

Antibody dynamics for Cholera

Samuel Schildhauer

2024-12-12

Running full model, unstratified with the following inputs:

```
nchains <- 4; # nr of MC chains to run simultaneously
nadapt <- 10000; # nr of iterations for adaptation
nburnin <- 100; # nr of iterations to use for burn-in
nmc <- 2500; # nr of samples in posterior chains
niter <- 5000; # nr of iterations for posterior sample
nthin <- round(niter/nmc); # thinning needed to produce nmc from niter
```

```
## Standardizing OD values
# dL$result <- standardize(dL$result)

# Filtering down to two
dL <- dL %>%
  filter(antigen_iso %in% c("lpsol39o_IgA"))

#set seed for reproducibility
set.seed(54321)
#subset data for checking
# dL_sub <- dL %>%
#   filter(index_id %in% sample(unique(index_id), 50))

#prepare data for modeline
longdata <- prep_data(dL)
priors <- prep_priors(max_antigens = longdata$n_antigen_isos)

#inputs for jags model
nchains <- 4; # nr of MC chains to run simultaneously
```

```

nadapt  <- 10000;          # nr of iterations for adaptation
nburnin <- 100;           # nr of iterations to use for burn-in
nmc     <- 2500;          # nr of samples in posterior chains
niter   <- 5000;          # nr of iterations for posterior sample
nthin   <- round(niter/nmc); # thinning needed to produce nmc from niter

#pred.subj <- longdata$nsubj + 1;
#tomonitor <- c("par");
tomonitor <- c("y0", "y1", "t1", "alpha", "shape");

#This handles the seed to reproduce the results
initsfunction <- function(chain){
  stopifnot(chain %in% (1:4)); # max 4 chains allowed...
  .RNG.seed <- (1:4)[chain];
  .RNG.name <- c("base::Wichmann-Hill", "base::Marsaglia-Multicarry",
                "base::Super-Duper", "base::Mersenne-Twister")[chain];
  return(list(".RNG.seed"=.RNG.seed, ".RNG.name"=.RNG.name));
}

jags.post <- run.jags(model=file.mod, data=c(longdata, priors),
                     inits=initsfunction, method="parallel",
                     adapt=nadapt, burnin=nburnin, thin=nthin, sample=nmc,
                     n.chains=nchains,
                     monitor=tomonitor, summarise=FALSE);

```

```

Calling 4 simulations using the parallel method...
Following the progress of chain 1 (the program will wait for all chains
to finish before continuing):
Welcome to JAGS 4.3.2 (official binary) on Thu Dec 12 13:30:27 2024
JAGS is free software and comes with ABSOLUTELY NO WARRANTY
Loading module: basemod: ok
Loading module: bugs: ok
. . Reading data file data.txt
. Compiling model graph
  Resolving undeclared variables
  Allocating nodes
Graph information:
  Observed stochastic nodes: 636
  Unobserved stochastic nodes: 165
  Total graph size: 11055
. Reading parameter file inits1.txt
. Initializing model

```

```

. Adapting 10000
-----| 10000
+++++ 100%
Adaptation successful
. Updating 100
-----| 100
***** 100%
. . . . . Updating 5000
-----| 5000
***** 100%
. . . . . Updating 0
. Deleting model
.
All chains have finished
Simulation complete. Reading coda files...
Coda files loaded successfully
Finished running the simulation

```

```

mcmc_list <- as.mcmc.list(jags.post)
mcmc_df <- ggs(mcmc_list)

wide_predpar_df <- mcmc_df %>%
  mutate(
    parameter = sub("^((\\w+)\\.[.]*", "\\1", Parameter),
    index_id = as.numeric(sub("^\\w+\\[(\\d+),.]*", "\\1", Parameter)),
    antigen_iso = as.numeric(sub("^\\w+\\[(\\d+,(\\d+).)*", "\\1", Parameter))
  ) %>%
  mutate(
    index_id = factor(index_id, labels = attr(longdata, "ids")),
    antigen_iso = factor(antigen_iso, labels = attr(longdata, "antigens"))) %>%
  # mutate(value = exp(value)) %>%
  # mutate(value = ifelse(parameter == "r", value+1, value)) %>%
  ## only take the last subject (newperson)
  filter(index_id == "newperson") %>%
  select(-Parameter) %>%
  pivot_wider(names_from = "parameter", values_from="value") %>%
  rowwise() %>%
  #mutate(y1 = y0+y1) %>%
  droplevels() %>%
  ungroup() %>%
  rename(r = shape)

```

```
#Now plot longitudinal antibody decay
curve_params <- wide_predpar_df
  class(curve_params) =
  c("curve_params", class(curve_params))
  antigen_isos = unique(curve_params$antigen_iso)
  attr(curve_params, "antigen_isos") = antigen_isos
```

Diagnostics of full model

Note: Program is recommending additional adaptations – unsure how high to go. It sounds like this may be due to a complex model

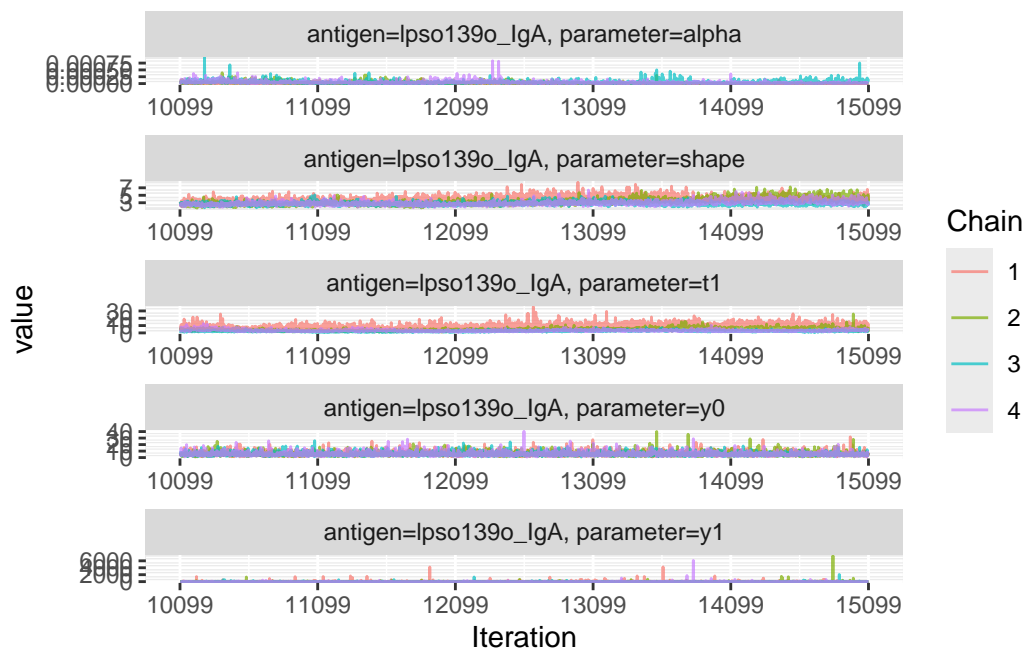
Summary

```
EffSize <- data.frame(effectiveSize(jags.post$mcmc))
EffSize <- rownames_to_column(EffSize)
EffSize <- EffSize %>%
  filter(grepl("159", rowname))
EffSize
```

	rowname	effectiveSize.jags.post.mcmc.
1	y0[159,1]	9271.50879
2	y1[159,1]	9159.31627
3	t1[159,1]	85.80553
4	alpha[159,1]	475.29022
5	shape[159,1]	270.26088

Traceplots

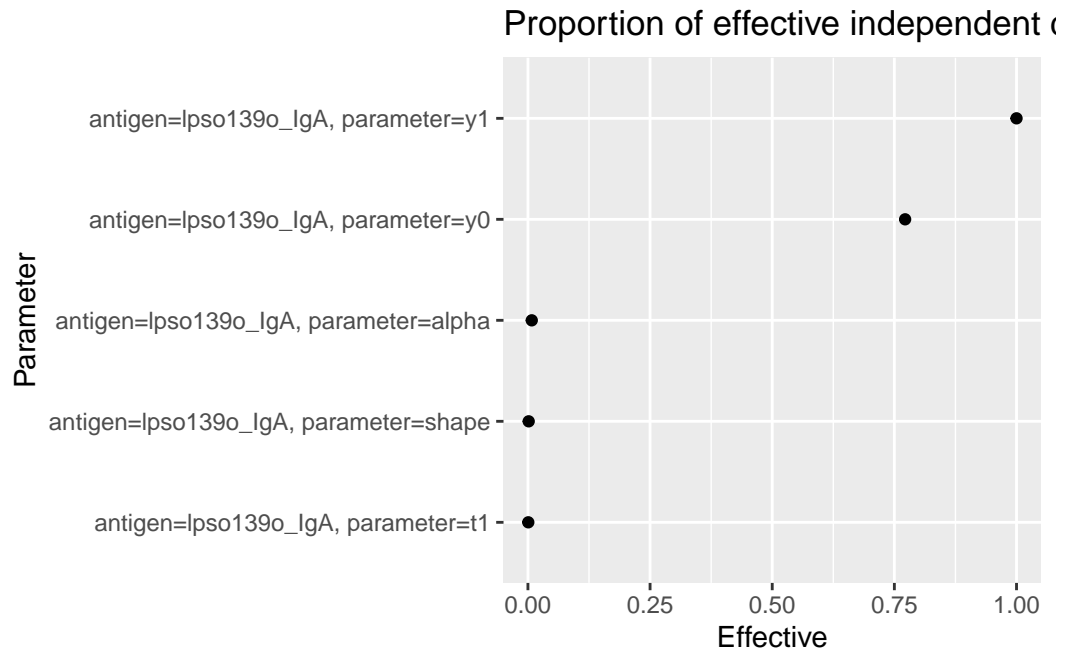
```
plot_jags_trace(data=jags.post, iso="lpso139o_IgA")
```



```
# posterior <- coda.samples(jags.post)
# effectiveSize(jags.post)
```

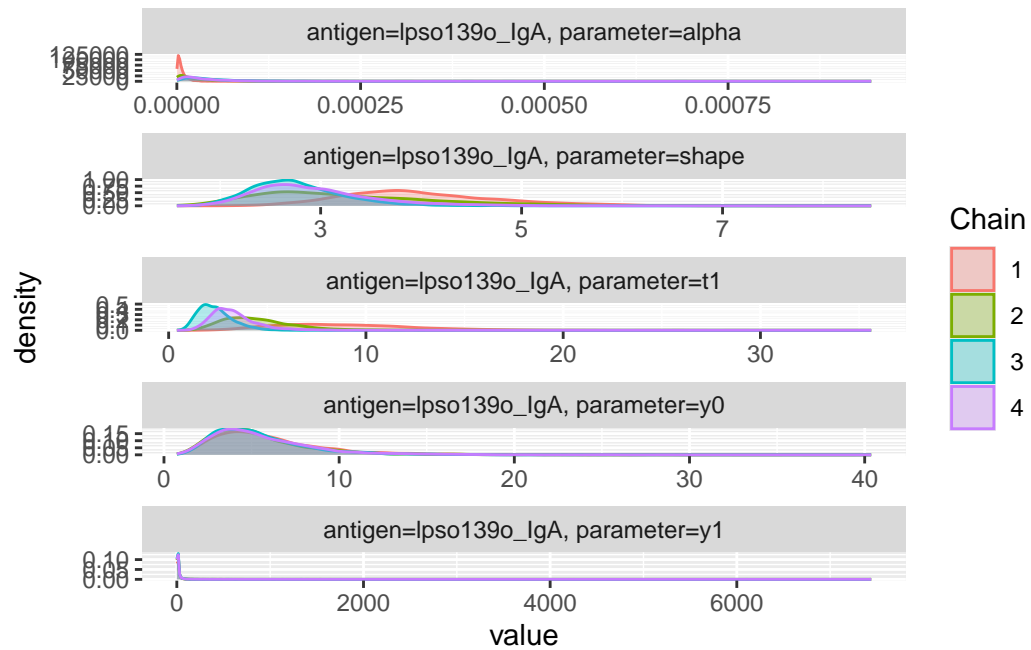
Number of effective samples

```
plot_jags_eff(data=jags.post, iso="lpso139o_IgA")
```



Density plots

```
plot_jags_dens(data=jags.post, iso="lpso139o_IgA")
```



Running stratified data

```
#Creating a function to run stratified data
Run_jags_fun <- function(data, strat) {
  strat_list <- unique(data[[strat]])

  for (i in strat_list) {
    dL_sub <- data %>%
      filter(data[[strat]]==i)
    # dL_sub <- dL %>%
    #   filter(Age_immune_combo=="Case_5under")

    #set seed for reproducibility
    set.seed(54321)
    #subset data for checking
    # dL_sub <- dL %>%
    #   filter(index_id %in% sample(unique(index_id), 50))

    #prepare data for modeline
    longdata <- prep_data(dL_sub)
    priors <- prep_priors(max_antigens = longdata$n_antigen_isos)

    #inputs for jags model
    nchains <- 4;                # nr of MC chains to run simultaneously
    nadapt <- 10000;             # nr of iterations for adaptation
    nburnin <- 100;              # nr of iterations to use for burn-in
    nmc <- 2500;                 # nr of samples in posterior chains
    niter <- 5000;               # nr of iterations for posterior sample
    nthin <- round(niter/nmc); # thinning needed to produce nmc from niter

    #pred.subj <- longdata$nsubj + 1;
    #tomonitor <- c("par");
    tomonitor <- c("y0", "y1", "t1", "alpha", "shape");

    #This handles the seed to reproduce the results
    initsfunction <- function(chain){
      stopifnot(chain %in% (1:4)); # max 4 chains allowed...
      .RNG.seed <- (1:4)[chain];
      .RNG.name <- c("base::Wichmann-Hill", "base::Marsaglia-Multicarry",
                    "base::Super-Duper", "base::Mersenne-Twister")[chain];
      return(list(".RNG.seed"=.RNG.seed, ".RNG.name"=.RNG.name));
    }
  }
}
```

```

jags.post <- run.jags(model=file.mod,data=c(longdata, priors),
                    inits=initsfunction,method="parallel",
                    adapt=nadapt,burnin=nburnin,thin=nthin,sample=nmc,
                    n.chains=nchains,
                    monitor=tomonitor,summarise=FALSE);
mcmc_list <- as.mcmc.list(jags.post)
mcmc_df <- ggs(mcmc_list)

wide_predpar_df <- mcmc_df %>%
  mutate(
    parameter = sub("^((\\w+)\\[\\.\\*]", "\\1", Parameter),
    index_id = as.numeric(sub("^\\w+\\[\\((\\d+),\\.\\*", "\\1", Parameter)),
    antigen_iso = as.numeric(sub("^\\w+\\[\\((\\d+,(\\d+).\\*", "\\1", Parameter))
  ) %>%
  mutate(
    index_id = factor(index_id, labels = attr(longdata, "ids")),
    antigen_iso = factor(antigen_iso, labels = attr(longdata, "antigens"))) %>%
# mutate(value = exp(value)) %>%
# mutate(value = ifelse(parameter == "r", value+1, value)) %>%
## only take the last subject (newperson)
filter(index_id == "newperson") %>%
select(-Parameter) %>%
pivot_wider(names_from = "parameter", values_from="value") %>%
rowwise() %>%
#mutate(y1 = y0+y1) %>%
droplevels() %>%
ungroup() %>%
rename(r = shape) %>%
mutate(stratification=i)

assign(paste0("wide_predpar_",i), data.frame(wide_predpar_df), envir=globalenv())

# wide_prepdar_df_shell <- rbind(wide_predpar_df_shell, wide_predpar_df)
}
}

## Running function
Run_jags_fun(dL, "Age_immune_combo")

```

Calling 4 simulations using the parallel method...

Following the progress of chain 1 (the program will wait for all chains to finish before continuing):


```

Welcome to JAGS 4.3.2 (official binary) on Thu Dec 12 13:31:38 2024
JAGS is free software and comes with ABSOLUTELY NO WARRANTY
Loading module: basemod: ok
Loading module: bugs: ok
. . Reading data file data.txt
. Compiling model graph
  Resolving undeclared variables
  Allocating nodes
Graph information:
  Observed stochastic nodes: 188
  Unobserved stochastic nodes: 57
  Total graph size: 3431
. Reading parameter file inits1.txt
. Initializing model
. Adapting 10000
-----| 10000
+++++ 100%
Adaptation successful
. Updating 100
-----| 100
***** 100%
. . . . . Updating 5000
-----| 5000
***** 100%
. . . . . Updating 0
. Deleting model
.
All chains have finished

Warning: The adaptation phase of one or more models was not completed in 10000
iterations, so the current samples may not be optimal - try increasing the
number of iterations to the "adapt" argument

Simulation complete. Reading coda files...
Coda files loaded successfully
Finished running the simulation
Calling 4 simulations using the parallel method...
Following the progress of chain 1 (the program will wait for all chains
to finish before continuing):
Welcome to JAGS 4.3.2 (official binary) on Thu Dec 12 13:31:48 2024
JAGS is free software and comes with ABSOLUTELY NO WARRANTY
Loading module: basemod: ok

```

```

Loading module: bugs: ok
. . Reading data file data.txt
. Compiling model graph
  Resolving undeclared variables
  Allocating nodes
Graph information:
  Observed stochastic nodes: 93
  Unobserved stochastic nodes: 36
  Total graph size: 1850
. Reading parameter file inits1.txt
. Initializing model
. Adapting 10000
-----| 10000
+++++++ 100%
Adaptation successful
. Updating 100
-----| 100
***** 100%
. . . . . Updating 5000
-----| 5000
***** 100%
. . . . Updating 0
. Deleting model
.
All chains have finished
Simulation complete. Reading coda files...
Coda files loaded successfully
Finished running the simulation
Calling 4 simulations using the parallel method...
Following the progress of chain 1 (the program will wait for all chains
to finish before continuing):
Welcome to JAGS 4.3.2 (official binary) on Thu Dec 12 13:31:55 2024
JAGS is free software and comes with ABSOLUTELY NO WARRANTY
Loading module: basemod: ok
Loading module: bugs: ok
. . Reading data file data.txt
. Compiling model graph
  Resolving undeclared variables
  Allocating nodes
Graph information:
  Observed stochastic nodes: 230
  Unobserved stochastic nodes: 57
  Total graph size: 3915

```

```

. Reading parameter file inits1.txt
. Initializing model
. Adapting 10000
-----| 10000
+++++ 100%
Adaptation successful
. Updating 100
-----| 100
***** 100%
. . . . . Updating 5000
-----| 5000
***** 100%
. . . . Updating 0
. Deleting model
.
All chains have finished

```

Warning: The adaptation phase of one or more models was not completed in 10000 iterations, so the current samples may not be optimal - try increasing the number of iterations to the "adapt" argument

```

Simulation complete. Reading coda files...
Coda files loaded successfully
Finished running the simulation
Calling 4 simulations using the parallel method...
Following the progress of chain 1 (the program will wait for all chains
to finish before continuing):
Welcome to JAGS 4.3.2 (official binary) on Thu Dec 12 13:32:05 2024
JAGS is free software and comes with ABSOLUTELY NO WARRANTY
Loading module: basemod: ok
Loading module: bugs: ok
. . Reading data file data.txt
. Compiling model graph
  Resolving undeclared variables
  Allocating nodes
Graph information:
  Observed stochastic nodes: 125
  Unobserved stochastic nodes: 36
  Total graph size: 2234
. Reading parameter file inits1.txt
. Initializing model
. Adapting 10000

```

```

-----| 10000
+++++ 100%
Adaptation incomplete.
. Updating 100
-----| 100
***** 100%
. . . . . Updating 5000
-----| 5000
***** 100%
. . . . Updating 0
. Deleting model
.
All chains have finished

```

Warning: The adaptation phase of one or more models was not completed in 10000 iterations, so the current samples may not be optimal - try increasing the number of iterations to the "adapt" argument

Simulation complete. Reading coda files...
Coda files loaded successfully
Finished running the simulation

```

# Putting dataframes back together
wide_predpar_final <- rbind(wide_predpar_Case_5under, wide_predpar_Case_over5, wide_predpar_V

#Now plot longitudinal antibody decay
curve_params <- wide_predpar_final
class(curve_params) =
c("curve_params", class(curve_params))
antigen_isos = unique(curve_params$antigen_iso)
attr(curve_params, "antigen_isos") = antigen_isos

```

Methods: Cholera data was run through jags simulation using the following settings: inputs for jags model:

```

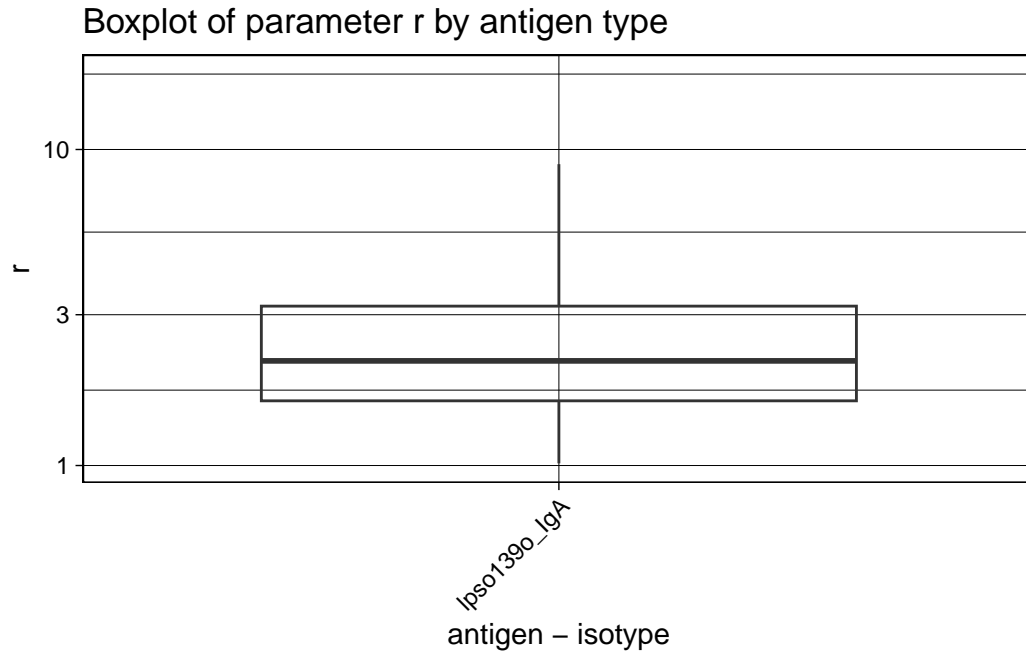
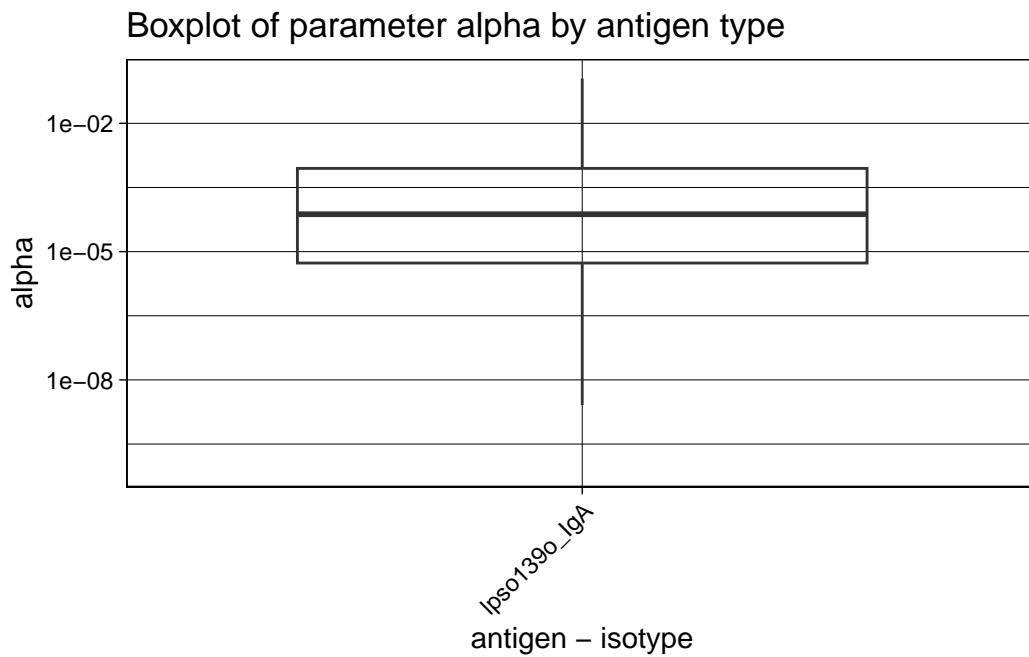
nchains <- 4; # nr of MC chains to run simultaneously
nadapt <- 10000; # nr of iterations for adaptation
nburnin <- 100; # nr of iterations to use for burn-in
nmc <- 2500; # nr of samples in posterior chains
niter <- 2500; # nr of iterations for posterior sample

```

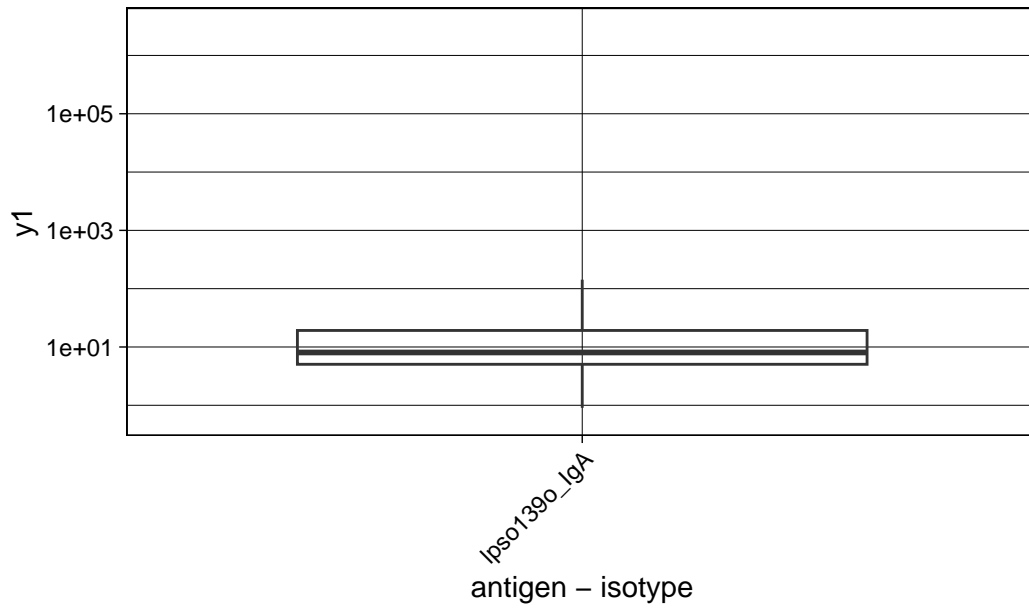
```
nthin <- round(niter/nmc); # thinning needed to produce nmc from niter
```

Data from the “New person” were subset and used for the following figures.

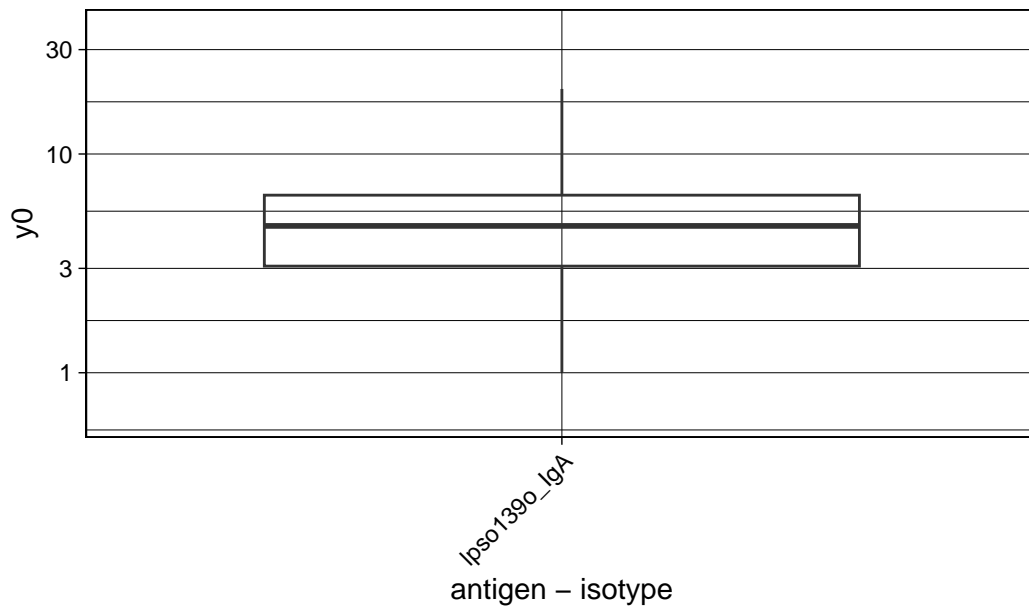
Parameter Values by Antigen Type

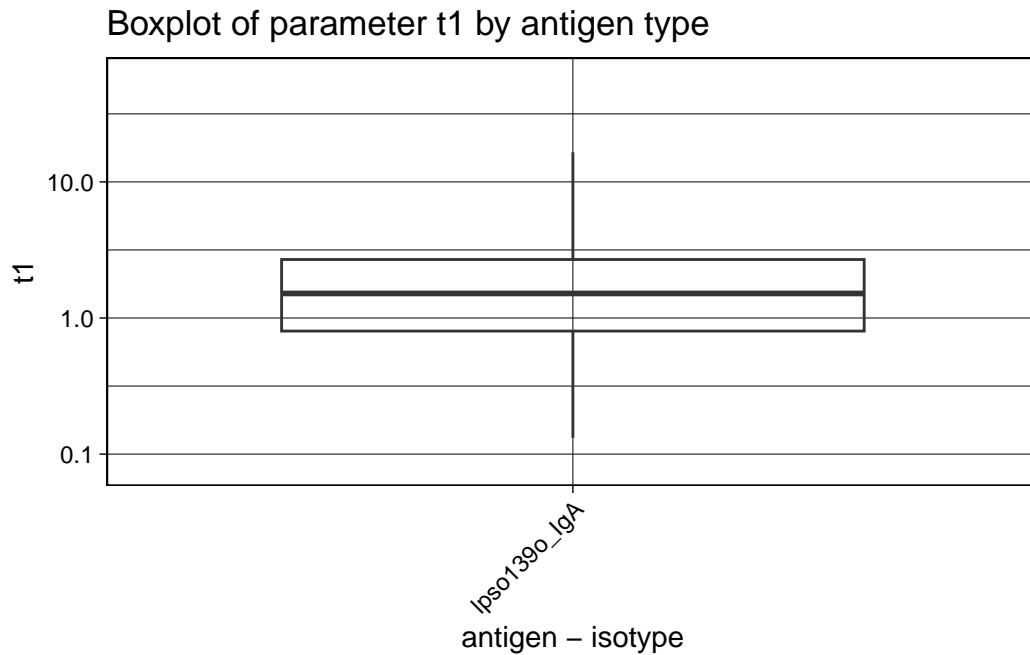


Boxplot of parameter y1 by antigen type



Boxplot of parameter y0 by antigen type



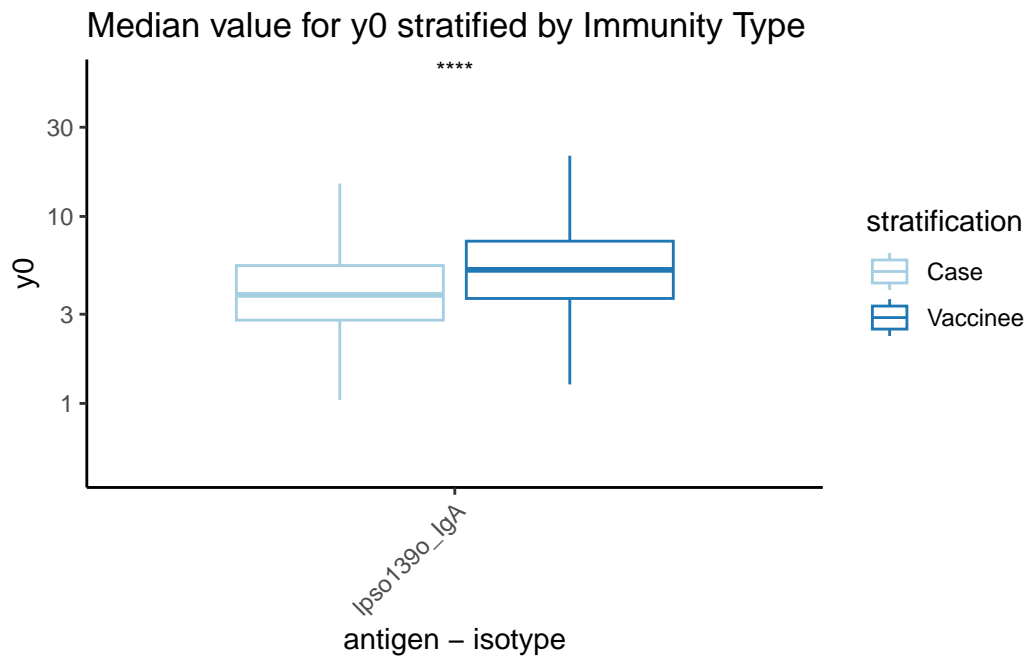
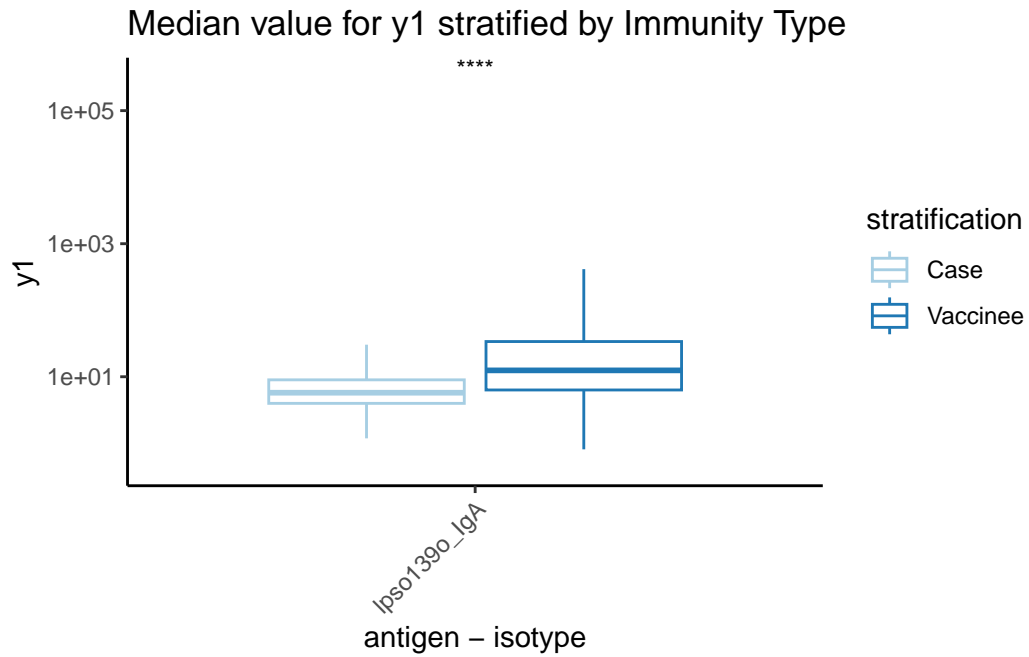


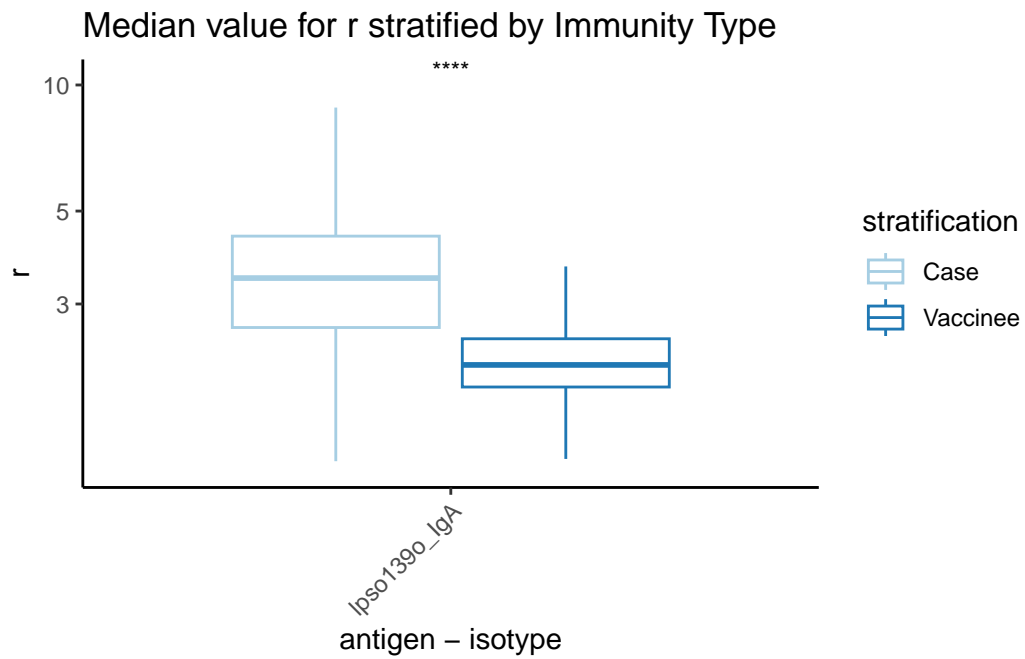
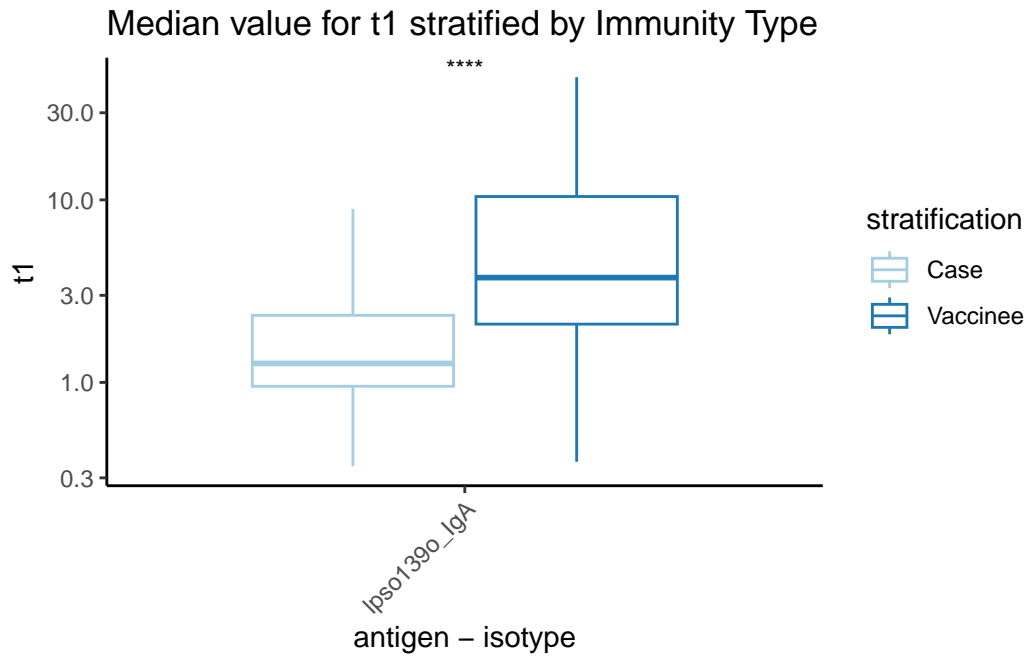
Methods: Cholera data set was stratified by case vs vaccinee and

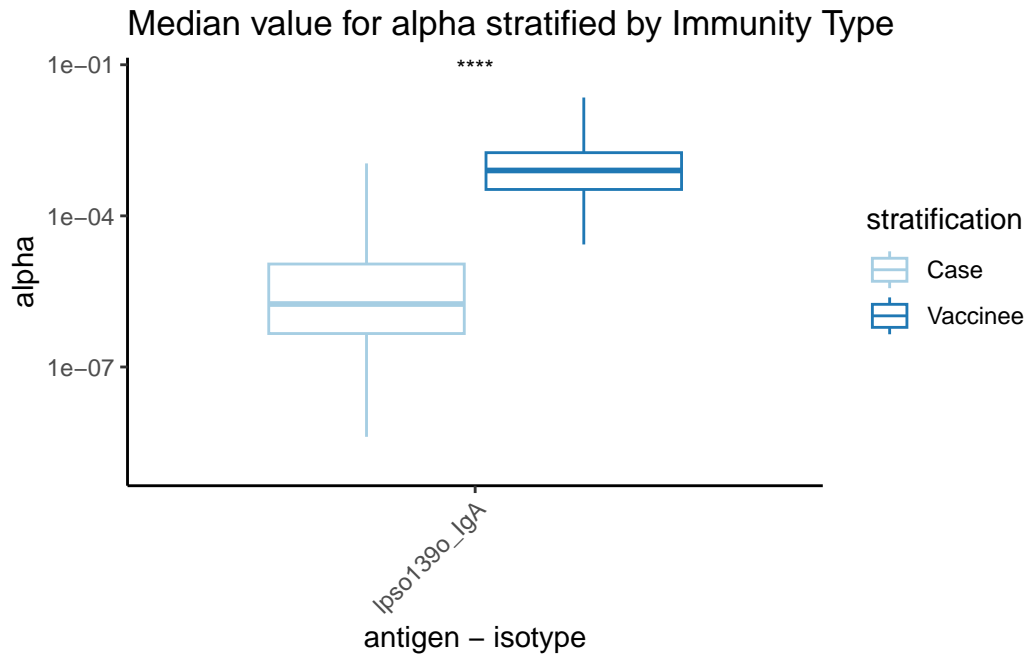
$$\leq 5$$

vs >5 years of age and run through jags simulation using the following inputs: inputs for jags model nchains <- 4; # nr of MC chains to run simultaneously nadapt <- 2000; # nr of iterations for adaptation nburnin <- 100; # nr of iterations to use for burn-in nmc <- 100; # nr of samples in posterior chains niter <- 1000; # nr of iterations for posterior sample nthin <- round(niter/nmc); # thinning needed to produce nmc from niter

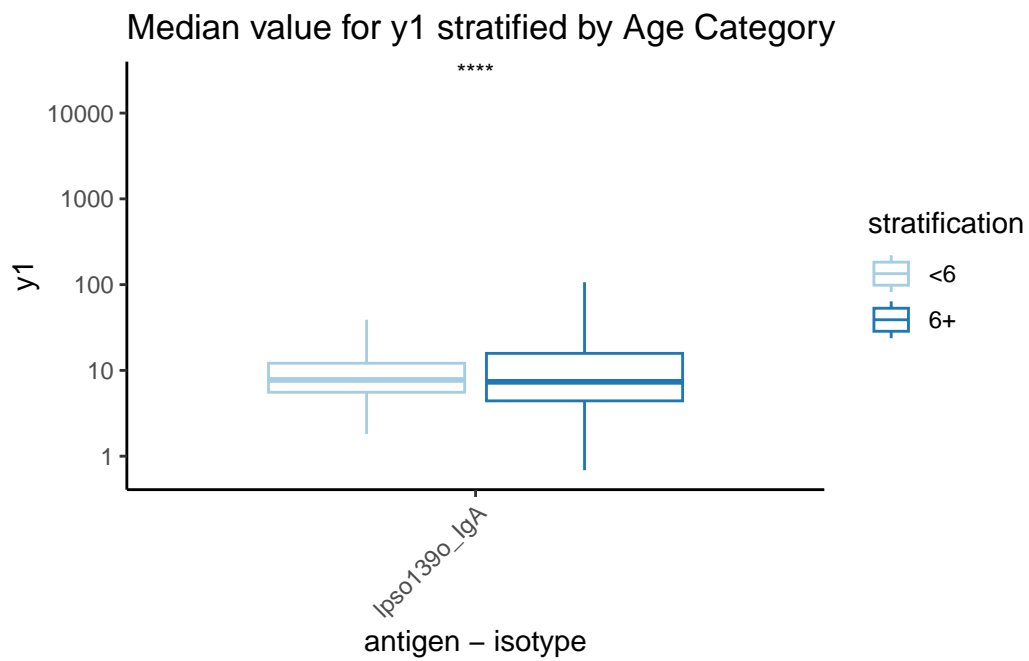
Antibody dynamic plots stratified by immunity status

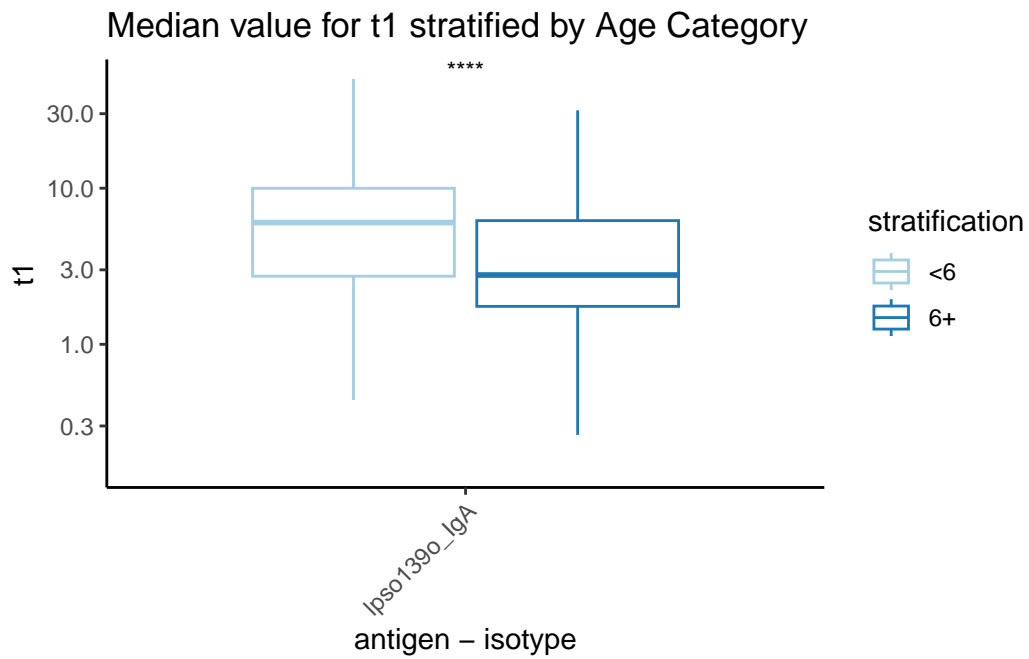
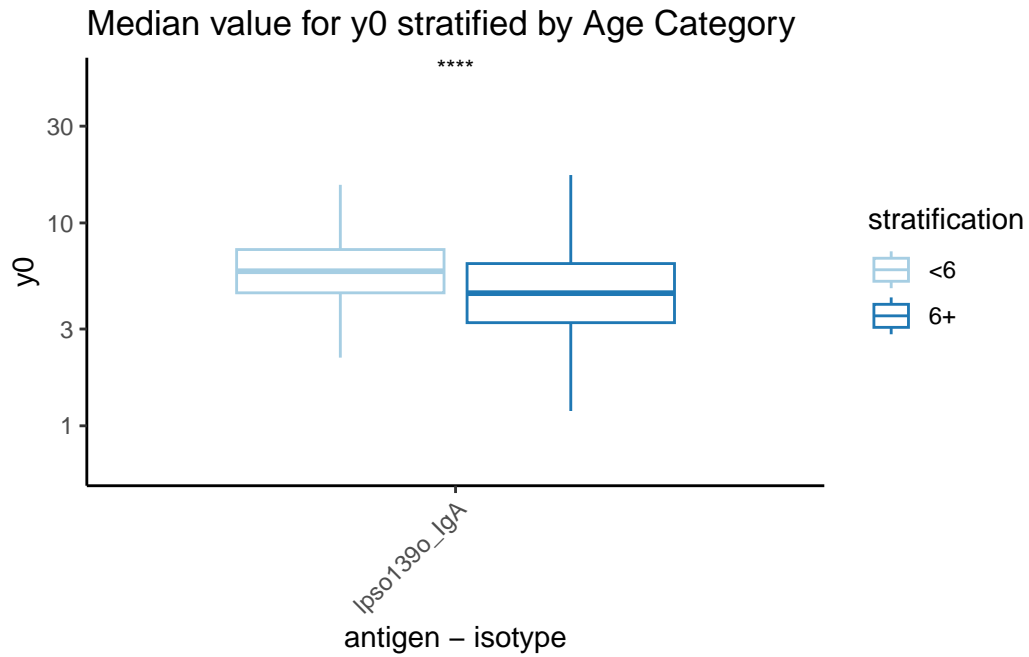


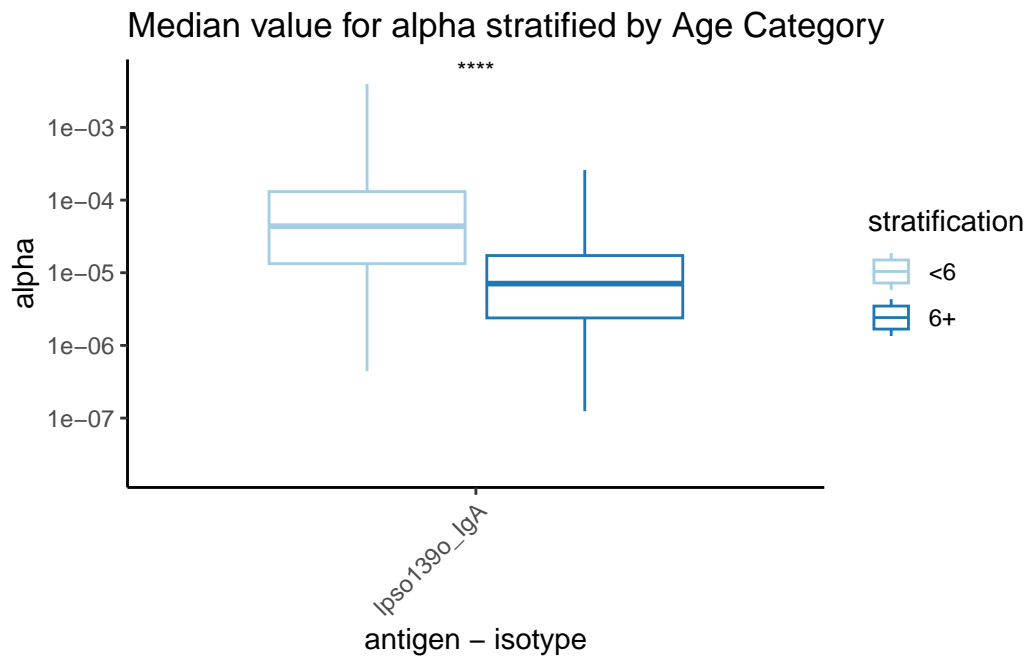
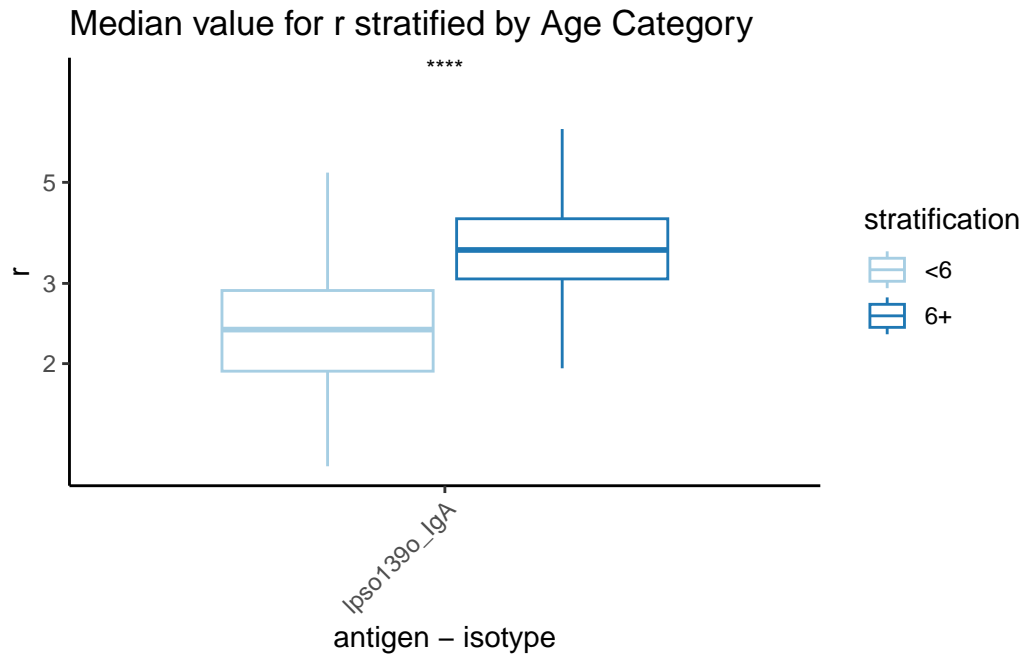




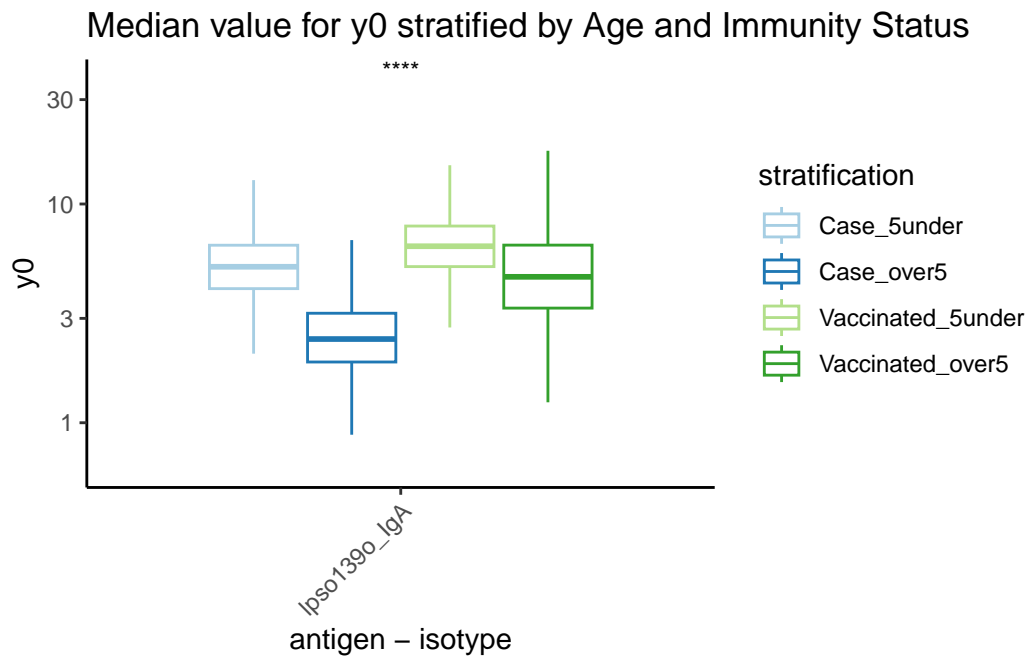
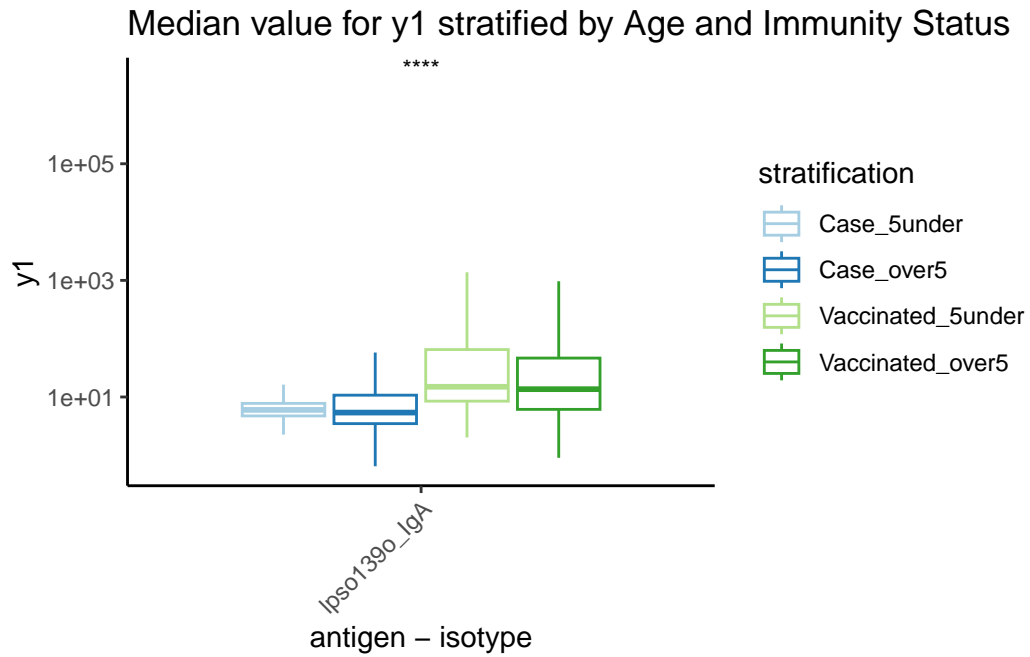
Antibody dynamic plots stratified by age group

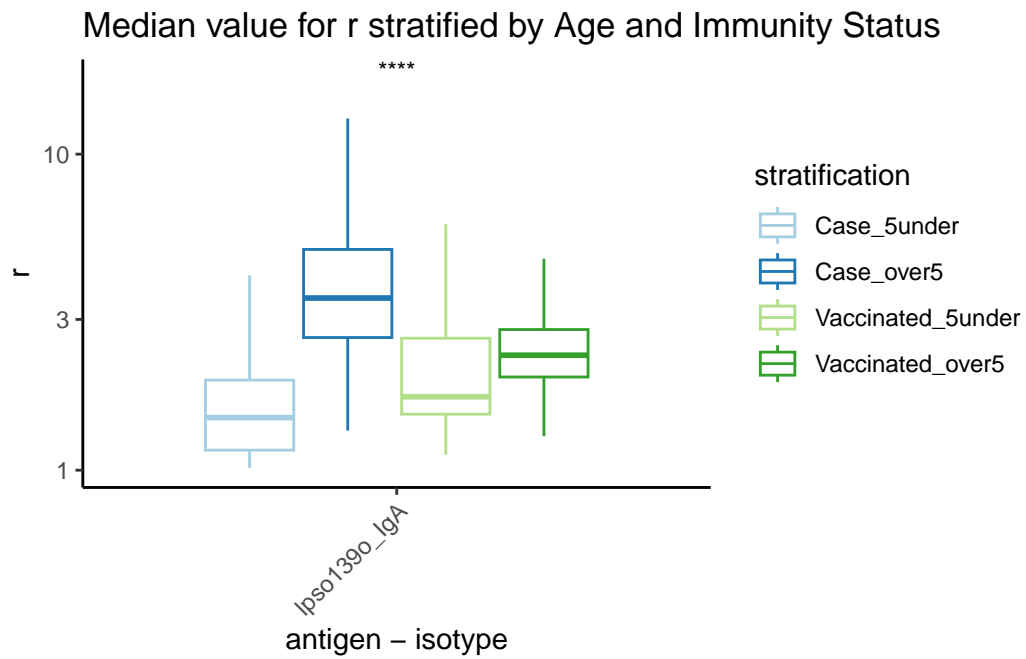
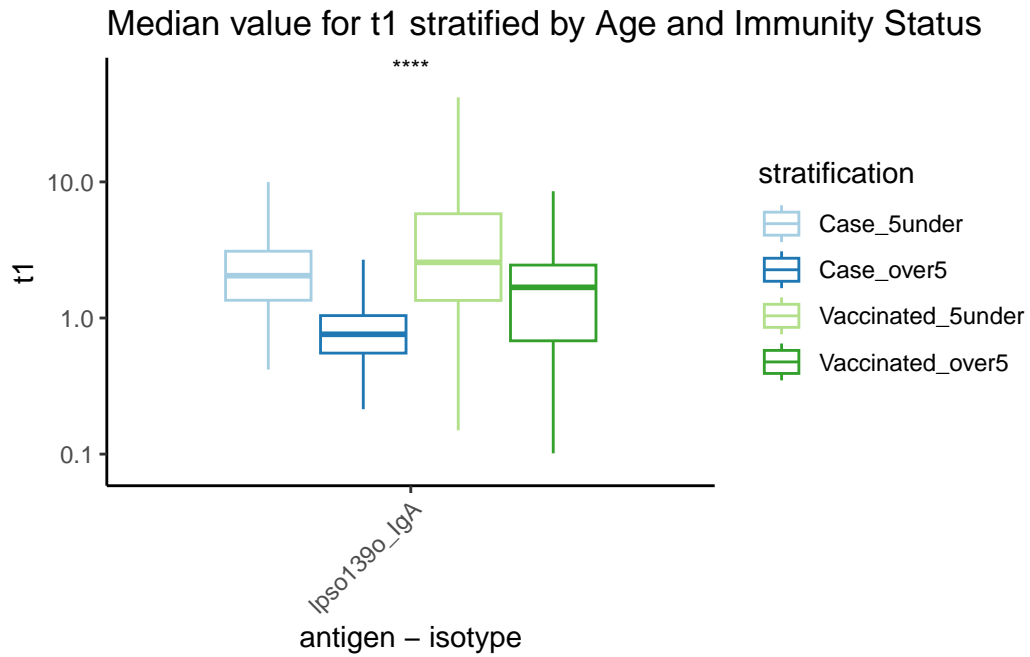


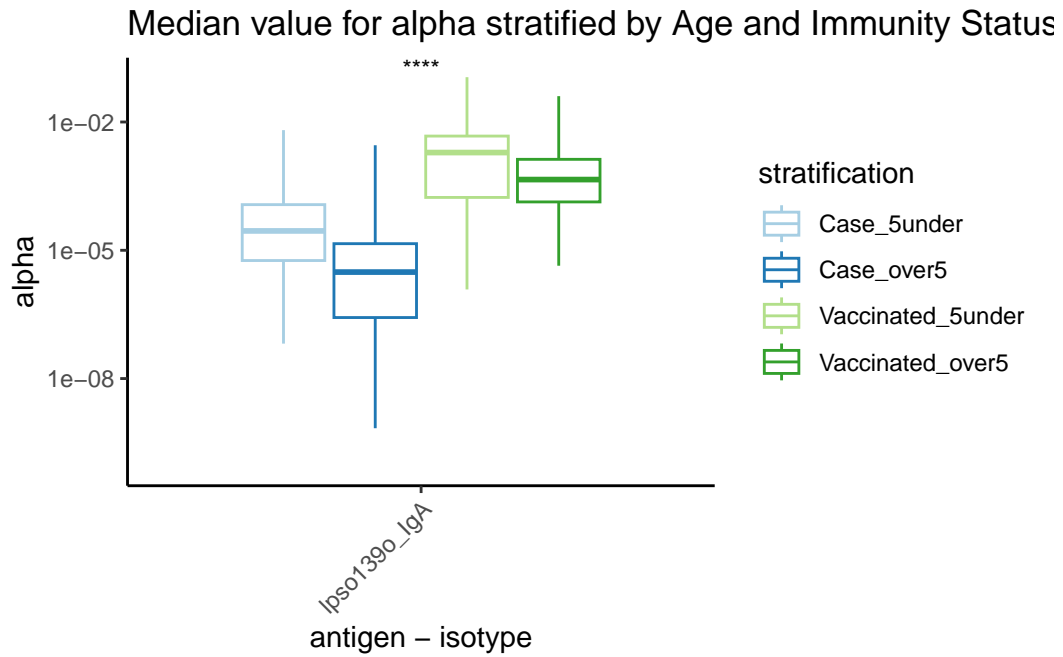




Antibody dynamic plots stratified by age group and immunity status







Antibody dynamic curves for each stratification