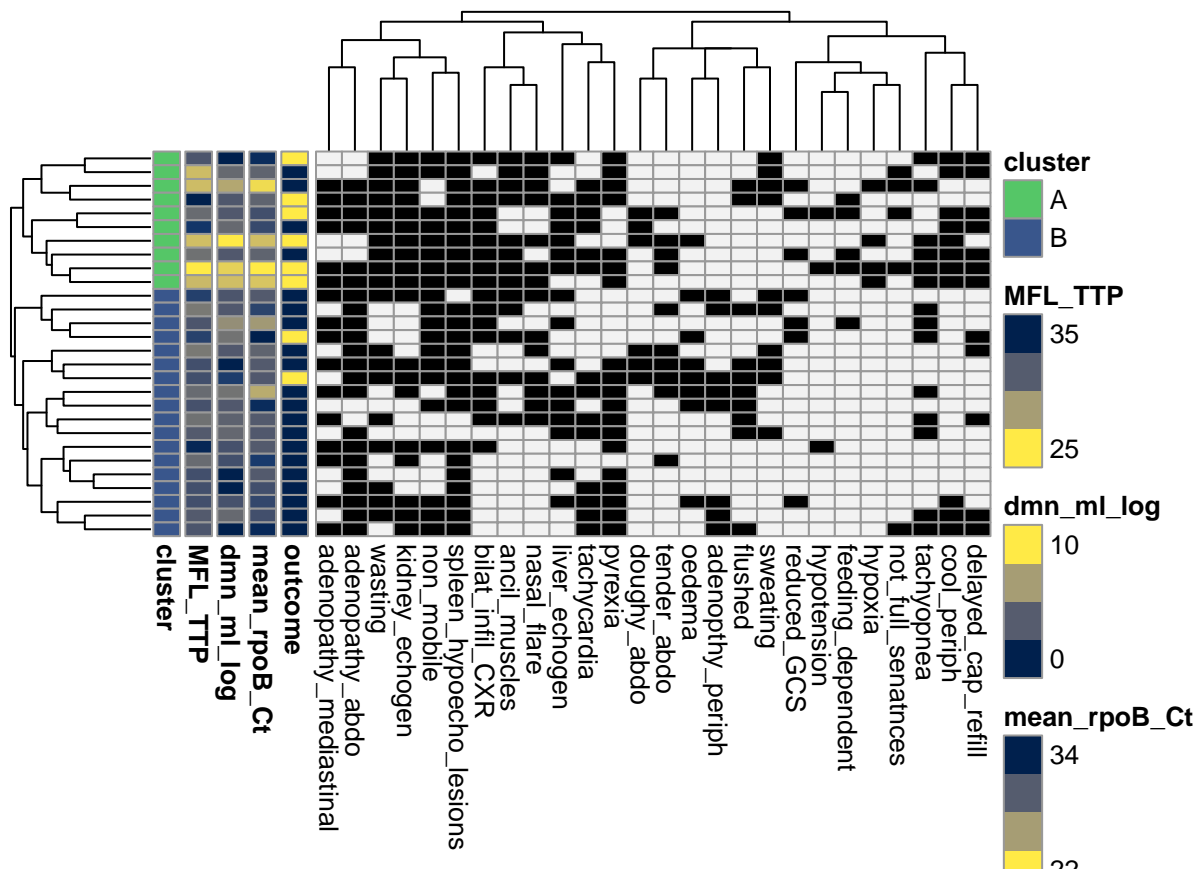
rownames(rowanno) <- baseline_signs$study_id

set.seed(221214)
grid.grabExpr(
  pheatmap(blsdf, legend = FALSE,
    annotation_row = rowanno,
```

```

color=c("grey95", "black"),
annotation_colors = list(
  outcome = c(Died = viridis_pal(option="E")(8)[8],
    Survived = viridis_pal(option="E")(8)[1]),
  mean_rpoB_Ct = viridis::cividis(n=8, alpha = 0.5, direction = -1),
  dmn_ml_log = viridis::cividis(n=8, alpha = 0.5, direction = 1),
  MFL_TTP = viridis::cividis(n=8, alpha = 0.5, direction = -1),
  cluster = c(A="#55C667FF",
    B="#39568CFF")),
show_rownames = FALSE)
) -> bls_hm

```



```

as_ggplot(bls_hm) -> g_bls_hm

# complimentary MCA:

bls <- names(baseline_signs[2:27])
baseline_signs %>%
  mutate(
    across(all_of(bls), as.factor),
    outcome = as.factor(outcome)
  ) %>%
  select(-died) -> baseline_signs
rownames(baseline_signs) <- baseline_signs$study_id
baseline_signs <- select(baseline_signs, -study_id)

```

```

mca1 <- MCA(baseline_signs, quali.sup = 30, quanti.sup = c(27:29))

mca1$ind$coord %>% as_tibble() %>% select(`Dim 1`, `Dim 2`) %>%
  mutate(
    study_id = rownames(baseline_signs),
    mean_rpoB_Ct = baseline_signs$mean_rpoB_Ct_0,
    dmnl_log = baseline_signs$dmnl_log_0,
    MFL_TTP = baseline_signs$ttp_0,
    outcome = baseline_signs$outcome
  ) %>%
  left_join(bls_clusters, by="study_id") -> mca1_coord

set.seed(221214)
mca1_coord %>%
  ggplot(aes(`Dim 1`, `Dim 2`, fill=outcome, shape=cluster, colour=cluster)) +
  xlim(-1, 1) + ylim(-1, 1) +
  geom_vline(xintercept = 0, colour="grey") +
  geom_hline(yintercept = 0, colour="grey") +
  geom_point(size=3, alpha=0.5,
    position = position_jitter(width=0.05, height=0.05)) +
  theme_dab() +
  scale_colour_manual(values = c("#55C667FF", "#39568CFF")) +
  scale_shape_manual(values = c(24, 21)) +
  scale_fill_manual(
    values = c(
      viridis_pal(option="E")(8)[8],
      viridis_pal(option="E")(8)[1]
    )
  ) +
  guides(fill = guide_legend(override.aes=list(shape=22))) +
  xlab("MCA Clinical signs Dim 1") +
  ylab("MCA clinical signs Dim 2") -> g_mca1_pts

mca1$quanti.sup$coord %>%
  as_tibble() %>%
  select(`Dim 1`, `Dim 2`) %>%
  bind_rows(
    mca1$quali.sup$coord %>%
      as_tibble() %>%
      select(`Dim 1`, `Dim 2`)
  ) %>%
  mutate(
    var = c("log\ndMN/ml", "Mean\nrpoB Ct", "MFL\nTTP", "Died", "Survived"),
    x = c(-0.3, -0.25, -0.9, 0.55, -0.35),
    y = c(0.65, -0.3, -0.40, 0.85, -0.2),
    sup_var_type = c("bacilli_burden", "bacilli_burden", "bacilli_burden",
      "outcome", "outcome") -> mca1_sup_loading

ggplot(mca1_sup_loading %>% filter(var!="Survived"),
  aes(colour=sup_var_type)) +
  xlim(-1, 1) + ylim(-1, 1) +
  geom_vline(xintercept = 0, colour="grey") +

```

```

geom_hline(yintercept = 0, colour="grey") +
geom_text(
  mapping = aes(x = x,
                y = y,
                label = var),
  hjust=0, vjust=1, size=3.5) +
geom_segment(
  mapping = aes(x = 0, xend = `Dim 1`,
                y = 0, yend = `Dim 2`),
  size=1,
  arrow = arrow(length = unit(0.3, "cm"))) +
theme_minimal() +
xlab("Loadings on MCA Dim 1") +
ylab("Loadings on MCA Dim 2") +
scale_colour_manual(values = c("#042333ff", "#a65c85ff")) +
theme(legend.position = "none") -> g_mca1_loadings

mca1_projn <- data.frame(
  xend = c(
    cor(mca1_coord$mean_rpoB_Ct, mca1_coord$`Dim 1`),
    cor(mca1_coord$dmn_ml_log, mca1_coord$`Dim 1`),
    cor(mca1_coord$MFL_TTP, mca1_coord$`Dim 1`)),
  yend = c(
    cor(mca1_coord$mean_rpoB_Ct, mca1_coord$`Dim 2`),
    cor(mca1_coord$dmn_ml_log, mca1_coord$`Dim 2`),
    cor(mca1_coord$MFL_TTP, mca1_coord$`Dim 2`)),
  label = c(
    "Mean\nrpoB Ct", "log\nDMN-tre", "MFL\nTTP"),
  x = c(-0.35, 0.25, -0.9),
  y = c(-0.4, 0.8, -0.1)
)

mca1_projn %>%
  ggplot(aes(colour=label)) +
  xlim(-1,1) + ylim(-1,1) +
  geom_vline(xintercept = 0, colour="grey") +
  geom_hline(yintercept = 0, colour="grey") +
  geom_segment(
    mapping = aes(x = 0, xend = xend,
                  y = 0, yend = yend),
    size=1,
    arrow = arrow(length = unit(0.2, "cm"))) +
  geom_text(
    mapping = aes(x = x,
                  y = y,
                  label = label),
    hjust=0, vjust=1, size=3.5) +
  theme_dab() + theme(legend.position = "none") +
  scale_colour_manual(values = c("#1F968BFF", "#453781FF", "#b8627dff")) +
  xlab("Correlation MCA Dim 1") +
  ylab("Correlation MCA Dim 2") -> g_mca1_cor

```

g\_bls\_hm

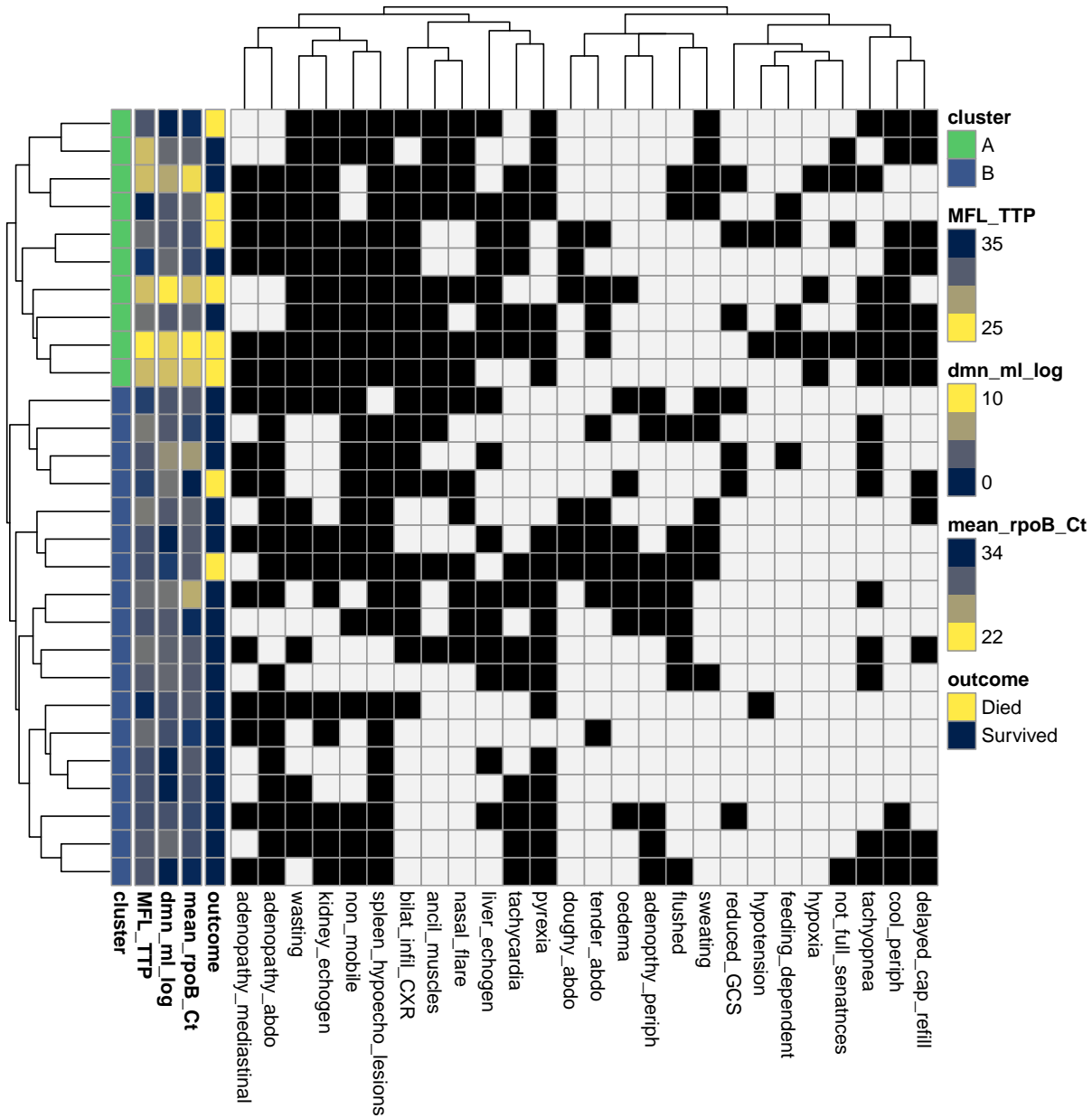


Figure 7: Baseline clinical signs dimension reduction: hclust heatmap

```
g_mca1_cor + g_mca1_loadings + g_mca1_pts +
  plot_annotation(tag_levels = "A") +
  plot_layout(nrow=1)
```

#### bloods PCA

```
blds <- read_csv("baseline_blds.csv")
```

```
blds %>%
```

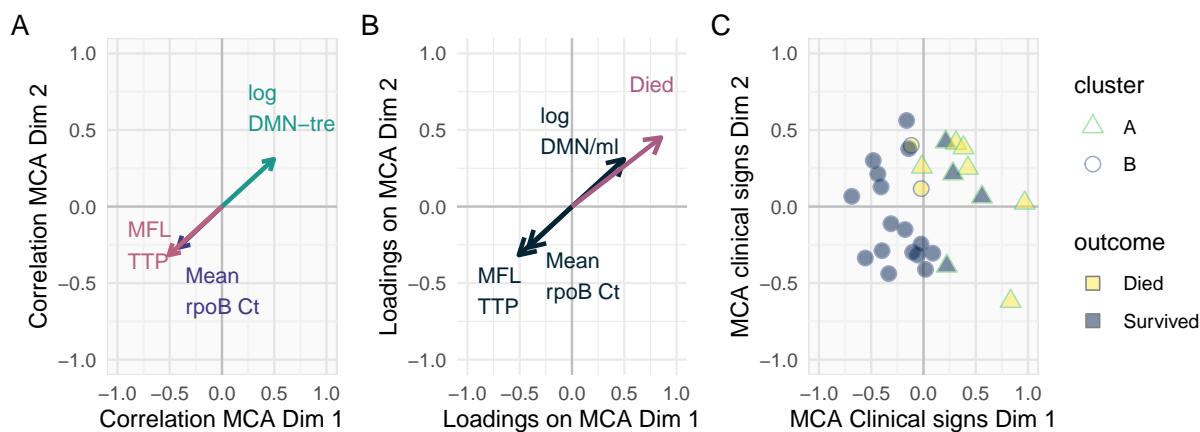


Figure 8: Baseline clinical signs dimension reduction: MCA

```
select(-X1) %>%
left_join(pids, by="study_id") %>%
mutate(study_id = pid) -> blds

blds %>%
  select(study_id,
         cd4, wcc, platelets, MPV,
         Hb, Hct, MCH, MCHC, MCV, rcc, RDW,
         Na, K, urea,
         AST, ALT, ALP = `alk phos`, GGT, Tbil, Cbil) %>%
  left_join(
    clin %>%
      select(study_id,
             pH, HCO3, BE, lactate, pO2, pCO2, glucose, creat),
    by = "study_id"
  ) -> blds

blds %>%
  select(study_id,
         cd4, Hb, platelets, MPV, lactate, AST,
         wcc, Na, urea, creat, glucose, MCHC) %>%
  left_join(
    mca1_coord %>% select(study_id, outcome),
    by = "study_id"
  ) %>%
  pivot_longer(2:13, names_to = "variable") %>%
  ggplot(
    aes(value, fill=outcome)
  ) +
  geom_density(alpha=0.5, adjust=1.8) +
  facet_wrap(~variable, scales = "free", nrow=4) +
  theme_dab() +
  scale_fill_manual(
    values = c(
```



```

    viridis_pal(option="E")(8)[8],
    viridis_pal(option="E")(8)[1]
  )) +
  xlab("") +
  theme(legend.position = "bottom",
        panel.spacing = unit(0, "lines"),
        axis.text.x = element_text(size=8),
        axis.text.y = element_text(size=8)) -> g_blds_dist

```

```

blds %>%
  transmute(
    study_id = study_id,
    cd4 = log(cd4+1),
    wcc = log(wcc),
    platelets = log(platelets),
    MPV = log(MPV),
    Hb = Hb,
    Hct = Hct,
    MCH = MCH,
    MCHC = MCHC,
    MCV = MCV,
    rcc = rcc,
    RDW = RDW,
    wcc = sqrt(wcc),
    Na = Na,
    K = K,
    urea = log(urea),
    creat = log(creat),
    AST = log(AST),
    ALT = log(ALT),
    ALP = log(ALP),
    GGT = log(GGT),
    u_bil = log(Tbil - Cbil),
    c_bil = log(Cbil),
    pH = pH,
    HCO3 = HCO3,
    BE = BE,
    lactate = log(lactate),
    pCO2 = pCO2,
    glucose = log(glucose)
  ) %>%
  mutate_at(c(2:10), funs(scale(.))) -> blds_scaled

```

```

set.seed(1)
imp_blds <- mice(blds_scaled, m=1)

```

```

##
## iter imp variable
## 1 1 MPV* RDW* Na* K* urea* AST* ALT* ALP* GGT* u_bil* c_bil*
## 2 1 MPV* RDW* Na* K* urea* AST* ALT* ALP* GGT* u_bil* c_bil*
## 3 1 MPV* RDW* Na* K* urea* AST* ALT* ALP* GGT* u_bil* c_bil*
## 4 1 MPV* RDW* Na* K* urea* AST* ALT* ALP* GGT* u_bil* c_bil*
## 5 1 MPV* RDW* Na* K* urea* AST* ALT* ALP* GGT* u_bil* c_bil*

```

```

## * Please inspect the loggedEvents
blds_scaled <- mice::complete(imp_blds)

blds_scaled$AST_resid <- resid(lm(AST ~ ALT, data=blds_scaled))
blds_scaled$wcc_resid <- resid(lm(wcc ~ cd4, data=blds_scaled))

# PCA*
rownames(blds_scaled) <- blds_scaled$study_id
pca1 <- principal(
  blds_scaled %>%
    select(cd4, Hb, platelets, MPV, lactate, AST_resid,
           wcc, Na, urea, creat, glucose, MCHC),
  nfactors = 2, rotate = "varimax")

pca1_loadings <- data.frame(pc1 = pca1$loadings[,1],
                           pc2 = pca1$loadings[,2],
                           variable = names(pca1$loadings[,1]),
                           adverse = c("l", "l", "l", "h", "h", "h",
                                         "u", "u", "h", "h", "u", "u"))

pca1_loadings %>%
  ggplot(
    aes(pc1, pc2, colour=adverse)) +
  geom_vline(xintercept = 0) +
  geom_hline(yintercept = 0) +
  geom_point(size=3, alpha=0.7) +
  geom_label_repel(
    aes(pc1, pc2,
        label = variable),
    box.padding = 0.35, point.padding = 0.5,
    segment.color = 'grey50') +
  theme_dab() +
  scale_color_manual(values =
    viridis(12, option="E")[c(9,3,6)]) +
  theme(legend.position = "none") +
  xlab("Blood results PC1 variable loadings") +
  ylab("Blood results PC2 variable loadings") -> g_pca1_loadings

pca1_coord <- pca1$scores %>%
  as_tibble() %>%
  transmute(
    study_id = rownames(pca1$scores),
    pc1 = RC1,
    pc2 = RC2
  ) %>%
  left_join(
    mca1_coord %>% select(-`Dim 1`, -`Dim 2`, -cluster),
    by = "study_id"
  )

```

```

pca1_coord %>%
  ggplot(
    aes(pc1, pc2, fill=outcome)
  ) +
  geom_vline(xintercept = 0, colour="grey") +
  geom_hline(yintercept = 0, colour="grey") +
  geom_point(size=3.5, alpha=0.5, shape=21) +
  scale_fill_manual(
    values = c(
      viridis_pal(option="E")(8)[8],
      viridis_pal(option="E")(8)[1]
    )
  ) +
  theme_dab() +
  xlab("Blood results PC1") +
  ylab("Blood results PC2") -> g_pca1_pts

pca1_projn <- data.frame(
  xend = c(
    cor(pca1_coord$mean_rpoB_Ct, pca1_coord$pc1),
    cor(pca1_coord$dmn_ml_log, pca1_coord$pc1),
    cor(pca1_coord$MFL_TTP, pca1_coord$pc1)),
  yend = c(
    cor(pca1_coord$mean_rpoB_Ct, pca1_coord$pc2),
    cor(pca1_coord$dmn_ml_log, pca1_coord$pc2),
    cor(pca1_coord$MFL_TTP, pca1_coord$pc2)),
  label = c(
    "Mean\nrpoB Ct", "log\nDMN-tre", "MFL\nTTP"),
  x = c(-0.4, 0.3, -0.8),
  y = c(-0.6, 0.9, -0.1)
)

pca1_projn %>%
  ggplot(aes(colour=label)) +
  xlim(-1,1) + ylim(-1,1) +
  geom_vline(xintercept = 0, colour="grey") +
  geom_hline(yintercept = 0, colour="grey") +
  geom_segment(
    mapping = aes(x = 0, xend = xend,
                  y = 0, yend = yend),
    size=1,
    arrow = arrow(length = unit(0.2, "cm"))) +
  geom_text(
    mapping = aes(x = x,
                  y = y,
                  label = label),
    hjust=0, vjust=1, size=3.5) +
  theme_dab() + theme(legend.position = "none") +
  scale_colour_manual(values = c("#1F968BFF", "#453781FF", "#b8627dff")) +
  xlab("Correlation with PC1") +
  ylab("Correlation with PC2") -> g_pca1_cor

#####

```

```

#fit_pc1_ct <- brm(pc1 ~ mean_rpoB_Ct, data = pca1_coord)
#fit_pc1_ct_R2 <- bayes_R2(fit_pc1_ct, summary = TRUE)
#fit_pc1_ct_prob_neg <- round((sum(posterior_samples(fit_pc1_ct)$b_mean_rpoB_Ct < 0))/4000, 3)

#fit_pc1_dmn <- brm(pc1 ~ dmn_ml_log, data = pca1_coord)
#fit_pc1_dmn_R2 <- bayes_R2(fit_pc1_dmn, summary = TRUE)
#fit_pc1_dmn_prob_pos <- round((sum(posterior_samples(fit_pc1_dmn)$b_dmn_ml_log > 0))/4000, 3)

#fit_pc1_ttp <- brm(pc1 ~ MFL_TTP, data = pca1_coord)
#fit_pc1_ttp_R2 <- bayes_R2(fit_pc1_ttp, summary = TRUE)
#fit_pc1_ttp_prob_neg <- round((sum(posterior_samples(fit_pc1_ttp)$b_MFL_TTP < 0))/4000, 3)

#fit_pc2_ct <- brm(pc2 ~ mean_rpoB_Ct, data = pca1_coord)
#fit_pc2_ct_R2 <- bayes_R2(fit_pc2_ct, summary = TRUE)
#fit_pc2_ct_prob_neg <- round((sum(posterior_samples(fit_pc2_ct)$b_mean_rpoB_Ct < 0))/4000, 3)

#fit_pc2_dmn <- brm(pc2 ~ dmn_ml_log, data = pca1_coord)
#fit_pc2_dmn_R2 <- bayes_R2(fit_pc2_dmn, summary = TRUE)
#fit_pc2_dmn_prob_pos <- round((sum(posterior_samples(fit_pc2_dmn)$b_dmn_ml_log > 0))/4000, 3)

#fit_pc2_ttp <- brm(pc2 ~ MFL_TTP, data = pca1_coord)
#fit_pc2_ttp_R2 <- bayes_R2(fit_pc2_ttp, summary = TRUE)
#fit_pc2_ttp_prob_neg <- round((sum(posterior_samples(fit_pc2_ttp)$b_MFL_TTP < 0))/4000, 3)

#c(fit_pc1_dmn_R2[1], fit_pc1_ttp_R2[1], fit_pc1_ct_R2[1])
#c(fit_pc1_ct_prob_neg, fit_pc1_dmn_prob_pos, fit_pc1_ttp_prob_neg)
#c(fit_pc2_dmn_R2[1], fit_pc2_ttp_R2[1], fit_pc2_ct_R2[1])
#c(fit_pc2_ct_prob_neg, fit_pc2_dmn_prob_pos, fit_pc2_ttp_prob_neg)

#fit_pc1_outcome <- brm(pc1 ~ outcome, data = pca1_coord)
#fit_pc2_outcome <- brm(pc2 ~ outcome, data = pca1_coord)

#####

g_blds_dist

g_pca1_loadings + g_pca1_cor + g_pca1_pts +
  plot_annotation(tag_levels = "A") +
  plot_layout(nrow=1)

```

## 8 Agreement between methods

```

quant_df %>% select(study_id, timepoint,
                    dmn_ml_log,
                    blood_mean_rpoB_Ct = mean_rpoB_Ct,
                    ttp) %>%

left_join(
  quant_gxp_urine,
  by = c("study_id", "timepoint")
) -> quant_df_u

```

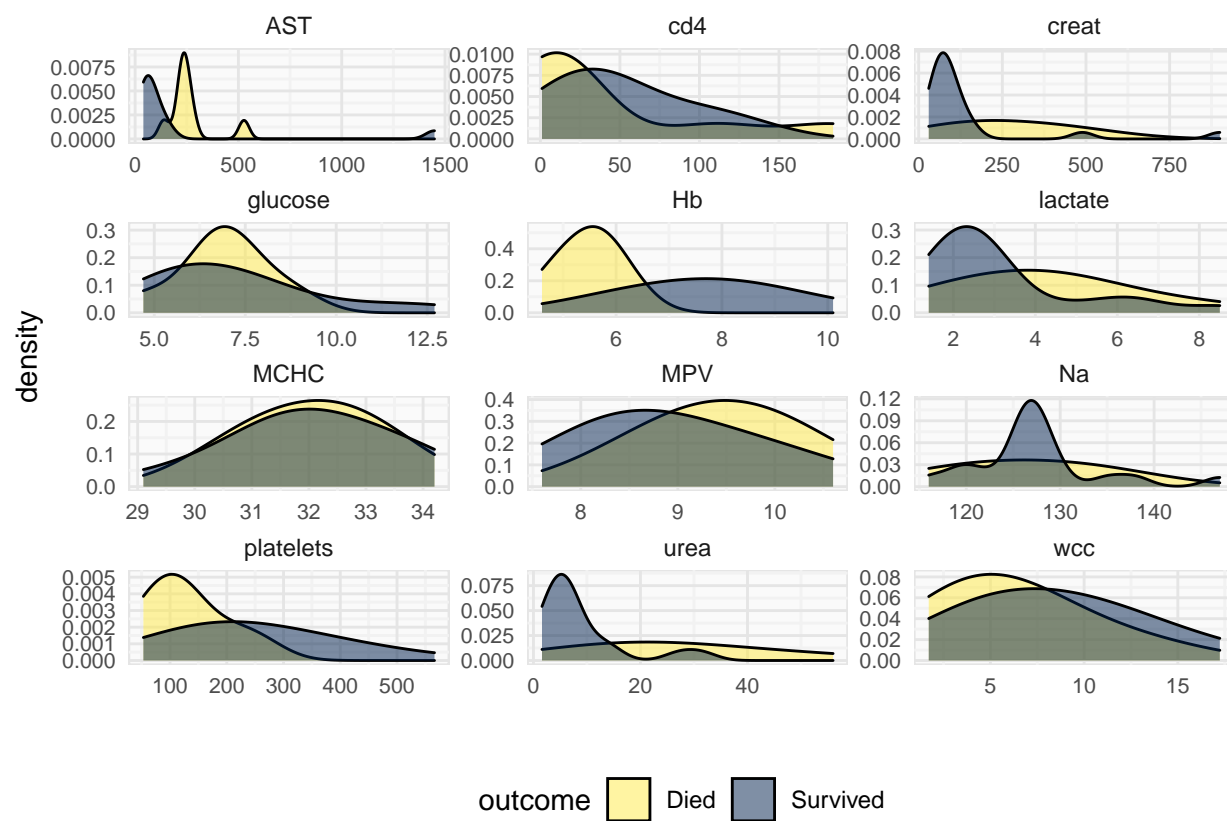


Figure 9: Patient blood results

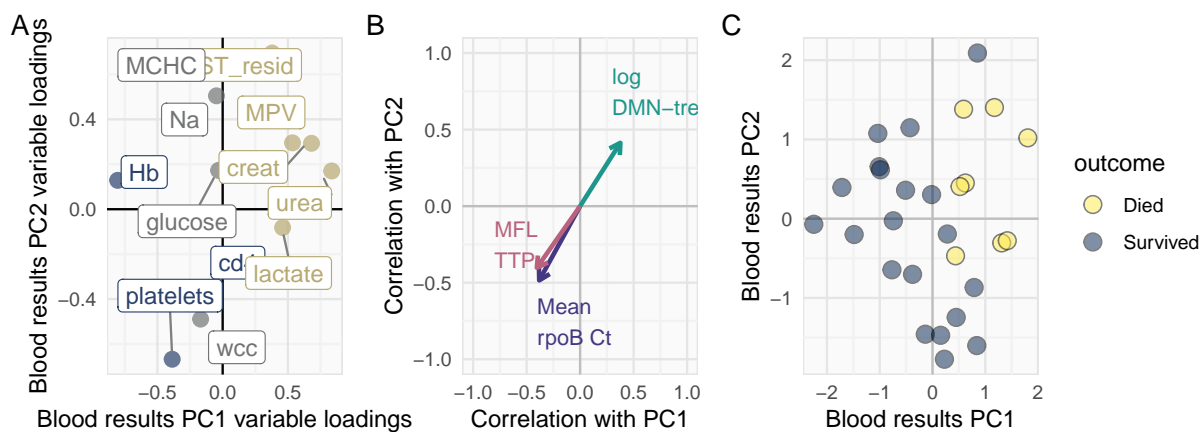


Figure 10: Patient blood results dimension reduction

```

fit_ttp_ct <- brm(ttp ~ ns(blood_mean_rpoB_Ct, df=3), data = quant_df_u)
fit_dmn_ttp <- brm(dmn_ml_log ~ ns(ttp, df=3), data = quant_df_u)
fit_ct_dmn <- brm(blood_mean_rpoB_Ct ~ ns(dmn_ml_log, df=3), data = quant_df_u)
fit_ct_uct <- brm(blood_mean_rpoB_Ct ~ ns(urine_mean_rpoB_Ct, df=3), data = quant_df_u)
fit_dmn_uct <- brm(dmn_ml_log ~ ns(urine_mean_rpoB_Ct, df=3), data = quant_df_u)
fit_ttp_uct <- brm(ttp ~ ns(urine_mean_rpoB_Ct, df=3), data = quant_df_u)

r2_ttp_ct <- as.numeric(round(bayes_R2(fit_ttp_ct), 2))
r2_dmn_ttp <- as.numeric(round(bayes_R2(fit_dmn_ttp), 2))
r2_ct_dmn <- as.numeric(round(bayes_R2(fit_ct_dmn), 2))
r2_ct_uct <- as.numeric(round(bayes_R2(fit_ct_uct), 2))

fit_ttp_ct_df <- fitted(fit_ttp_ct,
  newdata = data.frame(
    blood_mean_rpoB_Ct = seq(20,37, by = 0.2))) %>%
  as_tibble() %>%
  bind_cols(
    data.frame(
      blood_mean_rpoB_Ct = seq(20,37, by = 0.2))) %>%
  select(blood_mean_rpoB_Ct, ttp_hat = Estimate, Q2.5, Q97.5)

fit_dmn_ttp_df <- fitted(fit_dmn_ttp,
  newdata = data.frame(ttp = seq(20,42, by = 0.2))) %>%
  as_tibble() %>%
  bind_cols(
    data.frame(ttp = seq(20,42, by = 0.2))) %>%
  select(ttp, dmn_ml_log_hat = Estimate, Q2.5, Q97.5)

fit_ct_dmn_df <- fitted(fit_ct_dmn,
  newdata = data.frame(dmn_ml_log = seq(0,10, by = 0.2))) %>%
  as_tibble() %>%
  bind_cols(
    data.frame(dmn_ml_log = seq(0,10, by = 0.2))) %>%
  select(dmn_ml_log, blood_mean_rpoB_Ct_hat = Estimate, Q2.5, Q97.5)

fit_ct_uct_df <- fitted(fit_ct_uct,
  newdata = data.frame(
    urine_mean_rpoB_Ct = seq(19,37, by = 0.2))) %>%
  as_tibble() %>%
  bind_cols(
    data.frame(urine_mean_rpoB_Ct = seq(19,37, by = 0.2))) %>%
  select(urine_mean_rpoB_Ct, blood_mean_rpoB_Ct_hat = Estimate, Q2.5, Q97.5)

quant_df_u %>%
  ggplot(aes(dmn_ml_log, blood_mean_rpoB_Ct)) +
  geom_point(alpha=0.4, size=1.5, colour="#9E9677FF") +
  geom_line(data = fit_ct_dmn_df,
    aes(dmn_ml_log, blood_mean_rpoB_Ct_hat)) +
  geom_ribbon(
    data = fit_ct_dmn_df,
    mapping =
      aes(x=dmn_ml_log, ymin = Q2.5, ymax=Q97.5),

```

```

    inherit.aes = FALSE,
    alpha=0.3) +
  theme_minimal() +
  theme(legend.position = "none") +
  geom_text(x = 10, y = 38, parse=TRUE,
            hjust = 1, vjust = 1, size=3.5,
            label =
              paste0(
                "R^2==", r2_ct_dmn[1])) +
  geom_text(x = 10, y = 35, parse = TRUE,
            hjust = 1, vjust = 1, size=3.5,
            label =
              paste0(
                "(", r2_ct_dmn[3], "-", r2_ct_dmn[4], ")")
            )) +
  xlab("Log DMN-tre bacilli/ml") + ylab("Blood mean rpoB Ct") -> g3
ggExtra::ggMarginal(g3, margins = "x",
                    type = "density",
                    fill="#1F968BFF",
                    alpha=0.25) -> g3

quant_df_u %>%
  ggplot(aes(ttp, dmn_ml_log)) +
  geom_point(alpha=0.4, size=1.5, colour="#9E9677FF") +
  geom_line(data = fit_dmn_ttp_df,
            aes(ttp, dmn_ml_log_hat)) +
  geom_ribbon(
    data = fit_dmn_ttp_df,
    mapping =
      aes(x=ttp, ymin = Q2.5, ymax=Q97.5),
    inherit.aes = FALSE,
    alpha=0.3) +
  theme_minimal() +
  theme(legend.position = "none") +
  geom_text(x = 42, y = 15, parse = TRUE,
            hjust = 1, vjust = 1, size=3.5,
            label =
              paste0(
                "R^2==", r2_dmn_ttp[1])) +
  geom_text(x = 42, y = 12.5, parse = TRUE,
            hjust = 1, vjust = 1, size=3.5,
            label =
              paste0(
                "(", r2_dmn_ttp[3], "-", r2_dmn_ttp[4], ")")) +
  ylab("Log DMN-tre bacilli/ml") + xlab("MFL blood culture TTP") -> g1
ggExtra::ggMarginal(g1, margins = "x",
                    type = "density",
                    fill="#b8627dff", alpha=0.25) -> g1

quant_df_u %>%

```

```

ggplot(aes(blood_mean_rpoB_Ct, ttp)) +
  geom_point(alpha=0.4, size=1.5, colour="#9E9677FF") +
  geom_line(data = fit_ttp_ct_df,
            aes(blood_mean_rpoB_Ct, ttp_hat)) +
  geom_ribbon(
    data = fit_ttp_ct_df,
    mapping =
      aes(x=blood_mean_rpoB_Ct, ymin = Q2.5, ymax=Q97.5),
    inherit.aes = FALSE,
    alpha=0.3) +
  theme_minimal() +
  geom_text(x = 37.5, y = 21, parse = TRUE,
            hjust = 1, vjust = 0, size=3.5,
            label =
              paste0(
                "R^2==", r2_ttp_ct[1])) +
  geom_text(x = 37.5, y = 17, parse = TRUE,
            hjust = 1, vjust = 0, size=3.5,
            label =
              paste0(
                "(", r2_ttp_ct[3], "-", r2_ttp_ct[4], ")")
            )) +
  xlab("Blood mean rpoB Ct") + ylab("MFL blood culture TTP") -> g2
ggExtra::ggMarginal(g2, margins = "x",
                    type = "density",
                    fill="#453781FF", alpha=0.25) -> g2

quant_df_u %>%
  ggplot(aes(urine_mean_rpoB_Ct, blood_mean_rpoB_Ct)) +
  geom_point(alpha=0.4, size=1.5, colour="#9E9677FF") +
  geom_line(data = fit_ct_uct_df,
            aes(urine_mean_rpoB_Ct, blood_mean_rpoB_Ct_hat)) +
  geom_ribbon(
    data = fit_ct_uct_df,
    mapping =
      aes(x=urine_mean_rpoB_Ct, ymin = Q2.5, ymax=Q97.5),
    inherit.aes = FALSE,
    alpha=0.3) +
  theme_minimal() +
  theme(legend.position = "none") +
  geom_text(x = 37, y = 25, parse=TRUE,
            hjust = 1, vjust = 1, size=3.5,
            label =
              paste0(
                "R^2==", r2_ct_uct[1])) +
  geom_text(x = 37, y = 22, parse = TRUE,
            hjust = 1, vjust = 1, size=3.5,
            label =
              paste0(
                "(", r2_ct_uct[3], "-", r2_ct_uct[4], ")")
            )) +
  xlab("Urine mean rpoB Ct") + ylab("Blood mean rpoB Ct") -> g4

```



```

ggExtra::ggMarginal(g4, margins = "x",
                    type = "density",
                    fill="#f68f46ff",
                    alpha=0.25) -> g4

# grid.arrange(g1,g3,g2,g4, nrow=2)

#table(rowSums(qual[,3:6]=="Not available"))

qual %>%
  mutate(
    bld_xpert = na_if(bld_xpert, "Not available"),
    urn_xpert = na_if(urn_xpert, "Not available"),
    dmn_micro = na_if(dmn_micro, "Not available"),
    mfl_culture = na_if(mfl_culture, "Not available")
  ) %>%
  mutate(
    bld_xpert = as.numeric((bld_xpert=="M.tb detected")),
    urn_xpert = as.numeric((urn_xpert=="M.tb detected")),
    dmn_micro = as.numeric((dmn_micro=="M.tb detected")),
    mfl_culture = as.numeric((mfl_culture=="M.tb detected"))
  ) -> qual

u = qual$urn_xpert
b = qual$bld_xpert
d = qual$dmn_micro
m = qual$mfl_culture

k <- c(
  cohen.kappa(table(u,u))$k, cohen.kappa(table(u,b))$k, cohen.kappa(table(u,d))$k, cohen.kappa(table(u,m))$k,
  cohen.kappa(table(b,u))$k, cohen.kappa(table(b,b))$k, cohen.kappa(table(b,d))$k, cohen.kappa(table(b,m))$k,
  cohen.kappa(table(d,u))$k, cohen.kappa(table(d,b))$k, cohen.kappa(table(d,d))$k, cohen.kappa(table(d,m))$k,
  cohen.kappa(table(m,u))$k, cohen.kappa(table(m,b))$k, cohen.kappa(table(m,d))$k, cohen.kappa(table(m,m))$k)
k <- matrix(k, nrow=4, byrow = TRUE)
rownames(k) <- c("Urine Xpert", "Blood Xpert-ultra", "DMN-tre micro.", "MFL blood culture")
colnames(k) <- c("Urine Xpert", "Blood Xpert-ultra", "DMN-tre micro.", "MFL blood culture")

updf <- as.data.frame(qual[,3:6])

UpSetR::upset(updf,
              sets = c("bld_xpert", "urn_xpert",
                      "dmn_micro", "mfl_culture"),
              order.by = "freq", matrix.color = "#332288",
              sets.bar.color = c("#CC6677", "#DDCC77",
                                "#44AA99", "yellow"))

colfunc <- colorRampPalette(c("black", "white"))
pheatmap(k, display_numbers = TRUE, color = colfunc(10) )

```

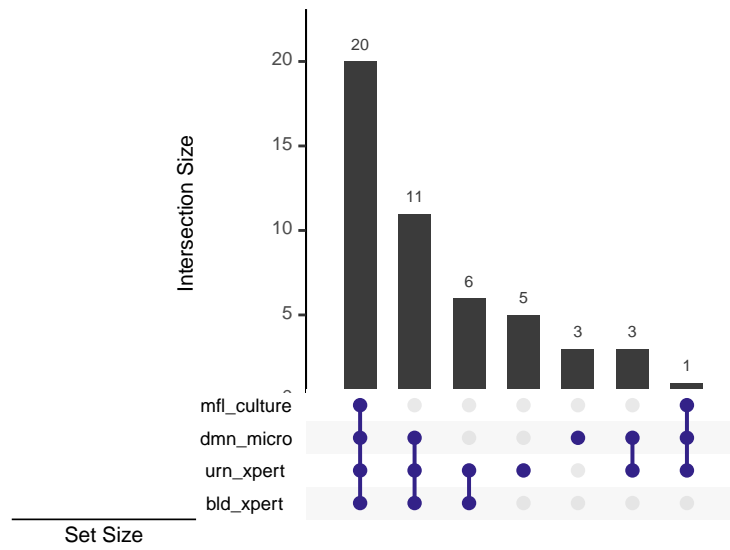


Figure 11: Agreement between qualitative results

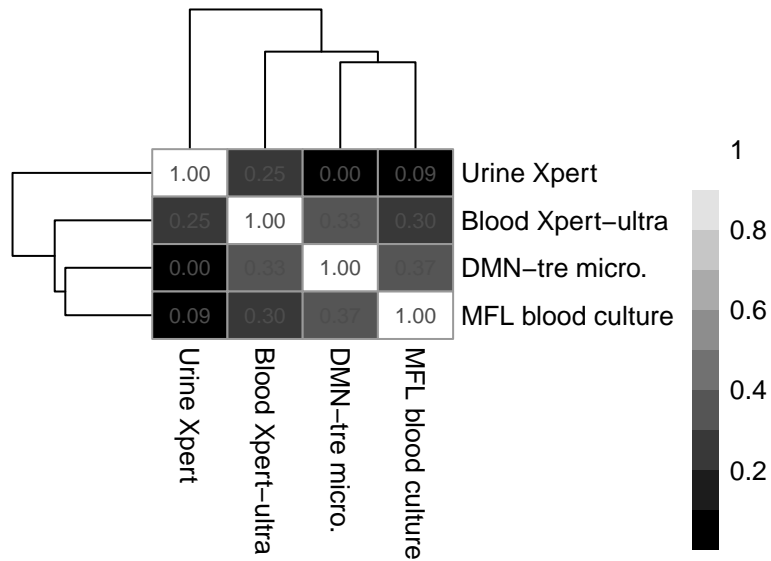


Figure 12: Agreement between qualitative results

```
grid.arrange(g1,g3,g2,g4, nrow=2)
```

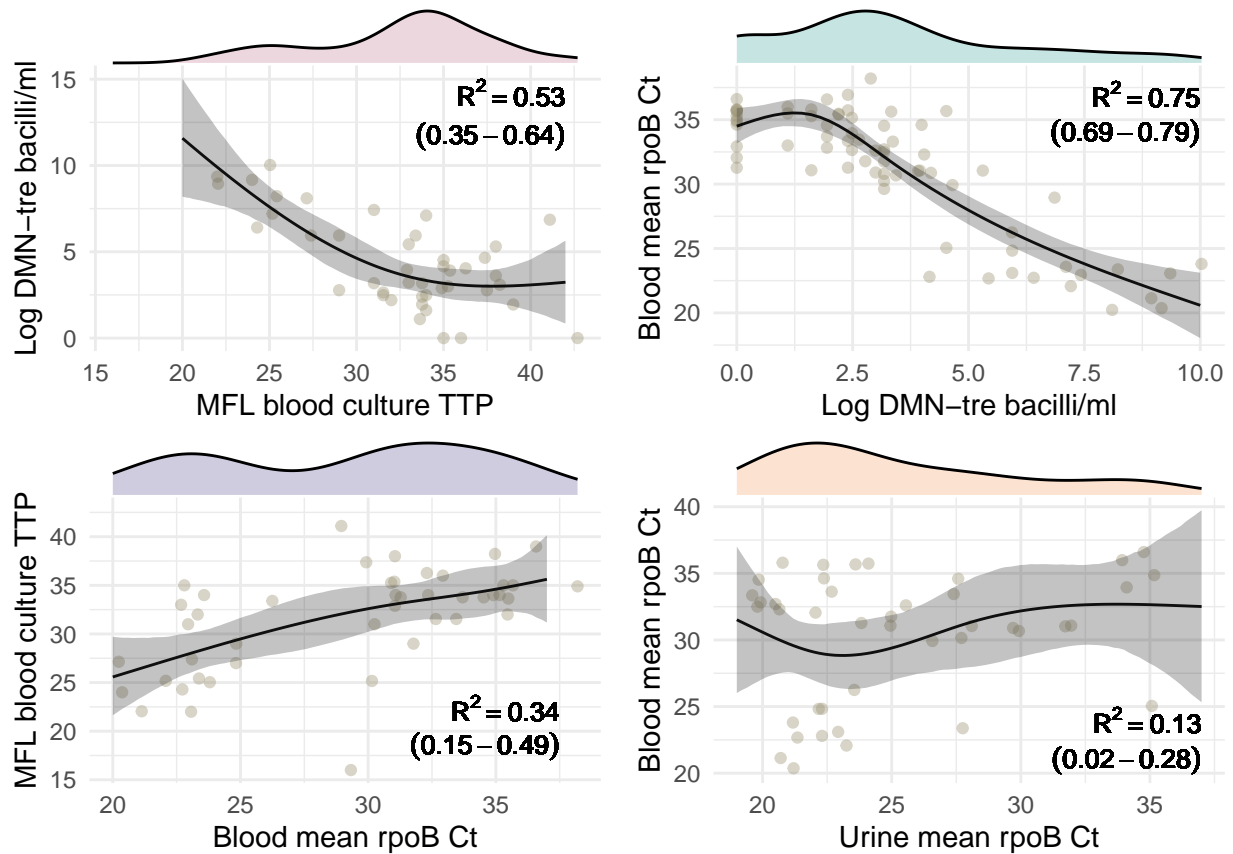


Figure 13: Correlation between quantitative measures of bacillary load

## 9 Ordinal scaling data

```
#### GXP
gxp_original <- bind_rows(
  read_csv("tidy_gxp.csv"),
  read_csv("tidy_gxp2.csv")) # two batches

# empirical LOQ for Xpert = set as max observed Ct + 1
gxp_original %>%
  select(rpoB1_CT, rpoB2_CT,
         rpoB3_CT, rpoB4_CT,
         IS1081_IS6110_CT) %>%
  pivot_longer(1:5) %>%
  summarise(LOQ = max(value)+1) -> LOQ
LOQ <- as.numeric(LOQ$LOQ)

gxp_original %>%
  filter(sample_type=="bldextract") %>%
```

```

mutate(
  rpoB1_CT = ifelse(rpoB1_CT==0, LOQ, rpoB1_CT),
  rpoB2_CT = ifelse(rpoB2_CT==0, LOQ, rpoB2_CT),
  rpoB3_CT = ifelse(rpoB3_CT==0, LOQ, rpoB3_CT),
  rpoB4_CT = ifelse(rpoB4_CT==0, LOQ, rpoB4_CT),
  IS1081_IS6110_CT = ifelse(IS1081_IS6110_CT==0, LOQ, IS1081_IS6110_CT),
  mean_ct = pmap_dbl(
    list(rpoB1_CT, rpoB2_CT, rpoB3_CT, rpoB4_CT, IS1081_IS6110_CT),
    ~mean(c(...))),
  mean_ct =
    ifelse(result=="ERROR", NA, mean_ct),
  result =
    factor(result,
      levels = c("ERROR", "MTB NOT DETECTED",
        "MTB DETECTED TRACE", "MTB DETECTED VERY LOW",
        "MTB DETECTED LOW", "MTB DETECTED MEDIUM",
        "MTB DETECTED HIGH")),
  ct_ordinal = as.integer(cut(-1*mean_ct, breaks = 10))) %>%
select(sample_id, study_id, sample_type, timepoint,
  result, ct_ordinal, mean_ct,
  rpoB1_CT, rpoB2_CT, rpoB3_CT,
  rpoB4_CT, IS1081_IS6110_CT) -> gxp_ord

#### MFL TTP
mfl <- read_csv("MFL.csv")
mfl$MFL[mfl$TTP<10 & !is.na(mfl$TTP)] # mixed (contam) growth but MTB was recovered...

## [1] "MTB" "MTB"

mfl$TTP[mfl$TTP<10] <- NA

# MFL quant results to long format with preseved sample types pellet v lysate
mfl %>%
  select(study_id, timepoint, type, TTP, MFL) %>%
  mutate(
    mfl = str_replace(MFL,
      "contam|lost_viability",
      replacement = "Not available"),
    mfl = replace_na(mfl, "Not available"),
    mfl = str_replace(mfl, "MTB", replacement = "M.tb detected"),
    mfl = ifelse( is.na(TTP) & MFL == "MTB",
      "Not available", mfl)) %>%
  select(-MFL) %>%
  pivot_wider(id_cols = c("study_id", "timepoint"),
    names_from = "type",
    values_from = c("mfl", "TTP")) %>%
  rename_all(tolower) -> quant_mfl

# empirical LOQ for MFL = set as max observed ttp + 1
quant_mfl %>% select(ttp_lystate, ttp_pellet) %>%
  pivot_longer(1:2) %>% summarise(LOQ = max(value, na.rm = TRUE)+1) -> LOQ
LOQ <- as.numeric(LOQ$LOQ)

```

```

# scale the pellet and lysate ttps same
quant_mfl2 <- quant_mfl
m_p_1 <- lm(ttp_lysat ~ rcs(ttp_pellet,5), data=quant_mfl2)
quant_mfl2$ttp_lysat_imputed <- predict(m_p_1, newdata=data.frame(ttp_pellet = quant_mfl2$ttp_pellet))

quant_mfl2 %>%
  mutate(
    ttp =
      case_when(
        is.na(ttp_lysat) & is.na(ttp_lysat_imputed) ~ 999,
        !is.na(ttp_lysat) & is.na(ttp_lysat_imputed) ~ ttp_lysat,
        is.na(ttp_lysat) & !is.na(ttp_lysat_imputed) ~ ttp_lysat_imputed,
        !is.na(ttp_lysat) & is.na(ttp_lysat_imputed) ~ ttp_lysat,
        !is.na(ttp_lysat) & !is.na(ttp_lysat_imputed) ~ (ttp_lysat_imputed + ttp_lysat)/2,
        ttp =
          na_if(ttp, 999),
        ttp_ordinal =
          as.integer(cut(-1*ttp, 9))+1,
        ttp_ordinal =
          ifelse(is.na(ttp_ordinal) &
                (mfl_pellet=="negative" | mfl_lysat=="negative"),
                1, ttp_ordinal)
      ) %>%
    select(study_id, timepoint, mfl_pellet, mfl_lysat,
           ttp, ttp_pellet, ttp_lysat, ttp_ordinal) -> mfl_ord

#### DMN
read_csv("DMN.csv") %>%
  dplyr::mutate(DMN_ml =
    # FOV calculation for /ml
    round((2850/3) * (bacilli/random_fields), 0),
    qualitative_result =
      str_replace(qualitative_result,
                  pattern = "non-quantifiable positive",
                  replacement = "negative") ) -> dmnn

dmnn %>%
  mutate(dmnn_ordinal = as.integer(cut(log(DMN_ml+1), 10))) -> dmnn_ord

#### JOIN
dmnn_ord %>% select(study_id, timepoint, dmnn_ordinal) %>%
  left_join(
    gxp_ord %>% select(study_id, timepoint, ct_ordinal),
    by=c("study_id", "timepoint")
  ) %>%
  left_join(
    mfl_ord %>% select(study_id, timepoint, ttp_ordinal),
    by=c("study_id", "timepoint")
  ) -> df_ord

```

```

df_ord %>%
  group_by(dmn_ordinal, ttp_ordinal) %>% count() %>%
  ggplot(aes(dmn_ordinal, ttp_ordinal, size=n, fill=n)) +
  geom_point(shape=21) +
  scale_fill_gradient(low = canva_pal("Green fields")(4)[3],
                      high = canva_pal("Green fields")(4)[2]) +
  geom_abline(a=0, b=1) +
  theme_minimal() +
  theme(legend.position = "none") -> g1
df_ord %>%
  group_by(dmn_ordinal, ct_ordinal) %>% count() %>%
  ggplot(aes(dmn_ordinal, ct_ordinal, size=n, fill=n)) +
  geom_point(shape=21) +
  scale_fill_gradient(low = canva_pal("Green fields")(4)[3],
                      high = canva_pal("Green fields")(4)[2]) +
  geom_abline(a=0, b=1) +
  theme_minimal() +
  theme(legend.position = "none") -> g2
df_ord %>%
  group_by(ct_ordinal, ttp_ordinal) %>% count() %>%
  ggplot(aes(ct_ordinal, ttp_ordinal, size=n, fill=n)) +
  geom_point(shape=21) +
  scale_fill_gradient(low = canva_pal("Green fields")(4)[3],
                      high = canva_pal("Green fields")(4)[2]) +
  geom_abline(a=0, b=1) +
  theme_minimal() +
  theme(legend.position = "none") -> g3

#grid.arrange(g1, g2, g3)

df_ord %>%
  pivot_longer(3:5,
               names_to = "method",
               values_to = "bacilli") %>%
  mutate(
    method = str_replace_all(method,
                              pattern = "_ordinal",
                              replacement = ""),
    timepoint = round(timepoint/24,2)) -> df_ord

df_ord %>% filter(method=="ct") %>%
  group_by(timepoint, bacilli) %>% count() %>%
  ungroup() %>% group_by(timepoint) %>% rename(obs = n) %>%
  drop_na() %>%
  add_tally(obs) %>%
  mutate(prob = obs/n,
          timepoint = str_c(timepoint, " days")) %>%
  ggplot(aes(bacilli, prob,
             fill=as.factor(timepoint),
             group=as.factor(timepoint))) +
  geom_bar(stat="identity", colour="black", alpha=0.5) +

```

```

facet_wrap(~timepoint, ncol=1) +
theme_minimal() +
scale_fill_manual(values = viridis_pal(option = "E")(6) ) +
theme(legend.position = "none") + ylab("") +
xlab("Bacilli quantification \n(ordinal)") +
ggtitle("Xpert-ultra") -> g_gxp_ord

df_ord %>% filter(method=="dmn") %>%
group_by(timepoint, bacilli) %>% count() %>%
ungroup() %>% group_by(timepoint) %>% rename(obs = n) %>%
drop_na() %>%
add_tally(obs) %>%
mutate(prob = obs/n,
        timepoint = str_c(timepoint, " days")) %>%
ggplot(aes(bacilli, prob,
            fill=as.factor(timepoint),
            group=as.factor(timepoint))) +
geom_bar(stat="identity", colour="black", alpha=0.5) +
facet_wrap(~timepoint, ncol=1) +
theme_minimal() +
scale_fill_manual(values = viridis_pal(option = "E")(6) ) +
theme(legend.position = "none") + ylab("") +
xlab("Bacilli quantification \n(ordinal)") +
ggtitle("DMNtre") -> g_dmn_ord

df_ord %>% filter(method=="ttp") %>%
group_by(timepoint, bacilli) %>% count() %>%
ungroup() %>% group_by(timepoint) %>% rename(obs = n) %>%
drop_na() %>%
add_tally(obs) %>%
mutate(prob = obs/n,
        timepoint = str_c(timepoint, " days")) %>%
ggplot(aes(bacilli, prob,
            fill=as.factor(timepoint),
            group=as.factor(timepoint))) +
geom_bar(stat="identity", colour="black", alpha=0.5) +
facet_wrap(~timepoint, ncol=1) +
theme_minimal() +
scale_fill_manual(values = viridis_pal(option = "E")(6) ) +
theme(legend.position = "none") + ylab("") +
xlab("Bacilli quantification \n(ordinal)") +
ggtitle("MFL") -> g_mfl_ord

df_ord %>%
group_by(timepoint, bacilli) %>% count() %>%
ungroup() %>% group_by(timepoint) %>% rename(obs = n) %>%
drop_na() %>%
add_tally(obs) %>%
mutate(prob = obs/n,
        timepoint = str_c(timepoint, " days")) %>%
ggplot(aes(bacilli, prob,
            group=as.factor(timepoint))) +
geom_bar(stat="identity", fill="black", colour="black", alpha=0.8) +

```

```
facet_wrap(~timepoint, ncol=1) +
theme_minimal() +
theme(legend.position = "none") + ylab("Proportion") +
xlab("Bacilli quantification \n(ordinal)") +
ggtitle("Aggregated")-> g_combined_ord
```

```
g1 + g2 + g3 +
plot_layout(ncol=1)
```

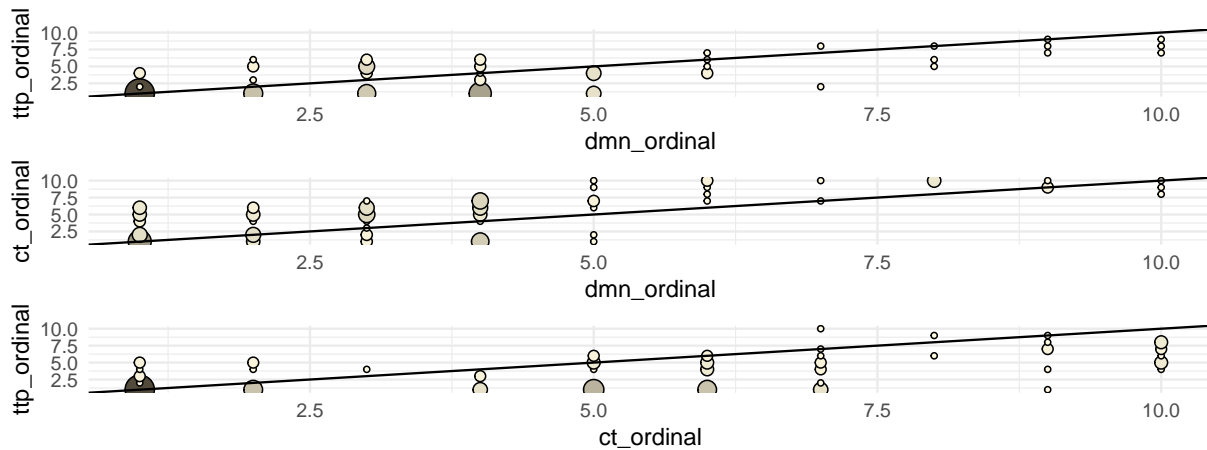


Figure 14: Correlation of ordinal results by assay

```
grid.arrange(g_combined_ord,
g_dmn_ord,
g_mfl_ord,
g_gxp_ord, nrow=1)
```

## 10 Extracting data from KDHTB cohort to use as priors in ordinal regression model

```
# get an informative prior for the intercept effect of died
# on bacilli ordinal result using data from main KDHTB study

kdhtb <- read_csv("kdhtb_cohort_bloodxpertultra.csv")

kdhtb %>%
  select(rpoB1, rpoB2, rpoB3, rpoB4, IS1081_IS6110) %>%
  pivot_longer(1:5) %>% summarise(LOQ = max(value, na.rm = T)+1) -> LOQ
LOQ <- as.numeric(LOQ$LOQ)

kdhtb %>%
  filter(confirmed_tb==1,
         !is.na(result),
         result!="Error") %>%
  mutate(
```



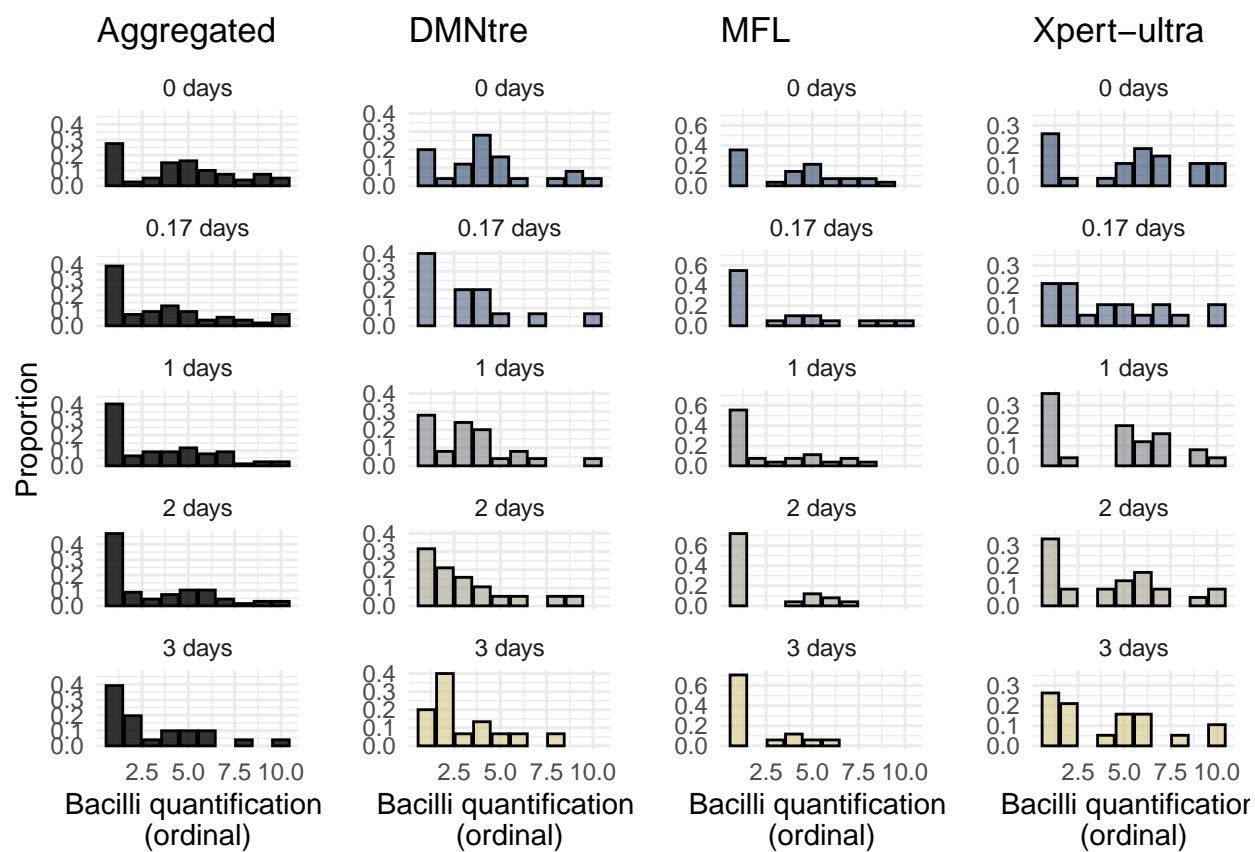


Figure 15: Ordinal results by time

```

rpoB1 = ifelse(rpoB1==0, LOQ, rpoB1),
rpoB2 = ifelse(rpoB2==0, LOQ, rpoB2),
rpoB3 = ifelse(rpoB3==0, LOQ, rpoB3),
rpoB4 = ifelse(rpoB4==0, LOQ, rpoB4),
IS1081_IS6110 = ifelse(IS1081_IS6110==0, LOQ, IS1081_IS6110),
mean_ct = pmap_dbl(
  list(rpoB1, rpoB2, rpoB3, rpoB4, IS1081_IS6110),
  ~mean(c(...))),
ct_ordinal = as.integer(cut(-1*mean_ct, breaks = 10)),
died = as.numeric(outcome=="Died")
) %>%
select(ct_ordinal, died) -> kdhtb

ggplot(kdhtb, aes(ct_ordinal, fill=as.factor(died))) +
  geom_histogram(binwidth = 1,
    colour="black", alpha=0.7) +
  scale_fill_manual(values = viridis_pal(option="E")(8)[c(2,6)]) +
  facet_wrap(~died) +
  theme_dab() -> kdhtb_bldxpt_ordinal

kdhtb_fit <- brm(
  ct_ordinal ~ died,
  data = kdhtb,
  family = cumulative("logit"),
  control = list(adapt_delta = 0.97),
  seed = 140916,
  cores=2)

kdhtb_bldxpt_ordinal

```

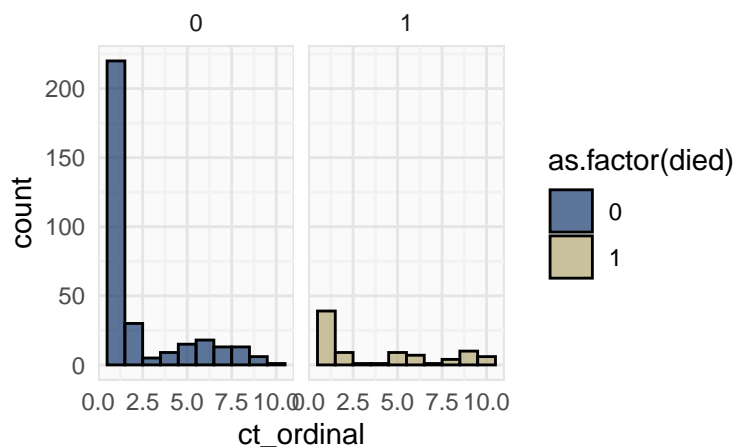


Figure 16: Blood Xpert-ultra Ct value distribution by outcome in KDHTB study

Giving these posterior distributions for parameters relating mortality to blood Xpert-ultra Ct value in KDHTB, to use to inform priors for ordinal regression model:

```
mean(posterior_samples(kdhtb_fit)$b_died)
```

```
## [1] 1.03167
```

```
sd(posterior_samples(kdhtb_fit)$b_died)
```

```
## [1] 0.2260973
```

## 11 Ordinal regression model relating blood bacilliary load dynamics to mortality

```
df_ord %>%
  left_join(pids, by = "study_id") %>%
  select(study_id = pid, timepoint, method, bacilli) -> df_ord

# (this is just a df to annotate plot):
oc2 <- oc %>% mutate(
  outcome = ifelse(died==1, "Died", "survived"),
  timepoint = 1,
  bacilli = ifelse(
    study_id=="pid3" | study_id == "pid8" | study_id == "pid10",
    1, 10
  ))

df_ord %>%
  mutate(study_id = factor(study_id,
                           levels = paste0("pid", 1:28))) %>%
  ggplot(aes(timepoint, bacilli)) +
  geom_point(aes(colour=method)) +
  geom_smooth(method = "lm", se=FALSE, aes(colour=method)) +
  geom_smooth(method = "lm", se=FALSE, colour="black") +
  facet_wrap(~study_id, nrow=7) +
  theme_minimal() +
  scale_colour_manual(values = viridis_pal(option = "D")(3) ) +
  geom_text(data = oc2,
            mapping = aes(label=outcome),
            colour="black", size=3) -> g_ord_study_id

rm(oc2)
```

```
g_ord_study_id
```

```
left_join(
  df_ord,
  oc,
  by = "study_id"
) -> df_ord

fit <- brm(
  bacilli ~ timepoint + died + timepoint:died + method + timepoint:method +
    (timepoint|study_id),
  data = df_ord,
  family = cumulative("logit"),
  prior = c(
    set_prior("normal(1,0.5)", class = "b", coef = "died"),
    set_prior("normal(-0.5,1)", class = "b", coef = "timepoint"),
```

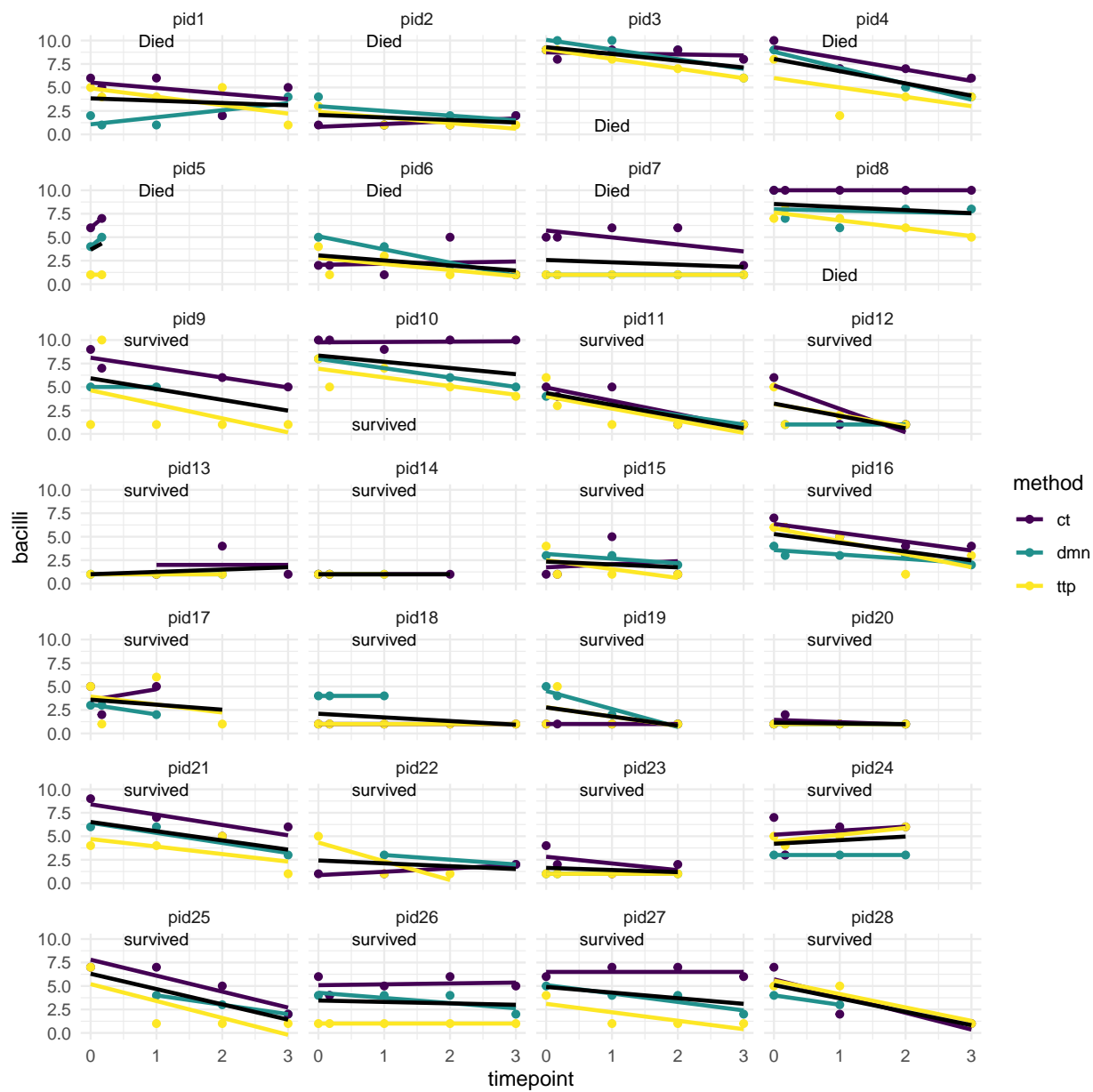


Figure 17: Ordinal data by time and patient, simple linear fit

```

    set_prior("normal(0,1)", class = "b", coef = "timepoint:died"),
    set_prior("normal(0,1)", class = "b", coef = "methoddmn"),
    set_prior("normal(0,1)", class = "b", coef = "methodttp"),
    set_prior("normal(0,2)", class = "b", coef = "timepoint:methoddmn"),
    set_prior("normal(0,2)", class = "b", coef = "timepoint:methodttp"),
    set_prior("student_t(3, 0, 2.5)", class = "sd",
              coef = "Intercept", group = "study_id"),
    set_prior("student_t(3, 0, 2.5)", class = "sd",
              coef = "timepoint", group = "study_id")
  ),
  control = list(adapt_delta = 0.97),
  seed = 140916,
  iter = 8000, warmup = 2000, chains=2,
  cores=4, sample_prior = TRUE
)

bind_rows(
  hypothesis(fit, hypothesis = "timepoint < 0" )$hypothesis,
  hypothesis(fit, hypothesis = "died > 0" )$hypothesis,
  hypothesis(fit, hypothesis = "timepoint:died > 0" )$hypothesis,
  hypothesis(fit, hypothesis = "methoddmn < 0")$hypothesis,
  hypothesis(fit, hypothesis = "methodttp < 0")$hypothesis,
  hypothesis(fit, hypothesis = "timepoint:methoddmn < 0")$hypothesis,
  hypothesis(fit, hypothesis = "timepoint:methodttp < 0")$hypothesis
) %>%
as_tibble() %>% select(-Star, -Hypothesis) %>%
mutate(
  `Hypothesis statement` = c(
    "Decrease in bacilli load over time on Rx",
    "Patients who died had higher bacilli load",
    "Patients who died had slower decrease in bacilli load over time on Rx",
    "DMN-tre observations of bacilli load were lower than Xpert-ultra",
    "Culture TTP observations of bacilli load were lower than Xpert-ultra",
    "DMN-tre observations of bacilli load declined faster than Xpert-ultra",
    "Culture TTP observations of bacilli load declined faster than Xpert-ultra"
  ),
  `coefficient (beta)` = c(
    "Time in days (slope)",
    "Died (intercept)",
    "Time * Died interaction (slope)",
    "DMN-tre (intercept)",
    "Culture-TTP (intercept)",
    "Time * DMN-tre interaction (slope)",
    "Time * Culture-TTP interaction (slope)"
  ),
  h1 = c("<0", ">0", ">0", "<0", "<0", "<0", "<0")
) %>%
select(`coefficient (beta)`, h1, `Hypothesis statement`, everything()) -> h1df

h1df[,4:8] <- round(h1df[,4:8], 2)
h1df[,9] <- round(h1df[,9], 3)

```

```

max_iter = 1e3

posterior_samples(fit, subset = 1:max_iter) %>%
  select(
    b_timepoint, b_died, `b_timepoint:died`,
    b_methoddmn, b_methodttp,
    `b_timepoint:methoddmn`, `b_timepoint:methodttp`
  ) %>%
  pivot_longer(1:7,
    names_to = "coefficient",
    values_to = "posterior_draw") %>%
  mutate(
    coefficient = factor(
      coefficient,
      levels = c("b_timepoint", "b_died", "b_timepoint:died",
        "b_methoddmn", "b_methodttp",
        "b_timepoint:methoddmn", "b_timepoint:methodttp"),
      labels = c(
        "Time in days (slope)",
        "Died (intercept)",
        "Time * Died interaction (slope)",
        "DMN-tre (intercept)",
        "Culture-TTP (intercept)",
        "Time * DMN-tre interaction (slope)",
        "Time * Culture-TTP interaction (slope)"
      )
    )
  )
)) -> post_samples

annotate_df <- data.frame(
  coefficient = factor(
    c("b_timepoint", "b_died", "b_timepoint:died",
      "b_methoddmn", "b_methodttp",
      "b_timepoint:methoddmn", "b_timepoint:methodttp"),
    levels = c("b_timepoint", "b_died", "b_timepoint:died",
      "b_methoddmn", "b_methodttp",
      "b_timepoint:methoddmn", "b_timepoint:methodttp"),
    labels = c(
      "Time in days (slope)",
      "Died (intercept)",
      "Time * Died interaction (slope)",
      "DMN-tre (intercept)",
      "Culture-TTP (intercept)",
      "Time * DMN-tre interaction (slope)",
      "Time * Culture-TTP interaction (slope)"
    )
  ),
  h1 = c("<0", ">0", ">0", "<0", "<0", "<0", "<0"),
  post_prob = signif(h1df$Post.Prob, 3),
  evid_ratio = formatC(h1df$Evid.Ratio, digits = 2),
  y1 = 4.5, y2=6, y3=8.5
)

```

## 11.1 Summary of model

```
fit
```

```
## Family: cumulative
## Links: mu = logit; disc = identity
## Formula: bacilli ~ timepoint + died + timepoint:died + method + timepoint:method + (timepoint | study_id)
## Data: df_ord (Number of observations: 330)
## Samples: 2 chains, each with iter = 8000; warmup = 2000; thin = 1;
##           total post-warmup samples = 12000
##
## Group-Level Effects:
## ~study_id (Number of levels: 28)
##
```

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat
sd(Intercept)	2.71	0.47	1.95	3.77	1.00
sd(timepoint)	0.28	0.19	0.01	0.69	1.00
cor(Intercept,timepoint)	-0.12	0.45	-0.89	0.84	1.00

```
## Bulk_ESS Tail_ESS
## sd(Intercept)      3175    5615
## sd(timepoint)      2559    5478
## cor(Intercept,timepoint) 10335    6476
##
## Population-Level Effects:
##
```

	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS
Intercept[1]	-2.87	0.60	-4.09	-1.74	1.00	2434
Intercept[2]	-2.20	0.59	-3.41	-1.10	1.00	2419
Intercept[3]	-1.69	0.59	-2.88	-0.59	1.00	2391
Intercept[4]	-0.79	0.58	-1.98	0.29	1.00	2494
Intercept[5]	0.40	0.57	-0.77	1.47	1.00	2533
Intercept[6]	1.65	0.58	0.48	2.77	1.00	2698
Intercept[7]	2.95	0.62	1.75	4.16	1.00	3093
Intercept[8]	3.76	0.64	2.51	5.03	1.00	3287
Intercept[9]	4.87	0.68	3.54	6.21	1.00	3692
timepoint	-0.58	0.20	-0.97	-0.20	1.00	7647
died	1.14	0.46	0.22	2.04	1.00	6979
methoddmn	-1.00	0.36	-1.70	-0.30	1.00	8534
methodttp	-1.32	0.37	-2.05	-0.62	1.00	8888
timepoint:died	0.29	0.25	-0.20	0.79	1.00	9535
timepoint:methoddmn	-0.25	0.23	-0.70	0.20	1.00	7791
timepoint:methodttp	-0.94	0.26	-1.46	-0.44	1.00	8774

```
## Tail_ESS
## Intercept[1]      4903
## Intercept[2]      4625
## Intercept[3]      4731
## Intercept[4]      4756
## Intercept[5]      4867
## Intercept[6]      5134
## Intercept[7]      5617
## Intercept[8]      5828
## Intercept[9]      6586
## timepoint        8112
## died             7922
## methoddmn        8519
## methodttp        9507
```

```
## timepoint:died      8285
## timepoint:methoddmn 7865
## timepoint:methodttp 8801
##
## Family Specific Parameters:
##      Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## disc      1.00      0.00      1.00      1.00 1.00      12000      12000
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

## 11.2 Formal hypothesis testing and parameter posterior distributions

```
kable(hidf, caption = "Model parameters: posterior distribution and hypothesis tests", booktabs = T) %>%
  kable_styling(latex_options = c("striped", "hold_position", "scale_down"))
```

Table 1: Model parameters: posterior distribution and hypothesis tests

coefficient (beta)	h1	Hypothesis statement	Estimate	Est.Error	CI.Lower	CI.Upper	Evid.Ratio	Post.Prob
Time in days (slope)	<0	Decrease in bacilli load over time on Rx	-0.58	0.20	-0.90	-0.26	665.67	0.998
Died (intercept)	>0	Patients who died had higher bacilli load	1.14	0.46	0.38	1.88	132.33	0.992
Time * Died interaction (slope)	>0	Patients who died had slower decrease in bacilli load over time on Rx	0.29	0.25	-0.12	0.70	7.08	0.876
DMN-tre (intercept)	<0	DMN-tre observations of bacilli load were lower than Xpert-ultra	-1.00	0.36	-1.59	-0.40	362.64	0.997
Culture-TTP (intercept)	<0	Culture TTP observations of bacilli load were lower than Xpert-ultra	-1.32	0.37	-1.93	-0.73	11999.00	1.000
Time * DMN-tre interaction (slope)	<0	DMN-tre observations of bacilli load declined faster than Xpert-ultra	-0.25	0.23	-0.62	0.13	6.29	0.863
Time * Culture-TTP interaction (slope)	<0	Culture TTP observations of bacilli load declined faster than Xpert-ultra	-0.94	0.26	-1.38	-0.52	11999.00	1.000

```
post_samples %>%
  ggplot(
    aes(fct_rev(coefficient), posterior_draw)
  ) +
  # ggtitle("Posterior probability density\nfor population coefficients") +
  theme_minimal() +
  theme(plot.margin = unit(c(3,3,3,1), "lines"),
        panel.grid = element_blank(),
        plot.title=element_text(size=10,face="bold", hjust=0)) +
  annotation_custom(
    grob = linesGrob(gp = gpar(col="grey90")),
    ymin =-3, ymax=10, xmin = 0.5, xmax=0.5
  ) +
  annotation_custom(
    grob = linesGrob(gp = gpar(col="grey90")),
    ymin =-3, ymax=10, xmin = 1.5, xmax=1.5
  ) +
  annotation_custom(
    grob = linesGrob(gp = gpar(col="grey90")),
    ymin =-3, ymax=10, xmin = 2.5, xmax=2.5
  ) +
  annotation_custom(
    grob = linesGrob(gp = gpar(col="grey90")),
    ymin =-3, ymax=10, xmin = 3.5, xmax=3.5
  ) +
  annotation_custom(
    grob = linesGrob(gp = gpar(col="grey90")),
    ymin =-3, ymax=10, xmin = 4.5, xmax=4.5
  )
```



```

) +
annotation_custom(
  grob = linesGrob(gp = gpar(col="grey90")),
  ymin=-3, ymax=10, xmin = 5.5, xmax=5.5
) +
annotation_custom(
  grob = linesGrob(gp = gpar(col="grey90")),
  ymin=-3, ymax=10, xmin = 6.5, xmax=6.5
) +
annotation_custom(
  grob = linesGrob(gp = gpar(col="grey90")),
  ymin=-3, ymax=10, xmin = 7.5, xmax=7.5
) +
annotation_custom(
  grob = linesGrob(),
  ymin =3.5, ymax=10, xmin = 0.4, xmax=0.4
) +
annotation_custom(
  grob = linesGrob(),
  ymin =3.5, ymax=10, xmin = 7.5, xmax=7.5
) +
annotation_custom(
  grob = linesGrob(),
  ymin =3, ymax=-3, xmin = 7.5, xmax=7.5
) +
geom_hline(yintercept = 0) +
geom_violin(
  fill = viridis_pal(option="E")(8)[2],
  alpha=0.5,
  colour = viridis_pal(option="E")(8)[2]
) +
geom_boxplot(colour = "black", fill="white", alpha=0.7,
  outlier.alpha = 0, width=0.1) +
ylab("") + xlab("") +
coord_flip(clip = "off") +
annotation_custom(
  grob = textGrob(label = "Higher bacilli load",
    hjust = 0, gp = gpar(fontsize = 10)),
  ymin =0.3, ymax=0.3, xmin = -0.5, xmax=-0.5
) +
annotation_custom(
  grob = textGrob(label = "Lower bacilli load",
    hjust = 1, gp = gpar(fontsize = 10)),
  ymin =-0.3, ymax=-0.3, xmin = -0.5, xmax=-0.5
) +
annotation_custom(
  grob = textGrob(label = "Estimate for coefficient (logit scale)",
    gp = gpar(fontsize = 10, fontface="bold")),
  ymin =0, ymax=0, xmin = -1.5, xmax=-1.5
) +
annotation_custom(
  grob = linesGrob(
    arrow = arrow(type = "closed",

```

```

        ends = "last",
        length = unit(.2, "cm"))),
  ymin=0.1, ymax=3, xmin = 0.4, xmax=0.4
) +
annotation_custom(
  grob = linesGrob(
    arrow = arrow(type = "closed",
      ends = "first",
      length = unit(.2, "cm"))),
  ymin=-0.1, ymax=-3, xmin = 0.4, xmax=0.4
) +
geom_text(
  data = annotate_df,
  aes(x = coefficient, y=y1, label = h1),
  hjust = 0, size=3.5
) +
geom_text(
  data = annotate_df,
  aes(x = coefficient, y=y2, label = post_prob),
  hjust = 0, size=3.5
) +
geom_text(
  data = annotate_df,
  aes(x = coefficient, y=y3, label = evid_ratio),
  hjust = 0, size=3.5
) + scale_y_continuous(breaks=c(-2,-1,0, 1, 2)) +
geom_text(label = "Hypothesis", x=7.75, y=4, size=3.5,
  hjust=0.2, vjust=0) +
geom_text(label = "Posterior\nprobability", x=7.75, y=6, size=3.5,
  hjust=0, vjust=0) +
geom_text(label = "Evidence\nratio", x=7.75, y=8.5, size=3.5,
  hjust=0, vjust=0) +
geom_text(
  label = "Posterior probability density\nfor population coefficients",
  x=7.75, y=0, size=3.5,
  hjust=0.5, vjust=0) -> g_coef

```

g\_coef

### 11.3 Expected probabilities of specific ordinal values by time and assay

```

nd <- data.frame(
  timepoint = c(0,0.2,1,2,3),
  died = rep(c(0,1), each=5),
  method = rep(c("dmn", "ct", "ttp"), each=10)
)

fitted(fit,
  newdata=nd,
  summary = TRUE,
  re_formula = NA)[,1,] %>%
as_tibble() %>%

```

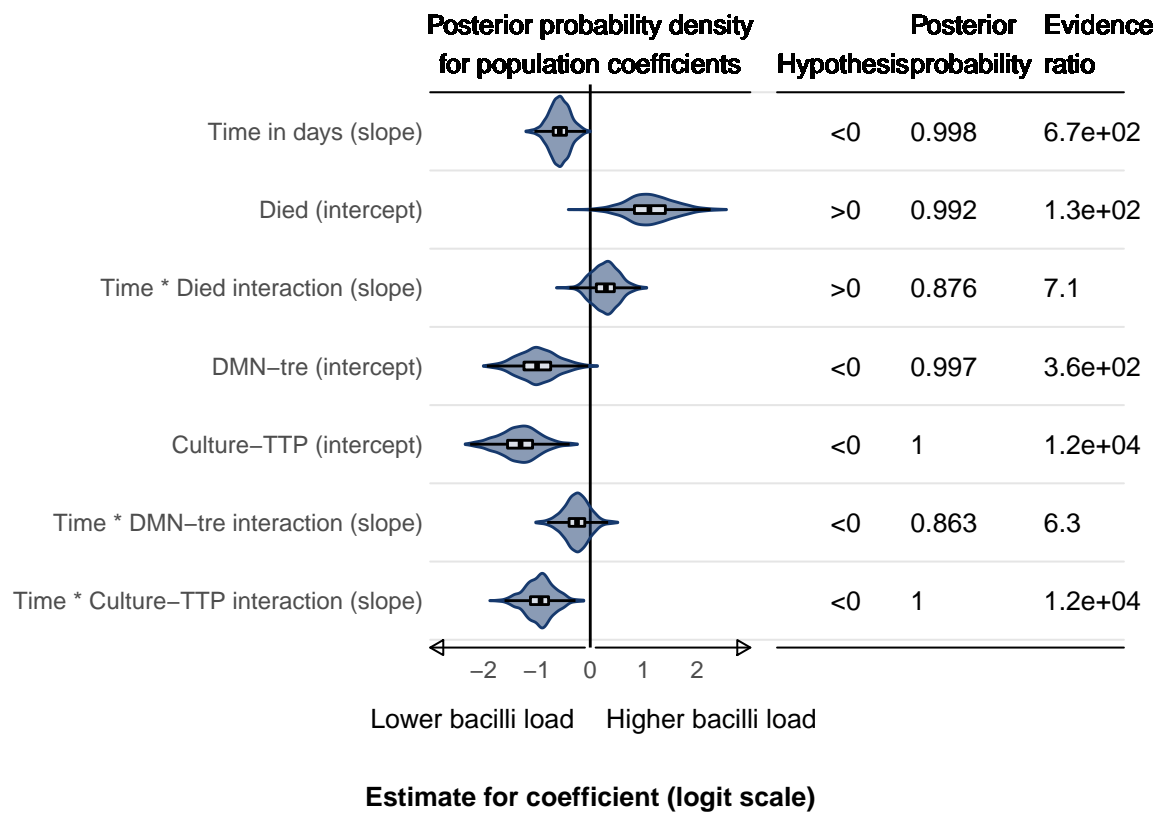


Figure 18: Posterior distribution model parameters and hypothesis tests

```

gather(key = "bacilli") %>%
mutate(
  timepoint = rep(nd$timepoint,10),
  died = rep(nd$died,10),
  method = rep(nd$method,10),
  method = case_when(
    method=="ct" ~ "Xpert-ultra",
    method=="dmn" ~ "DMN-tre",
    method=="ttp" ~ "MFL culture"
  ),
  bacilli = factor(bacilli,
    levels = 10:1),
  outcome = ifelse(died==1, "Died", "Survived")
) %>%
ggplot(
  aes(as.factor(timepoint), value, fill=bacilli)
) +
geom_bar(position="stack", stat = "identity",
  colour="grey80", width = 1, alpha=0.8, size=0.2) +
theme_minimal() + theme(legend.position = "bottom") +
scale_fill_viridis_d(option = "E",
  direction = -1, begin = 0.1 ) +
facet_wrap(method~outcome, ncol = 2 ) +
xlab("Time from start treatment (days)") +
ylab("Expected proportion of patients in ordinal category") +
labs(fill = "Ordinal\nScale") +
theme(
  # Change legend key size and key width
  legend.key.size = unit(0.4, "cm"),
  legend.key.width = unit(0.2,"cm"),
  strip.text.x = element_text( margin = margin( b = 0, t = 0) )
) -> g_expected_ordinal

# Expected probability of negative sample by 72 hours Rx

nd <- data.frame(
  timepoint = 3,
  died = rep(c(0,1), each=3),
  method = rep(c("dmn", "ct", "ttp"), 2)
)

max_iter = 1e3

fitted(fit,
  newdata=nd,
  summary = FALSE,
  re_formula = NA,
  subset = 1:max_iter)[,1] %>%
as_tibble() -> d3_pp_neg_wide

d3_pp_neg_wide %>%
  pivot_longer(1:nrow(nd),

```

```

        names_to = "condition",
        values_to = "post_prob") %>%
mutate(
  method = rep(nd$method, max_iter),
  died = rep(nd$died, max_iter),
  outcome = ifelse(died==1, "Died", "Survived")
) -> d3_pp_neg_long

d3_pp_neg_long %>%
  group_by(method, outcome) %>%
  summarise(
    median = median(post_prob),
    mad = mad(post_prob),
    q.025 = quantile(post_prob, 0.025),
    q.975 = quantile(post_prob, 0.975)
  ) -> d3_pp_neg_table

names(d3_pp_neg_wide) <- c("dmn_0", "ct_0", "ttp_0", "dmn_1", "ct_1", "ttp_1")
d3_pp_neg_wide %>%
  mutate(dmn_p = dmn_0>dmn_1,
         ct_p = ct_0>ct_1,
         ttp_p = ttp_0>ttp_1) %>%
  summarise(
    dmn_delta = sum(dmn_p)/length(dmn_p),
    ct_delta = sum(ct_p)/length(ct_p),
    ttp_delta = sum(ttp_p)/length(ttp_p)
  ) -> d3_pp_neg_delta

# Expected time to 50% probability sterilisation of blood

nd <- data.frame(
  timepoint = rep(seq(0.1,50, length.out = 500), 6),
  died = rep(c(0,1), each=1500),
  method = rep(c("dmn", "ct", "ttp"), each=500)
)

max_iter = 1e3

fitted(fit,
  newdata=nd,
  summary = FALSE,
  re_formula = NA,
  subset = 1:max_iter)[,1] %>%
as_tibble() %>%
pivot_longer(1:nrow(nd),
  names_to = "condition",
  values_to = "post_prob") %>%
mutate(
  iter = rep(1:max_iter, each=nrow(nd)),
  method = rep(nd$method, max_iter),
  died = rep(nd$died, max_iter),
  timepoint = rep(nd$timepoint, max_iter),

```

```

    outcome = ifelse(died==1, "Died", "Survived")) %>%
  filter(post_prob >= 0.5) %>%
  select(-condition, -outcome) %>%
  group_by(iter, method, died) %>%
  arrange(post_prob, by_group=TRUE) %>%
  slice(n=1) -> time_to_50perc_neg

time_to_50perc_neg %>%
  ungroup() %>% group_by(method, died) %>%
  summarise(
    median = median(timepoint),
    `IQR lower` = quantile(timepoint, 0.25),
    `IQR upper` = quantile(timepoint, 0.75)
  ) -> tt50_table

time_to_50perc_neg %>%
  select(-post_prob) %>%
  pivot_wider(names_from = c("method", "died"),
              values_from = "timepoint" ) %>%
  mutate(dmn_p = dmn_0<dmn_1,
         ct_p = ct_0<ct_1,
         ttp_p = ttp_0<ttp_1) %>%
  ungroup() %>%
  summarise(
    dmn_delta = sum(dmn_p, na.rm = T)/sum(!is.na(dmn_p)),
    ct_delta = sum(ct_p, na.rm = T)/sum(!is.na(ct_p)),
    ttp_delta = sum(ttp_p, na.rm = T)/sum(!is.na(ttp_p))
  ) -> tt50_delta_prob_table

g_expected_ordinal

```

Extrapolated median time (days) to 50% probability of negative sample (+ IQR), by assay and outcome status:

```

kable(tt50_table, caption = "Model estimated time in days to 50% probability of negative sample", booktabs = TRUE,
      kable_styling(latex_options = c("striped", "hold_position")))

```

\begin{table}[!h]

\caption{\label{tab:extrapolation for time to 50% clearing BSI}Model estimated time in days to 50% probability of negative sample}

method	died	median	IQR lower	IQR upper
ct	0	4.8	4.00	6.1
ct	1	11.5	8.25	18.4
dmn	0	2.3	1.80	2.8
dmn	1	5.5	4.20	7.6
ttp	0	1.0	0.80	1.2
ttp	1	2.2	1.80	2.6

\end{table}

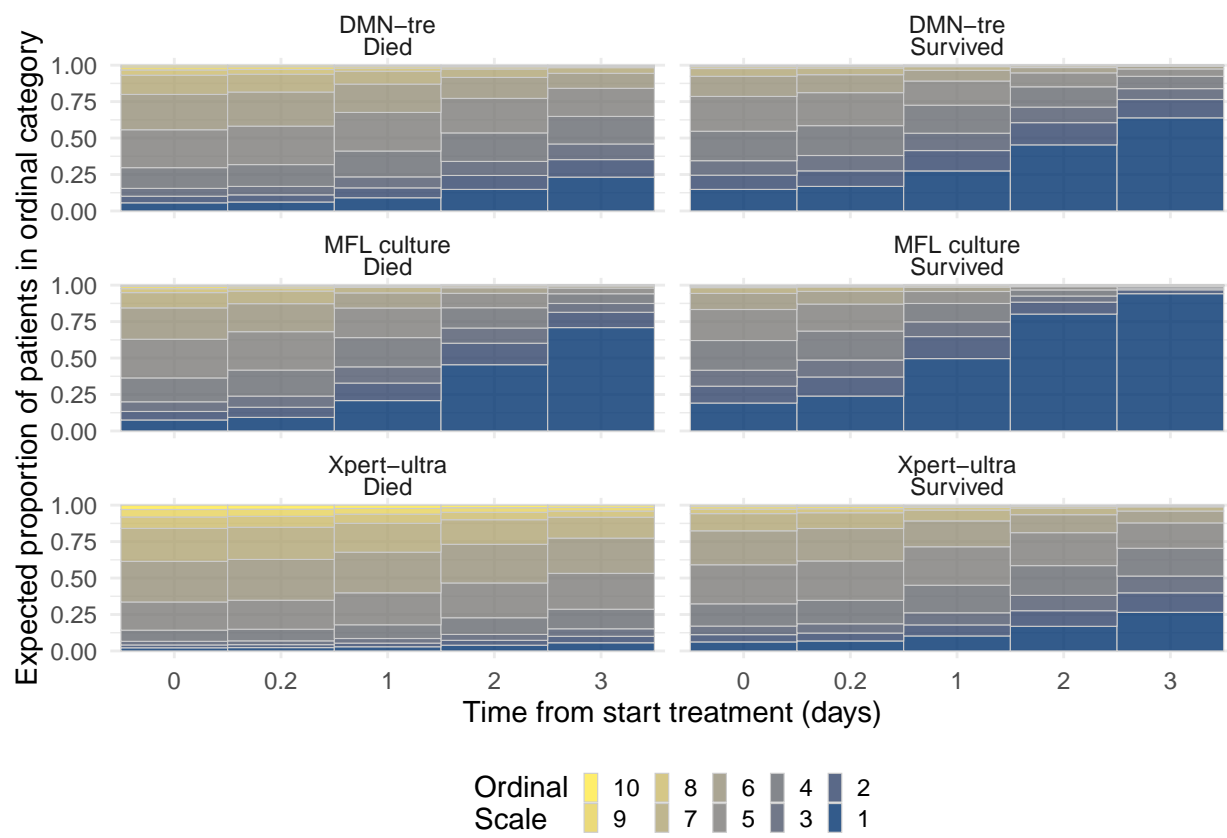


Figure 19: Expected proportions of patients in ordinal categories by time and assay

## 11.4 Prediction intervals for new patient

```
nd <- data.frame(
  timepoint = c(0, 0.167, 1, 2, 3),
  died = rep(c(0,1), each=5),
  study_id = 999,
  method = rep(c("dmn", "ct", "ttp"), each=10)
)
max_iter = 2e3

set.seed(140916)

posterior_predict(
  fit,
  newdata=nd,
  summary = FALSE,
  re_formula = bacilli ~ timepoint + died + timepoint:died +
    method + timepoint:method +
    (timepoint | study_id) ,
  subset = 1:max_iter,
  allow_new_levels = TRUE,
  sample_new_levels = "gaussian",
  seed = 140916) %>%
  as_tibble() %>%
  pivot_longer(1:nrow(nd),
    names_to = "condition",
    values_to = "bacilli") %>%

  mutate(
    timepoint = rep(nd$timepoint, max_iter),
    died = rep(nd$died, max_iter),
    method = rep(nd$method, max_iter)
  ) %>%
  select(-condition) %>%
  group_by(method, died, timepoint) %>%
  summarise(
    predicted_bacilli = median(bacilli),
    predicted_q_l = quantile(bacilli, probs = 0.25),
    predicted_q_h = quantile(bacilli, probs = 0.75)
  ) %>%
  mutate(bacilli = predicted_bacilli,
    outcome = ifelse(died==1, "Died", "Survived"),
    method = case_when(
      method=="ct" ~ "Xpert-ultra",
      method=="dmn" ~ "DMN-tre",
      method=="ttp" ~ "MFL culture"
    )) -> pred_new_pt

###

ggplot() +
  geom_point(
    data = df_ord %>%
      mutate(
```



```

outcome = ifelse(died==1, "Died", "Survived"),
method = case_when(
  method=="ct" ~ "Xpert-ultra",
  method=="dmn" ~ "DMN-tre",
  method=="ttp" ~ "MFL culture"
)),
aes(x=timepoint, y=bacilli, group=study_id, colour=method),
position = position_jitter(width=0.05, height=0.1),
alpha=0.4, size=0.9
) +
geom_line(
  data = df_ord %>% drop_na() %>%
  mutate(
    outcome = ifelse(died==1, "Died", "Survived"),
    method = case_when(
      method=="ct" ~ "Xpert-ultra",
      method=="dmn" ~ "DMN-tre",
      method=="ttp" ~ "MFL culture"
    )),
  aes(x=timepoint, y=bacilli, group=study_id, colour=method),
  size=0.5
) +
geom_line(
  data = pred_new_pt,
  aes(timepoint, bacilli),
  colour="black", size=1.1
) +
geom_ribbon(
  data = pred_new_pt,
  mapping = aes(x=timepoint,
    ymin=predicted_q_l, ymax=predicted_q_h),
  alpha=0.6, colour="grey80", fill="grey"
) +
facet_wrap(method~outcome, nrow=3) +
theme_dab() +
scale_colour_manual(values = viridis_pal(option = "D")(20)[c(16,3,9)]) +
scale_fill_manual(values = viridis_pal(option = "D")(20)[c(16,3,9)]) +
ylab("Bacilli ordinal scale") +
xlab("Time from start treatment (days)") +
theme(legend.position = "none",
  strip.text.x = element_text(
    margin = margin( b = 0, t = 0) )) -> g_pred_overlay_observed
g_pred_overlay_observed

```

## 11.5 Model diagnostics

```

post <- posterior_samples(fit, add_chain = T)

# rhats (chain convergence)
rhats <- rhat(fit)
mcmc_rhat(rhats) + yaxis_text(hjust = 1)

```

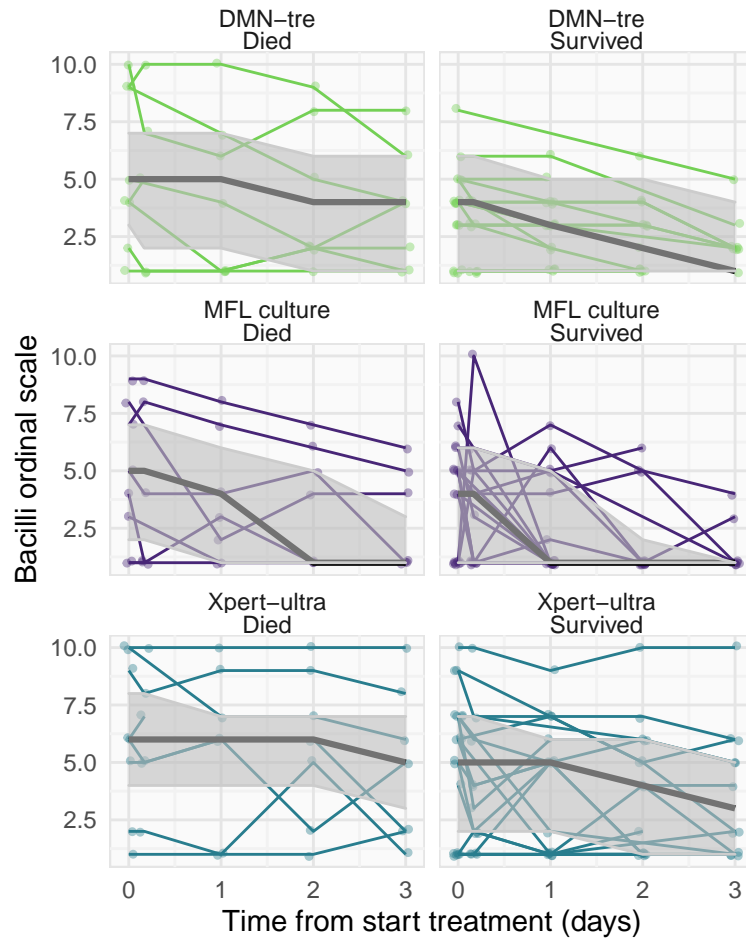


Figure 20: Model predictions for ordinal category by time, assay and outcome status, overlaid on observed data. Black line is median predicted value, shaded area is 50% prediction interval

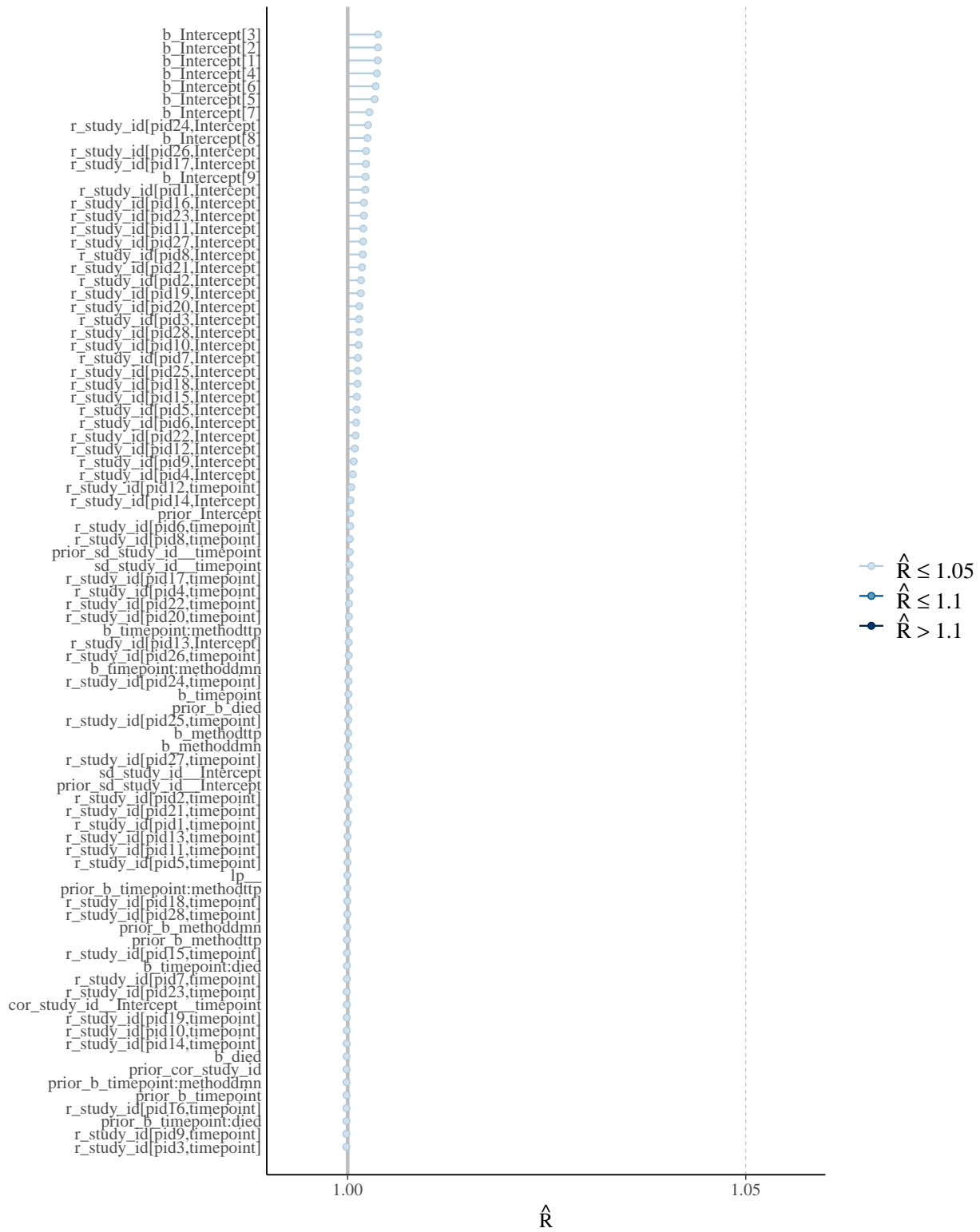
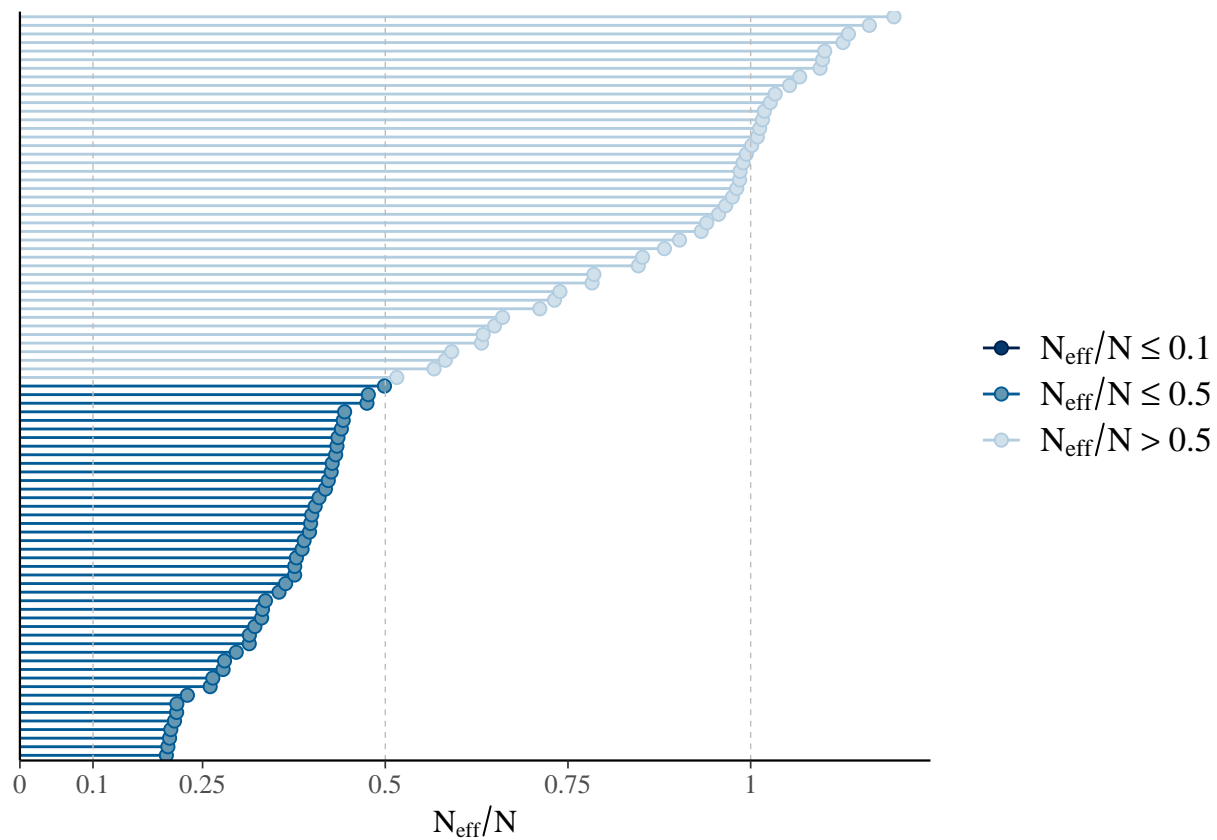


Figure 21: Rhats (chain convergence): none >1.05

```
ratios_cp <- neff_ratio(fit)
mcmc_neff(ratios_cp, size = 2)
```



```
mcmc_plot(fit, type = "trace")
```

```
post %>%
  mcmc_rank_overlay(
    pars = vars(b_timepoint:cor_study_id__Intercept__timepoint)
  ) +
  labs(title = "MCMC chain (trank) plots") +
  coord_cartesian(ylim = c(250, 350))

# autocorrelation
#post %>%
# mcmc_acf(
#   pars = vars(b_timepoint:cor_study_id__Intercept__timepoint),
#   lags = 8) # b_study_id... don't look great - but isn't that what we'd expect here?
```

## 11.6 Checking sensitivity to priors

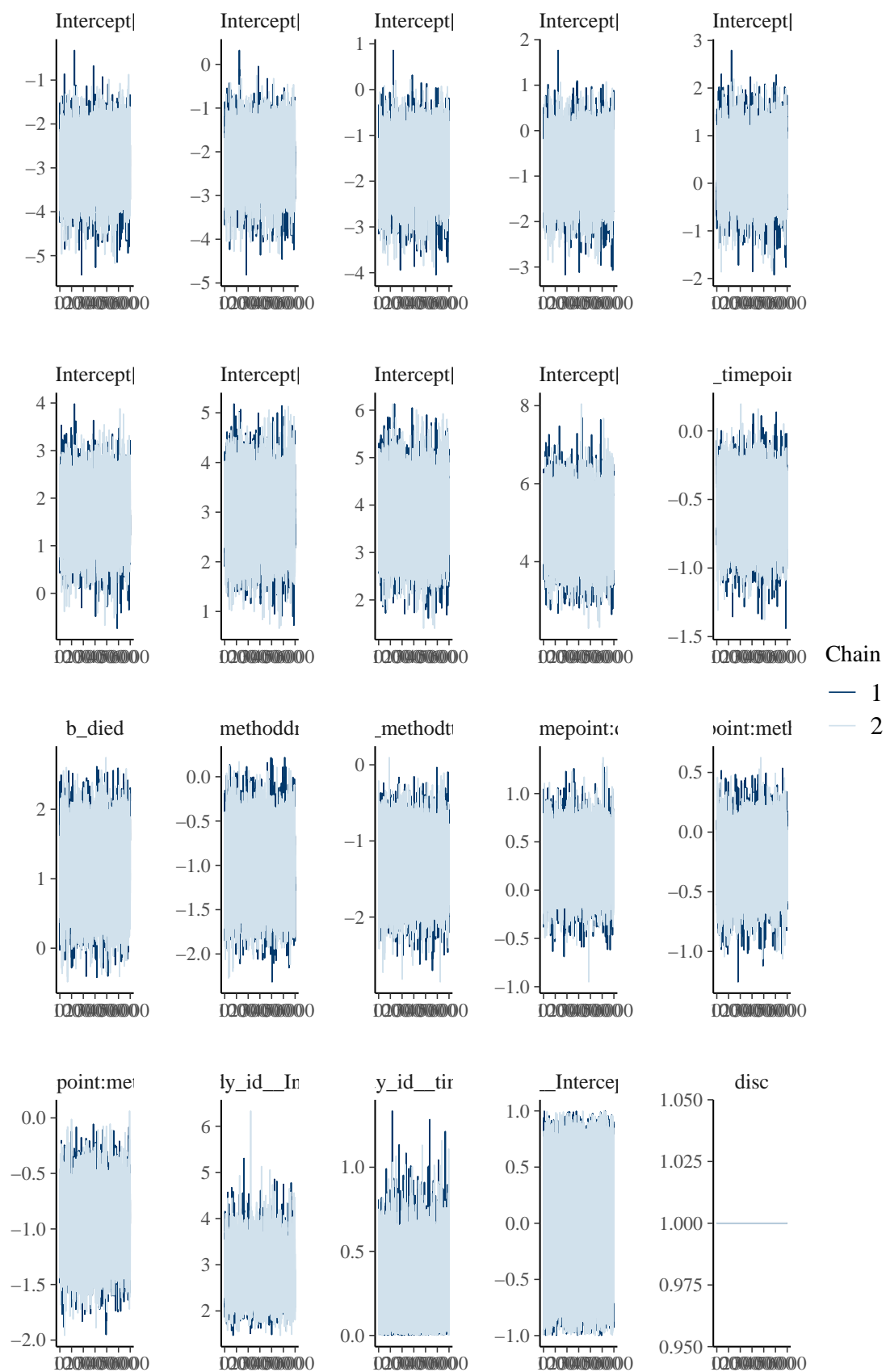


Figure 23: Trace plots (chain convergence): no divergences

# MCMC chain (trank) plots

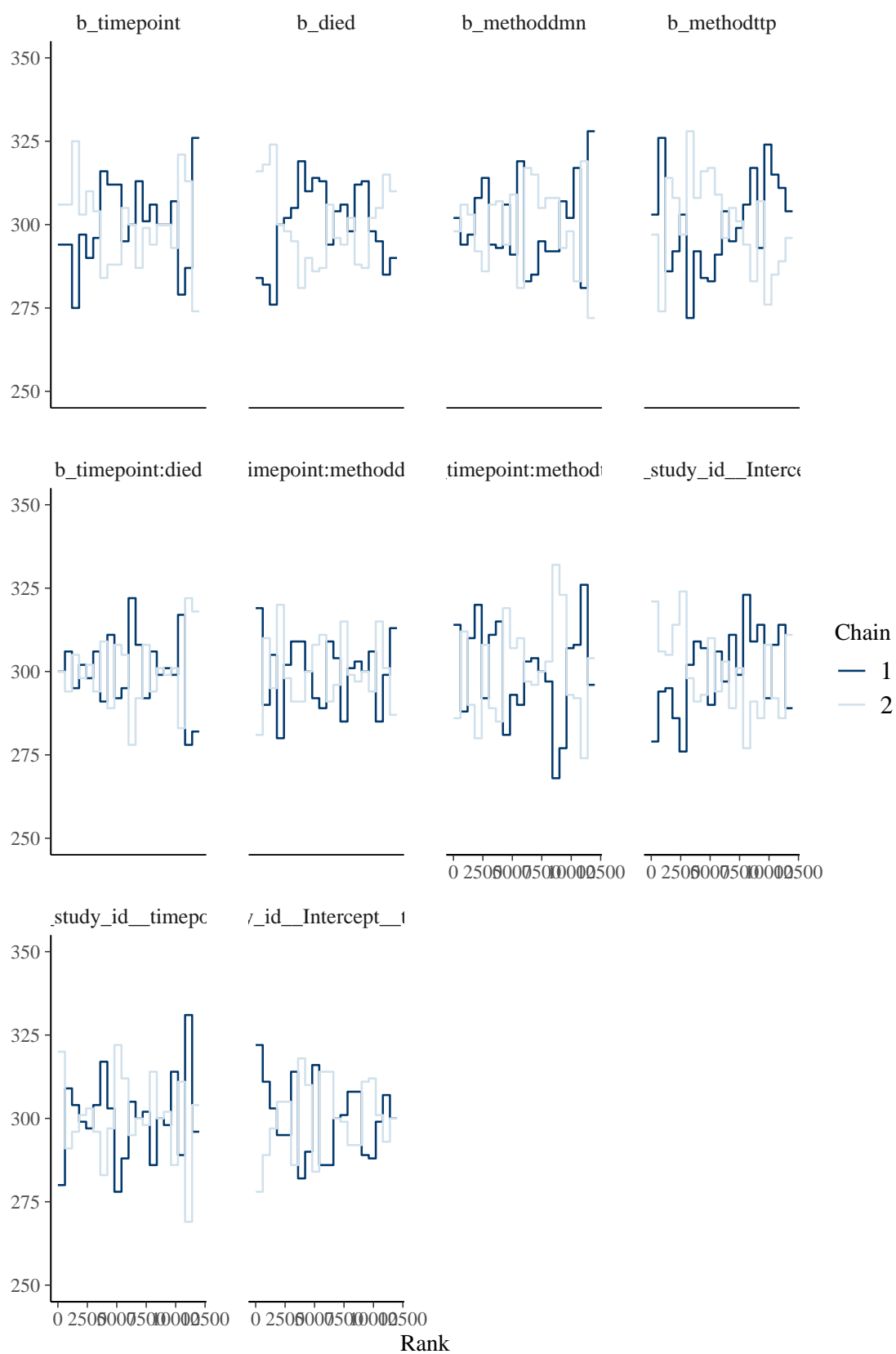


Figure 24: Trank plots (chain convergence): no divergences

```

fit2 <- brm(
  bacilli ~ timepoint + died + timepoint:died + method + timepoint:method +
    (timepoint|study_id),
  data = df_ord,
  family = cumulative("logit"),
  prior = c(
    set_prior("normal(0,2)", class = "b", coef = "died"),
    set_prior("normal(0,2)", class = "b", coef = "timepoint"),
    set_prior("normal(0,2)", class = "b", coef = "timepoint:died"),
    set_prior("normal(0,2)", class = "b", coef = "methoddmn"),
    set_prior("normal(0,2)", class = "b", coef = "methodttp"),
    set_prior("normal(0,2)", class = "b", coef = "timepoint:methoddmn"),
    set_prior("normal(0,2)", class = "b", coef = "timepoint:methodttp"),
    set_prior("student_t(3, 0, 2.5)", class = "sd",
      coef = "Intercept", group = "study_id"),
    set_prior("student_t(3, 0, 2.5)", class = "sd",
      coef = "timepoint", group = "study_id")
  ),
  control = list(adapt_delta = 0.97),
  seed = 140916,
  iter = 8000, warmup = 2000, chains=2,
  cores=4, sample_prior = TRUE
)

bind_rows(
  hypothesis(fit2, hypothesis = "timepoint < 0" )$hypothesis,
  hypothesis(fit2, hypothesis = "died > 0" )$hypothesis,
  hypothesis(fit2, hypothesis = "timepoint:died > 0" )$hypothesis,
  hypothesis(fit2, hypothesis = "methoddmn < 0")$hypothesis,
  hypothesis(fit2, hypothesis = "methodttp < 0")$hypothesis,
  hypothesis(fit2, hypothesis = "timepoint:methoddmn < 0")$hypothesis,
  hypothesis(fit2, hypothesis = "timepoint:methodttp < 0")$hypothesis
) %>%
as_tibble() %>% select(-Star, -Hypothesis) %>%
mutate(
  `Hypothesis statement` = c(
    "Decrease in bacilli load over time on Rx",
    "Patients who died had higher bacilli load",
    "Patients who died had slower decrease in bacilli load over time on Rx",
    "DMN-tre observations of bacilli load were lower than Xpert-ultra",
    "Culture TTP observations of bacilli load were lower than Xpert-ultra",
    "DMN-tre observations of bacilli load declined faster than Xpert-ultra",
    "Culture TTP observations of bacilli load declined faster than Xpert-ultra"
  ),
  `coefficient (beta)` = c(
    "Time in days (slope)",
    "Died (intercept)",
    "Time * Died interaction (slope)",
    "DMN-tre (intercept)",
    "Culture-TTP (intercept)",
    "Time * DMN-tre interaction (slope)",
    "Time * Culture-TTP interaction (slope)"
  )
),

```

```

    h1 = c("<0", ">0", ">0", "<0", "<0", "<0", "<0")
  ) %>%
  select(`coefficient (beta)`, h1, `Hypothesis statement`, everything()) -> h1df

max_iter = 1e3

posterior_samples(fit2, subset = 1:max_iter) %>%
  select(
    b_timepoint, b_died, `b_timepoint:died`,
    b_methoddmn, b_methodttp,
    `b_timepoint:methoddmn`, `b_timepoint:methodttp`
  ) %>%
  pivot_longer(1:7,
    names_to = "coefficient",
    values_to = "posterior_draw") %>%
  mutate(
    coefficient = factor(
      coefficient,
      levels = c("b_timepoint", "b_died", "b_timepoint:died",
        "b_methoddmn", "b_methodttp",
        "b_timepoint:methoddmn", "b_timepoint:methodttp"),
      labels = c(
        "Time in days (slope)",
        "Died (intercept)",
        "Time * Died interaction (slope)",
        "DMN-tre (intercept)",
        "Culture-TTP (intercept)",
        "Time * DMN-tre interaction (slope)",
        "Time * Culture-TTP interaction (slope)"
      )
    )
  ) -> post_samples

annotate_df <- data.frame(
  coefficient = factor(
    c("b_timepoint", "b_died", "b_timepoint:died",
      "b_methoddmn", "b_methodttp",
      "b_timepoint:methoddmn", "b_timepoint:methodttp"),
    levels = c("b_timepoint", "b_died", "b_timepoint:died",
      "b_methoddmn", "b_methodttp",
      "b_timepoint:methoddmn", "b_timepoint:methodttp"),
    labels = c(
      "Time in days (slope)",
      "Died (intercept)",
      "Time * Died interaction (slope)",
      "DMN-tre (intercept)",
      "Culture-TTP (intercept)",
      "Time * DMN-tre interaction (slope)",
      "Time * Culture-TTP interaction (slope)"
    )
  ),
  h1 = c("<0", ">0", ">0", "<0", "<0", "<0", "<0"),
  post_prob = signif(h1df$Post.Prob, 3),
  evid_ratio = formatC(h1df$Evid.Ratio, digits = 2),

```



```

    y1 = 4.5, y2=6, y3=8.5
  )

post_samples %>%
  ggplot(
    aes(fct_rev(coefficient), posterior_draw)
  ) +
  # ggtitle("Posterior probability density\nfor population coefficients") +
  theme_minimal() +
  theme(plot.margin = unit(c(3,3,3,1), "lines"),
        panel.grid = element_blank(),
        plot.title=element_text(size=10,face="bold", hjust=0)) +
  annotation_custom(
    grob = linesGrob(gp = gpar(col="grey90")),
    ymin =-3, ymax=10, xmin = 0.5, xmax=0.5
  ) +
  annotation_custom(
    grob = linesGrob(gp = gpar(col="grey90")),
    ymin =-3, ymax=10, xmin = 1.5, xmax=1.5
  ) +
  annotation_custom(
    grob = linesGrob(gp = gpar(col="grey90")),
    ymin =-3, ymax=10, xmin = 2.5, xmax=2.5
  ) +
  annotation_custom(
    grob = linesGrob(gp = gpar(col="grey90")),
    ymin =-3, ymax=10, xmin = 3.5, xmax=3.5
  ) +
  annotation_custom(
    grob = linesGrob(gp = gpar(col="grey90")),
    ymin =-3, ymax=10, xmin = 4.5, xmax=4.5
  ) +
  annotation_custom(
    grob = linesGrob(gp = gpar(col="grey90")),
    ymin =-3, ymax=10, xmin = 5.5, xmax=5.5
  ) +
  annotation_custom(
    grob = linesGrob(gp = gpar(col="grey90")),
    ymin =-3, ymax=10, xmin = 6.5, xmax=6.5
  ) +
  annotation_custom(
    grob = linesGrob(gp = gpar(col="grey90")),
    ymin =-3, ymax=10, xmin = 7.5, xmax=7.5
  ) +
  annotation_custom(
    grob = linesGrob(),
    ymin =3.5, ymax=10, xmin = 0.4, xmax=0.4
  ) +
  annotation_custom(
    grob = linesGrob(),
    ymin =3.5, ymax=10, xmin = 7.5, xmax=7.5
  ) +

```

```

annotation_custom(
  grob = linesGrob(),
  ymin = 3, ymax = -3, xmin = 7.5, xmax = 7.5
) +
geom_hline(yintercept = 0) +
geom_violin(
  fill = viridis_pal(option = "E")(8)[2],
  alpha = 0.5,
  colour = viridis_pal(option = "E")(8)[2]
) +
geom_boxplot(colour = "black", fill = "white", alpha = 0.7,
  outlier.alpha = 0, width = 0.1) +
ylab("") + xlab("") +
coord_flip(clip = "off") +
annotation_custom(
  grob = textGrob(label = "Higher bacilli load",
    hjust = 0, gp = gpar(fontsize = 10)),
  ymin = 0.3, ymax = 0.3, xmin = -0.5, xmax = -0.5
) +
annotation_custom(
  grob = textGrob(label = "Lower bacilli load",
    hjust = 1, gp = gpar(fontsize = 10)),
  ymin = -0.3, ymax = -0.3, xmin = -0.5, xmax = -0.5
) +
annotation_custom(
  grob = textGrob(label = "Estimate for coefficient (logit scale)",
    gp = gpar(fontsize = 10, fontface = "bold")),
  ymin = 0, ymax = 0, xmin = -1.5, xmax = -1.5
) +
annotation_custom(
  grob = linesGrob(
    arrow = arrow(type = "closed",
      ends = "last",
      length = unit(.2, "cm"))),
  ymin = 0.1, ymax = 3, xmin = 0.4, xmax = 0.4
) +
annotation_custom(
  grob = linesGrob(
    arrow = arrow(type = "closed",
      ends = "first",
      length = unit(.2, "cm"))),
  ymin = -0.1, ymax = -3, xmin = 0.4, xmax = 0.4
) +
geom_text(
  data = annotate_df,
  aes(x = coefficient, y = y1, label = h1),
  hjust = 0, size = 3.5
) +
geom_text(
  data = annotate_df,
  aes(x = coefficient, y = y2, label = post_prob),
  hjust = 0, size = 3.5
) +

```

```

geom_text(
  data = annotate_df,
  aes(x = coefficient, y=y3, label = evid_ratio),
  hjust = 0, size=3.5
) + scale_y_continuous(breaks=c(-2,-1,0, 1, 2)) +
geom_text(label = "Hypothesis", x=7.75, y=4, size=3.5,
          hjust=0.2, vjust=0) +
geom_text(label = "Posterior\nprobability", x=7.75, y=6, size=3.5,
          hjust=0, vjust=0) +
geom_text(label = "Evidence\nratio", x=7.75, y=8.5, size=3.5,
          hjust=0, vjust=0) +
geom_text(
  label = "Posterior probability density\nfor population coefficients",
  x=7.75, y=0, size=3.5,
  hjust=0.5, vjust=0) -> g_coef2

```

## 12 Serial observations

(not included)

## 13 Bacilli lengths

```

files <- list.files(path = paste0(getwd(), "/bacilli_lengths"),
                    pattern = ".csv",
                    recursive = TRUE, full.names = TRUE)

bldf <- data.frame()

for(i in 1:length(files)){

  path <- files[i]
  temp_dat <- read.csv(path)
  temp_dat$id <- path

  bldf <- rbind(bldf, temp_dat)

}
rm(temp_dat)

bldf %>% as_tibble() %>%
  mutate(
    id =
      str_replace(
        id,
        pattern = "C:/Users/David/OneDrive/PhD/manuscripts/serialsubstudy/wd/bacilli_lengths/",
        replacement = ""),
    id =
      str_replace(
        id,

```

```

    pattern = ".csv",
    replacement = ""),
    id_list = str_split(id, pattern = "/"),
    study_id = map_chr(.x=id_list, 1),
    timepoint = as.numeric(map_chr(.x=id_list, 2))/24
  ) %>%
  select(study_id, timepoint, id, X, Y) %>%
  dplyr::group_by(study_id, timepoint, id) %>%
  dplyr::summarise(length = max(X),
                    log_length = log(length)) %>%
  filter(length < 12.5) -> bldf

bldf %>%
  ggplot(aes(length)) +
  geom_histogram(fill=viridis(8, option="E")[2],
                 colour="black", alpha=0.6,
                 binwidth = 0.2) +
  theme_dab() +
  xlab(paste0("Bacilli length (", "\u03bc", "m)")) -> g_length_hist

# priors are based on Vijay et al. Front Microbiol 2017 https://doi.org/10.3389/fmicb.2017.02296
prior_dist = rnorm(1e4,
                  mean = rnorm(1e4, 1.05, 0.175), # mean should be between 2 and 4 (non log scale)
                  sd = rnorm(1e4, 0.43, 0.06))
# hist(exp(prior_dist))

m_lengths <- brm(
  log_length ~ timepoint + (timepoint|study_id),
  data=bldf,
  prior = c(
    set_prior("normal(1.05,0.175)", class = "Intercept"),
    set_prior("normal(0.43,0.06)", class = "sigma")),
  control = list(adapt_delta = 0.97),
  seed = 140916,
  cores=4
)

max_iter = 1e3
nd = data.frame(
  study_id = rep(unique(bldf$study_id), each = 5),
  timepoint = rep(c(0,1/6,1,2,3), 10)
)

posterior_predict(
  m_lengths,
  newdata = nd,
  subset = 1:max_iter,
  seed = 140916) %>%
  as_tibble() %>%
  pivot_longer(1:nrow(nd)) %>%
  mutate(
    study_id = rep(nd$study_id, max_iter),

```

```

    timepoint = rep(nd$timepoint, max_iter),
    iter = rep(1:max_iter, each = nrow(nd)),
    length_pred = exp(value)
  ) %>%
  select(iter, study_id, timepoint, length_pred) %>%
  ggplot(aes(as.factor(round(timepoint,2)), length_pred)) +
  geom_quasirandom(
    data=bldf,
    aes(x=as.factor(round(timepoint,2)), y=length),
    alpha=0.7, size = 0.5, colour = viridis(8, option="E")[2]
  ) +
  geom_violin(alpha=0.2) +
  facet_wrap(~study_id, nrow=2) +
  theme_dab() +
  theme(panel.spacing = unit(0.2, "lines"),
        axis.text.x = element_text(size=7, angle=90, hjust=1, vjust=0.5)) +
  ylab(paste0("Bacilli length (", "\u03bc", "m)")) +
  xlab("Time from start treatment (days)") -> g_length_fit_data

max_iter = 4e3
nd = data.frame(
  study_id = "pidx",
  timepoint = seq(0,3, length.out = 20)
)

posterior_predict(
  m_lengths,
  newdata = nd,
  subset = 1:max_iter,
  re_formula = log_length ~ timepoint + (timepoint|study_id),
  allow_new_levels = TRUE,
  sample_new_levels = "gaussian",
  seed = 140916
) %>%
as_tibble() %>%
pivot_longer(1:nrow(nd)) %>%
mutate(
  timepoint = rep(nd$timepoint, max_iter),
  iter = rep(1:max_iter, each = nrow(nd)),
  length_pred = exp(value)
) %>%
select(iter, timepoint, length_pred) %>%
group_by(timepoint) %>%
summarise(
  pred_length = median(length_pred),
  pred_q0.025 = quantile(length_pred, probs = 0.025),
  pred_q0.975 = quantile(length_pred, probs = 0.975)
) %>%
left_join(

  fitted(
    m_lengths, newdata = nd, summary = F,

```

```

re_formula = NA,
subset = 1:max_iter,
seed = 140916) %>%
as_tibble() %>%
pivot_longer(1:nrow(nd)) %>%
mutate(
  timepoint = rep(nd$timepoint, max_iter),
  iter = rep(1:max_iter, each = nrow(nd)),
  length_pred = exp(value)
) %>%
select(iter, timepoint, length_pred) %>%
group_by(timepoint) %>%
summarise(
  fit_length = median(length_pred),
  fit_q0.025 = quantile(length_pred, probs = 0.025),
  fit_q0.975 = quantile(length_pred, probs = 0.975)
),
by = "timepoint"
) -> m_lengths_fit_pred

m_lengths_fit_pred %>%
ggplot(
  aes(x=timepoint)
) +
geom_ribbon(
  aes(ymin=pred_q0.025, ymax=pred_q0.975),
  alpha=0.1, linetype=2,
  fill=viridis(8, option="E")[3],
  colour=viridis(8, option="E")[2]
) +
geom_ribbon(
  aes(ymin=fit_q0.025, ymax=fit_q0.975),
  alpha=0.7,
  fill=viridis(8, option="E")[3],
  colour=viridis(8, option="E")[2]
) +
geom_line(
  aes(y=fit_length)
) +
theme_dab() +
ylab(paste0("Predicted\nlength (", "\u03bc", "m)")) +
xlab("Time from start\ntreatment (days)") -> g_lengths_fit_pred

(g_length_hist | g_lengths_fit_pred) / g_length_fit_data

```

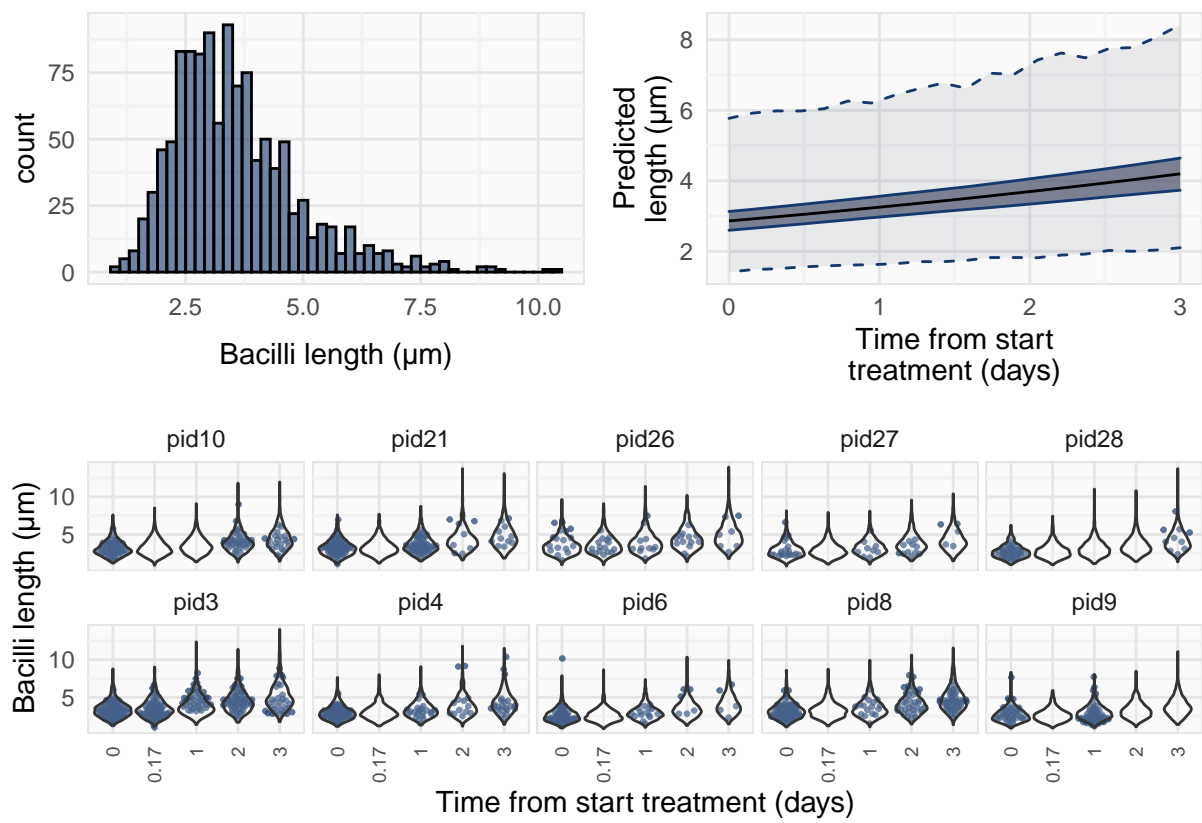


Figure 25: Bacilli length variation with time on treatment