# Analysis TIC10 effect on activation of NaV1.5

Michael Fauler

22/06/2021

## Data Import and Plot

Data is loaded from an Excel spreadsheet. It has been manually transformed into long format and saved in sheet *TIC10_act_long*.

```
D <- read_excel(
  data_file,
  sheet = 'TIC10_act_long',
  col_names = T
) %>%
  mutate(Vp = Vp*1000) # transform from V to mV

D_longer <- D %>%
  pivot_longer(
    cols = c(C, TIC10),
    names_to = "treatment",
    values_to = "gnorm"
  )
```

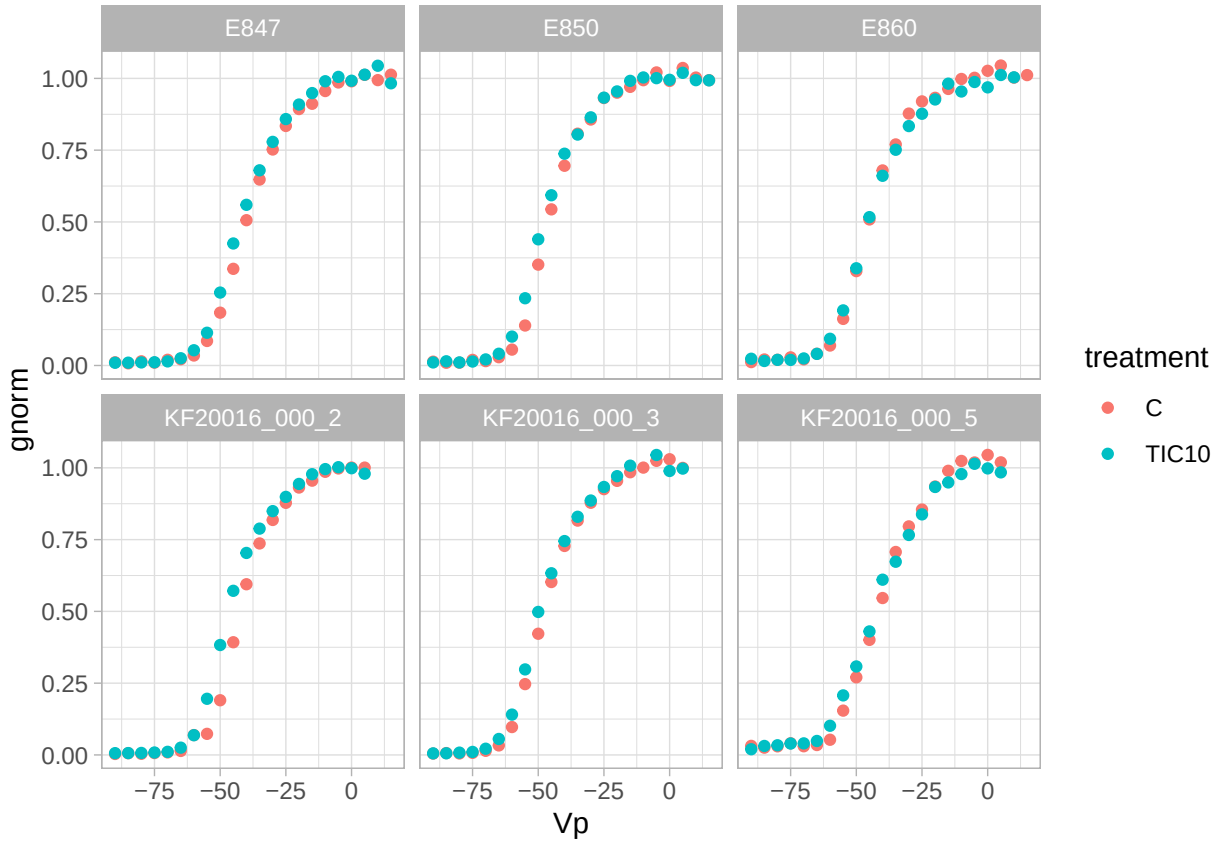
Plot Data. Data is normalized conductance either at *TIC10-treatment* or *control*.

```
fig1 <- ggplot(
  D_longer,
  aes(x=Vp, y=gnorm, col=treatment)
)
fig1 <- fig1 +
  geom_point()

fig1 <- fig1 +
  facet_wrap(~ID)

fig1 + theme_light()
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```




Activation curves are not symmetric. Sudden rise at threshold and flattened voltage-dependence above 75% of $g_{max}$. This might produce fitting problems. See *positive predictive checks* of bayesian modelling.

## Activation curve - Model Definition

The following model is fit to the data:

$$g_{norm} = \frac{1}{(1 + \frac{V_{50} + I_{TIC}\Delta V_{50} - V_p}{k + I_{TIC}\Delta k})}$$

Where $I_{TIC}$ is an indicator/dummy variable which is 1 for observations aquired in the presence of *TIC*-treatment and 0 otherwise.

```
fun.g_n_act <- function(Vp,treatment,v50, k, dv, dk) {
  n <- length(Vp)  # number of observations

  f1 <- rep(0, times=n) # pre-alloc memory

  ind_c <- treatment == 'C'
  ind_tic10 <- treatment == 'TIC10'

  f1[ind_c] <- (v50 - Vp[ind_c]) / k
  f1[ind_tic10] <- (v50 + dv - Vp[ind_tic10]) / (k + dk)

  f2 <- 1 + exp(f1)
```

```
  g <- 1 / f2

  return(g)
}
```

## Data fitting

Overall fit by applying the *nls()* function from R's base package.

```
model <- nls(
  gnorm ~ fun.g_n_act(Vp, treatment, v50, k, dv, dk),
  data = D_longer,
  start = list(
    v50 = -60,
    k = 1,
    dv = 0,
    dk = 0
  )
)

summary(model)
```

```
##
## Formula: gnorm ~ fun.g_n_act(Vp, treatment, v50, k, dv, dk)
##
## Parameters:
##     Estimate Std. Error  t value Pr(>|t|)
## v50 -43.1113     0.2738 -157.450  < 2e-16 ***
## k     7.4484     0.2411   30.891  < 2e-16 ***
## dv   -1.7043     0.3947   -4.318 2.28e-05 ***
## dk    0.5801     0.3477    1.668   0.0965 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04487 on 246 degrees of freedom
##
## Number of iterations to convergence: 7
## Achieved convergence tolerance: 3.697e-06
##   (2 observations deleted due to missingness)
```

Parameter $\Delta V_{50}$ ($= dv$) is significantly different to zero. Therefore, there is a shift of voltage-dependence in the TIC-treated group. The effect is small, its physiological relevance might be minor. Furthermore there is no appropriate control experiment ( *control* $\rightarrow$ *solvent* compared to *control* $\rightarrow$ *TIC*) which would capture systematic errors like drifts.

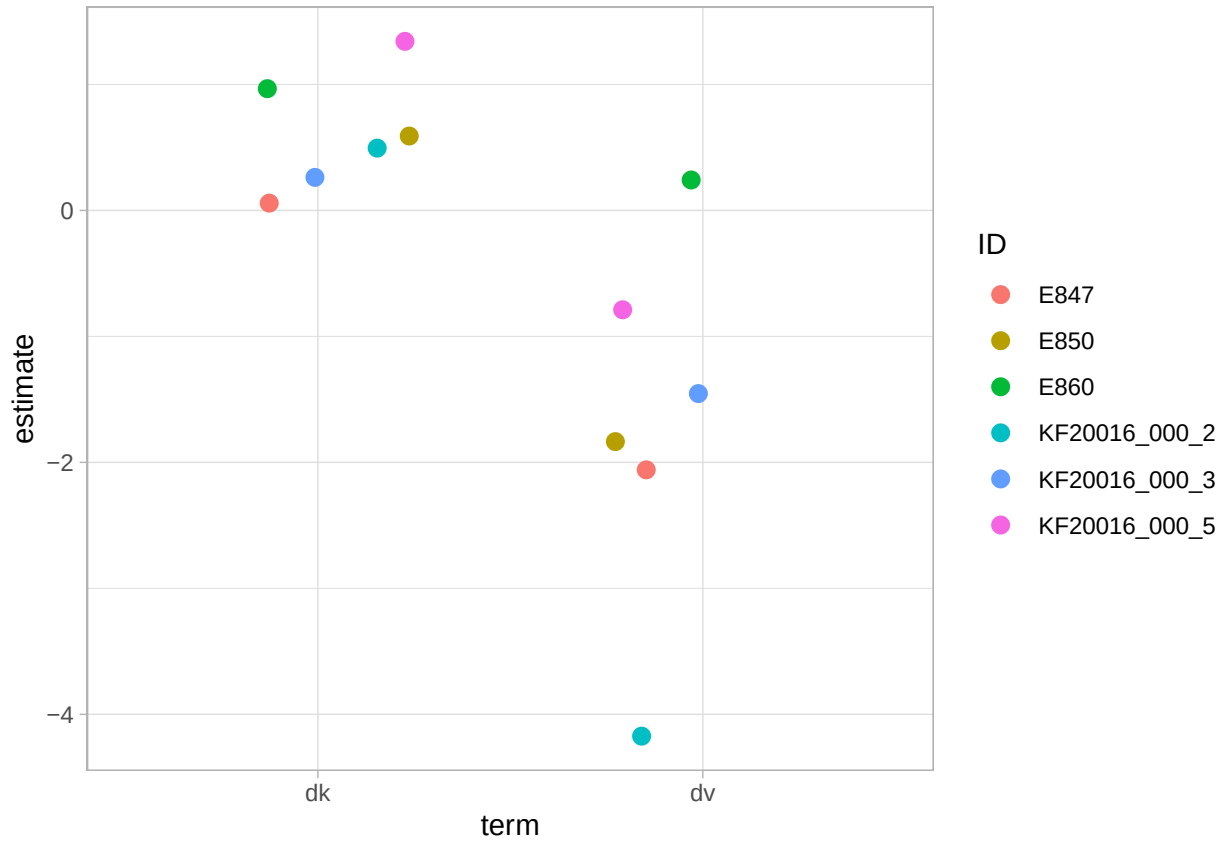
Inspection of the data suggests that there might be a relevant influence on the experiment level. Therefore, each experiment is fitted independently.

```
R.single_exp <- D_longer %>%
  group_by(ID) %>% # experiment ID, each experiment is fit within itself
  nest() %>%
  mutate(
```

```
    fit = purrr::map(
      data,
      ~ nls(
          gnorm ~ fun.g_n_act(Vp, treatment, v50, k, dv, dk),
          data = .,
          start = list(
            v50 = -60,
            k = 1,
            dv = 0,
            dk = 0
          )
        )
    )
  )

R.single_exp <- R.single_exp %>%
  mutate(
    params = fit %>% map(tidy)
  ) %>%
  unnest(params)

ggplot(
  R.single_exp %>% filter(term %in% c("dk","dv")),
  aes(x=term, col=ID)
) +
  geom_jitter(
    aes(y = estimate),
    width = .25,
    shape = 16,
    size = 3
  ) +
  theme_light()
```




The plot illustrates a rather large variance for the shift parameter of voltage-dependence *dv* ($=\Delta V_{50}$). Thus a hierarchical modelling framework is applied.

I use the *brms*-package which implements a bayesian approach based on *Stan*. Priors are defined informative, with a large enough variance as implied from previous fits. The parameter $V_{50}$ is augment by an additive *random effect* on the experiment level.

```
R.prior <- c(
  prior(normal(-40,10), nlpar='v50', class='b'),
  prior(normal(4,2), nlpar='k', lb=0, class='b'),
  prior(normal(0,4), nlpar='dv', class='b'),
  prior(normal(0,3), nlpar='dk', class='b')
)

D_longer <- D_longer %>%
  mutate(
    ID2 = case_when(
      ID %in% c('KF20016_000_2','KF20016_000_3','KF20016_000_5') ~ 'KF20016',
      T ~ ID
    )
  ) %>%
  mutate(
    treat_dummy = case_when(
      treatment == "C" ~ 0,
      treatment == "TIC10" ~ 1
    )
  )
```

```
R.bay_formula <- bf(
  gnorm ~ 1 / (1 + exp((v50 + treat_dummy*dv - Vp) / (k + treat_dummy*dk))),
  v50 ~ 1+(1|ID),
  k ~ 1,
  dv ~ 1,
  dk ~ 1,
  nl = T
)

R.bay_mod <- brm(
  formula = R.bay_formula,
  data = D_longer,
  family = gaussian(),
  prior = R.prior,
  warmup = 2000,
  iter = 1e4,
  sample_prior = 'yes',
  control = list(adapt_delta=.99),
  cores = 4
)
```

```
## Warning: Rows containing NAs were excluded from the model.
```

```
## Compiling Stan program...
```

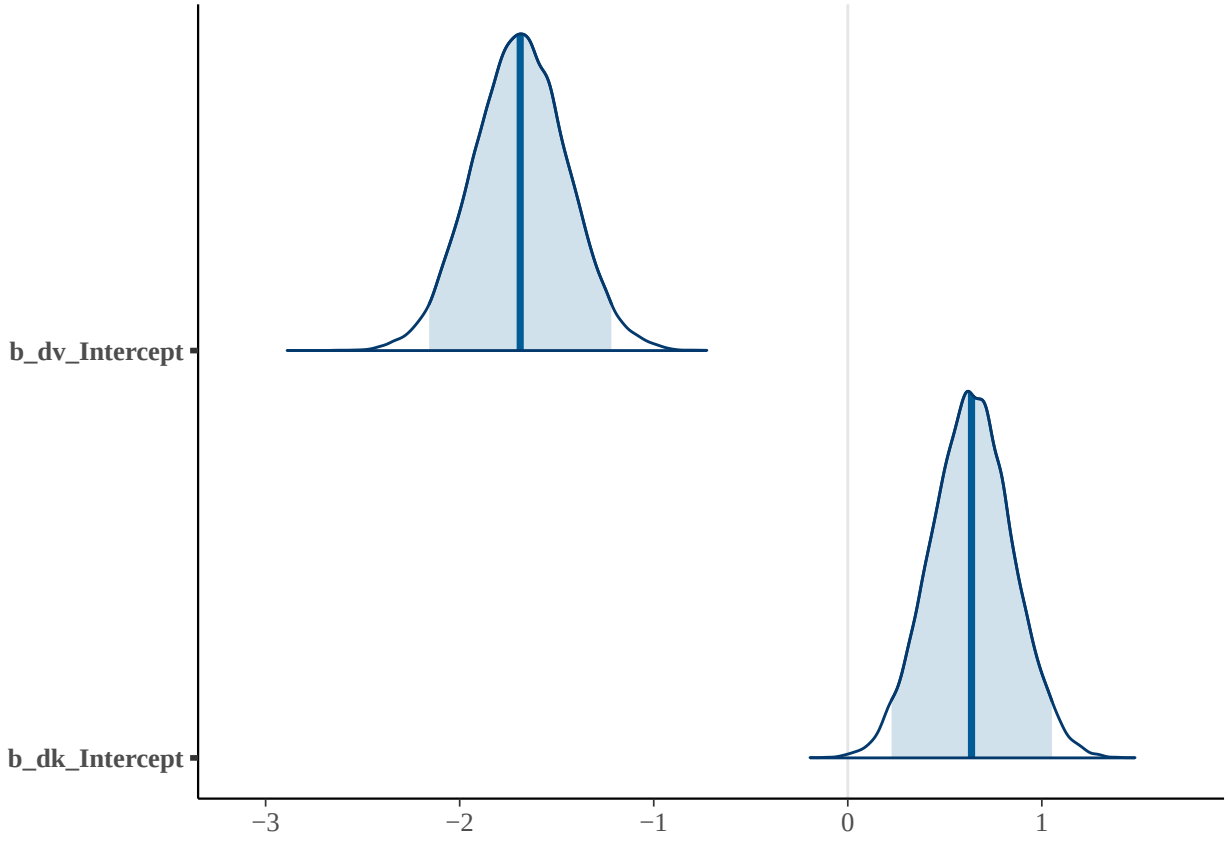
```
## Start sampling
```

```
summary(R.bay_mod)
```

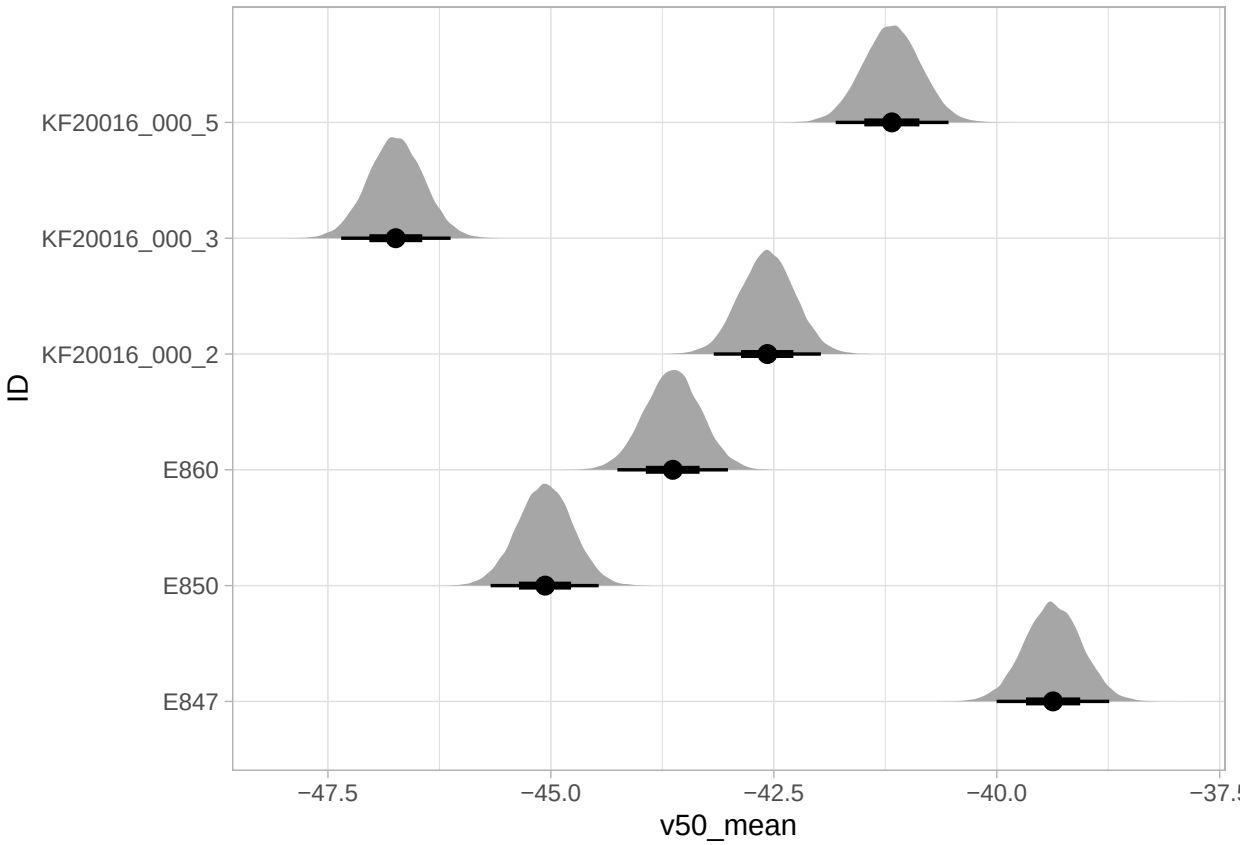
```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: gnorm ~ 1/(1 + exp((v50 + treat_dummy * dv - Vp)/(k + treat_dummy * dk)))
##          v50 ~ 1 + (1 | ID)
##          k ~ 1
##          dv ~ 1
##          dk ~ 1
##    Data: D_longer (Number of observations: 250)
## Samples: 4 chains, each with iter = 10000; warmup = 2000; thin = 1;
##          total post-warmup samples = 32000
##
## Group-Level Effects:
## ~ID (Number of levels: 6)
##                   Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(v50_Intercept)     2.95      0.98     1.66     5.36 1.00     9545    13676
##
## Population-Level Effects:
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## v50_Intercept   -43.07      1.26   -45.59   -40.51 1.00     7392    10350
## k_Intercept       7.24      0.15     6.96     7.53 1.00    22396    21732
## dv_Intercept     -1.69      0.24    -2.16    -1.22 1.00    28797    18989
## dk_Intercept      0.64      0.21     0.23     1.05 1.00    22790    21139
```

```
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.03      0.00     0.03     0.03 1.00    26798    19291
##
## 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).
```

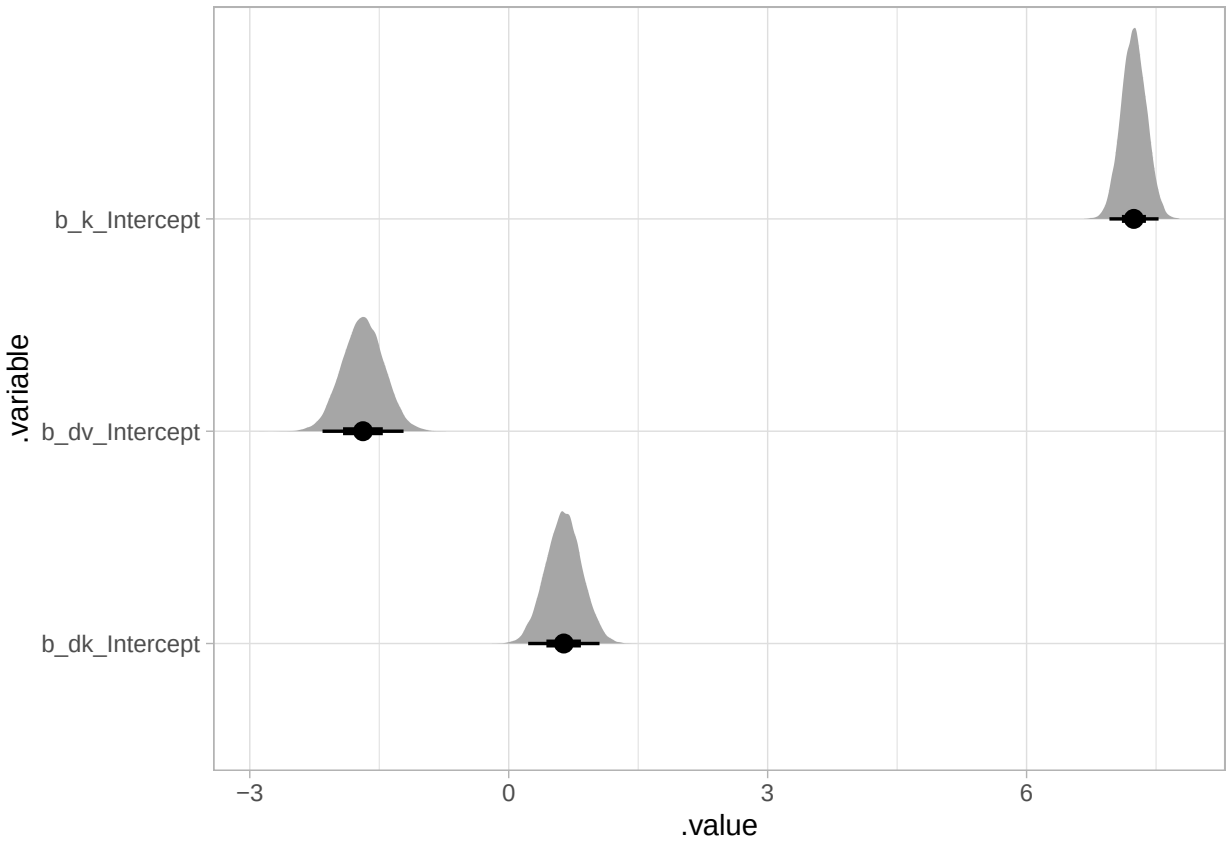
```
mcmc_areas(R.bay_mod$fit,pars=c('b_dv_Intercept','b_dk_Intercept'),prob=0.95)
```




```
R.bay_mod %>%
  spread_draws(b_v50_Intercept, r_ID__v50[ID,]) %>%
  mutate(v50_mean = b_v50_Intercept + r_ID__v50) %>%
  ggplot(
  aes(y=ID, x=v50_mean)
  ) +
  stat_halfeye() +
  theme_light()
```




```
R.bay_mod %>%
  gather_draws(b_k_Intercept, b_dv_Intercept, b_dk_Intercept) %>%
  ggplot(
    aes(y=.variable, x=.value)
  ) +
  stat_halfeye() +
  theme_light()
```



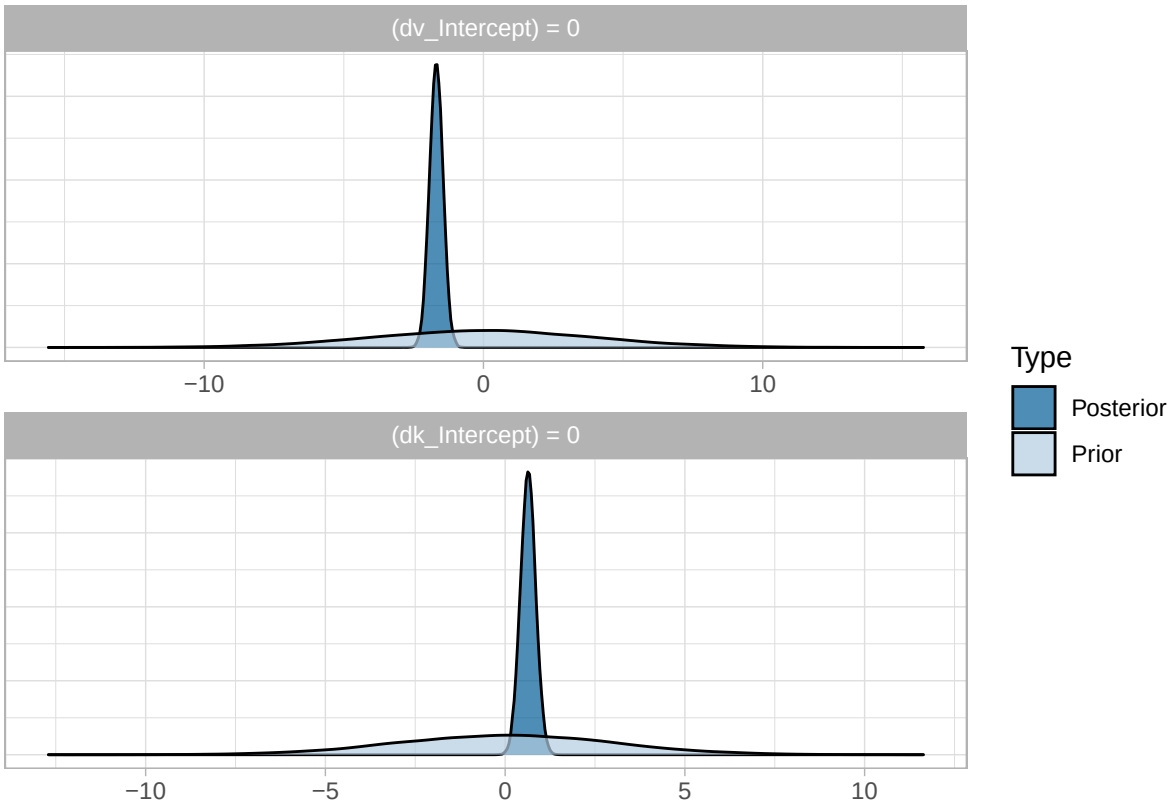

```
H <- brms::hypothesis(
       R.bay_mod,
       c(
         'dv_Intercept=0',
         'dk_Intercept=0'
       )
     )

print(H)
```

```
## Hypothesis Tests for class b:
##           Hypothesis Estimate Est.Error CI.Lower CI.Upper Evid.Ratio Post.Prob
## 1 (dv_Intercept) = 0    -1.69      0.24    -2.16    -1.22       0.00      0.00
## 2 (dk_Intercept) = 0     0.64      0.21     0.23     1.05       0.15      0.13
##   Star
## 1    *
## 2    *
## ---
## 'CI': 90%-CI for one-sided and 95%-CI for two-sided hypotheses.
## '*': For one-sided hypotheses, the posterior probability exceeds 95%;
## for two-sided hypotheses, the value tested against lies outside the 95%-CI.
## Posterior probabilities of point hypotheses assume equal prior probabilities.
```

```
plot(
  H,
```

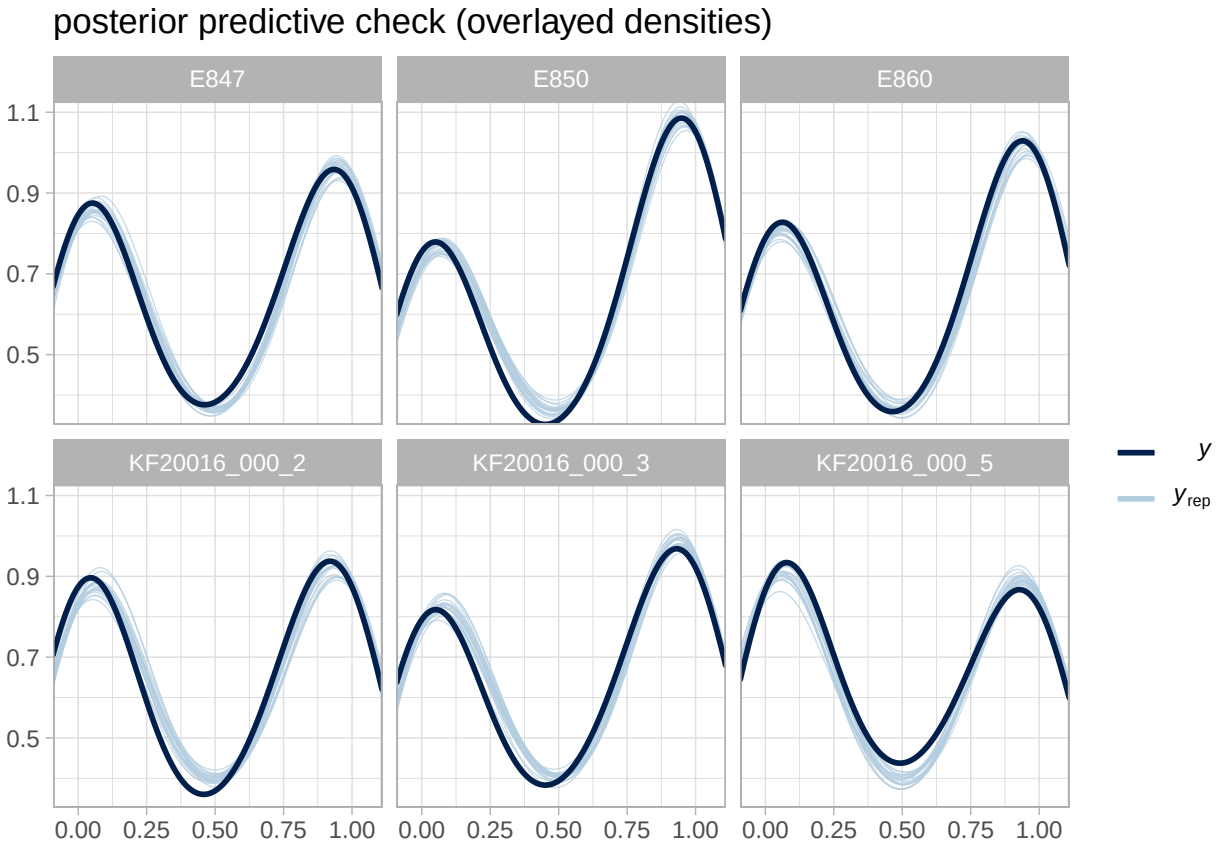
```
  theme = theme_light()
)
```




The data clearly informs the *posterior distribution*. The fit reveals a left-shift of the voltage-dependence of activation, since *dv_Intercept* is smaller than zero.

```
# D_longer %>%
#   data_grid(Vp=seq_range(Vp,25), ID, treatment, treat_dummy) %>%
#   add_predicted_draws(R.bay_mod) %>%
#   ggplot(aes(x = Vp, y=gnorm, col=ID)) +
#     stat_lineribbon(aes(y = .prediction)) +
#     scale_fill_brewer(palette='Greys') +
#     scale_color_brewer(palette='Set2') +
#     facet_wrap(~treatment) +
#     theme_light()

pp_check(
  R.bay_mod,
  type = 'dens_overlay_grouped',
  nsamples = 20,
  group = 'ID'
) +
  ggtitle("posterior predictive check (overlayed densities)") +
  theme_light()
```




The *posterior predictive check* highlights the modelling problems since data is not symmetric.