fig2_gck.Rmd

2025-06-02

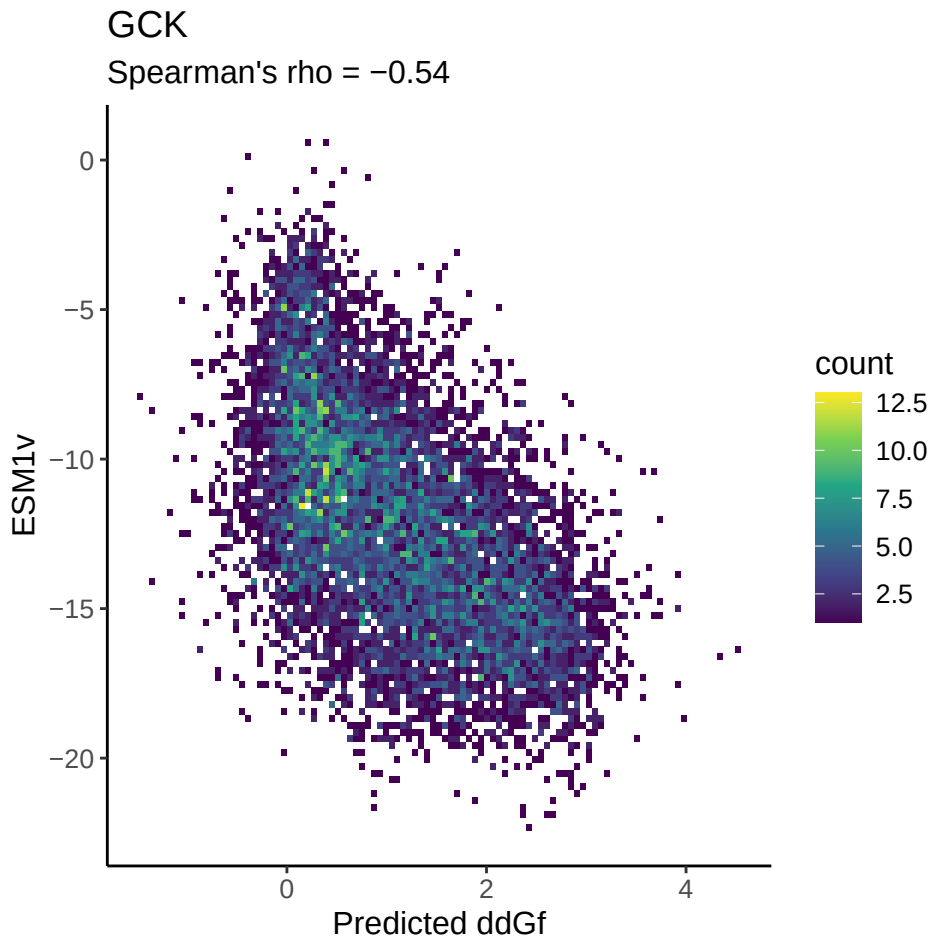
GCK

```
analyze_ddg_vs_esm1v <- function(ddg_file, esm_file, protein_name = "Protein") {
  # Load data
  test_ddg <- fread(ddg_file)
  test_esm <- fread(esm_file)

  # Prepare ESM1v column
  colnames(test_esm)[2] <- "ESM1v"

  # Construct variant column in ddg data
  test_ddg[, new_position := pos + 1]
  test_ddg[, variant := paste0(wtAA, new_position, mutAA)]

  # Merge on variant
  test_df <- merge(test_ddg, test_esm, by = "variant")

  # Rename ddG column
  test_df <- test_df %>% dplyr::rename(ddG_pred = `ddG (kcal/mol)`)

  # Spearman correlation
  spearman_rho <- cor.test(test_df$ddG_pred, test_df$ESM1v, method = "spearman")
  rho_value <- round(spearman_rho$estimate, 2)

  # Plot
  p <- ggplot(test_df, aes(x = ddG_pred, y = ESM1v)) +
    geom_bin2d(bins = 100) +
    scale_fill_continuous(type = "viridis") +
    theme_classic() +
    labs(
      x = "Predicted ddGf",
      y = "ESM1v",
      title = protein_name,
      subtitle = paste("Spearman's rho =", rho_value)
    ) +
    theme(
      text = element_text(size = 12),
      legend.position = "right"
    )

  # Output
  list(
```

```
    spearman_rho = spearman_rho,
    plot = p,
    merged_data = test_df
  )
}
gck_pred <- analyze_ddg_vs_esm1v(
  ddg_file = "/Users/xl7/Documents/0.Projects/01.protein-seq-evo-v1/data/decay_pdb/GCK/AF-P35557-F1-mode
  esm_file = "/Users/xl7/Documents/0.Projects/00.large_supplements/ESM1v_proteome_wide/all_ESM1v/P35557
  protein_name = "GCK"
)
```

```
## Warning in cor.test.default(test_df$ddG_pred, test_df$ESM1v, method =
## "spearman"): Cannot compute exact p-value with ties
```

```
#result$spearman_rho #-0.5410593
p0 <- gck_pred$plot
ggsave("/Users/xl7/Documents/0.Projects/01.protein-seq-evo-v1/figs/panels/fig2_gck_p0.pdf",
       plot = p0, width = 5, height = 5, dpi = 300)
p0
```

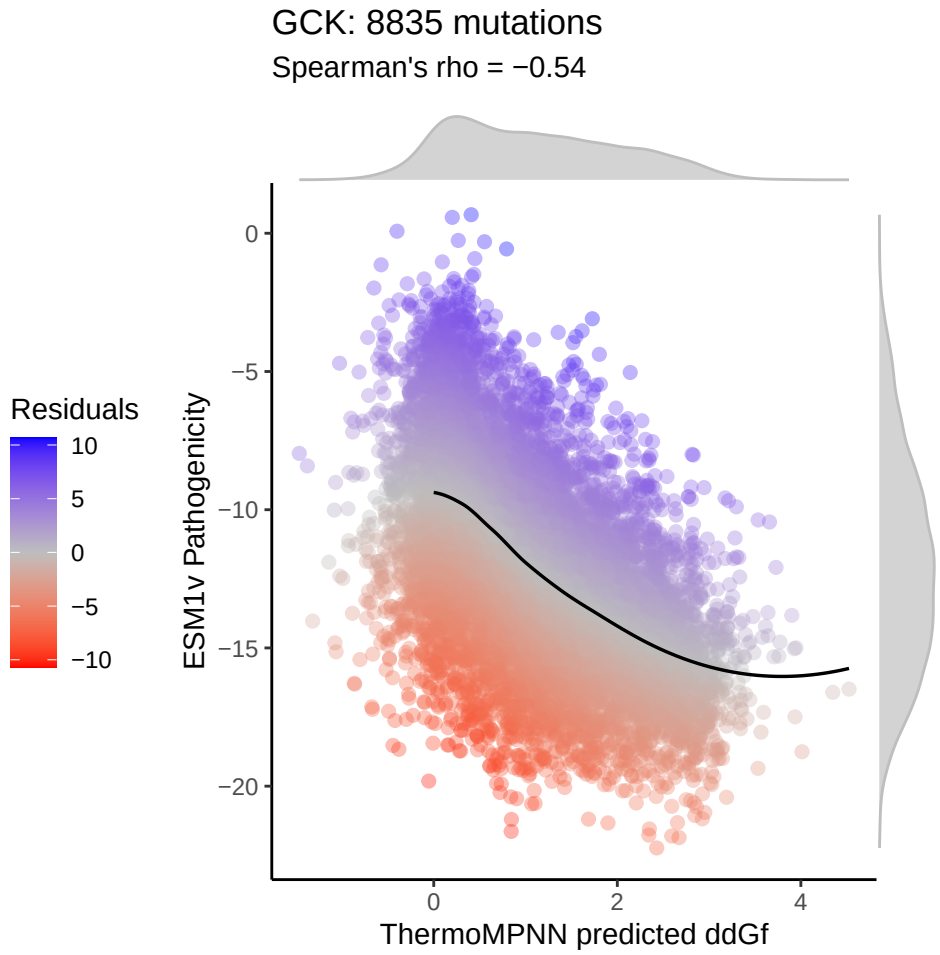


```r
plot_loess_residuals <- function(test_df, active_site_positions,
                                 span = 0.7, protein_name = "GCK") {

  # Filter out active site positions
  test_df_fil <- test_df %>% filter(!new_position %in% active_site_positions)

  # Fit loess model on filtered data
  loess_fit <- loess(ESM1v ~ ddG_pred, data = test_df_fil, span = span, family = "symmetric")

  # Predict on all data
  test_df$fitted_pred <- predict(loess_fit, newdata = test_df)
  test_df$residuals_pred <- test_df$ESM1v - test_df$fitted_pred

  # Fit line data for the smooth curve
  fit_line_df <- data.frame(
    ddG_pred = seq(0,
                   max(test_df$ddG_pred, na.rm = TRUE),
                   length.out = 200)
  )
  fit_line_df$ESM1v <- predict(loess_fit, newdata = fit_line_df)

  # Spearman correlation
  spearman_result <- suppressWarnings(cor.test(test_df$ddG_pred, test_df$ESM1v, method = "spearman"))
  spearman_rho <- spearman_result$estimate
  spearman_p <- spearman_result$p.value

  # Axis and color scale limits
  xlim_vals <- range(test_df$ddG_pred, na.rm = TRUE)
  ylim_vals <- range(test_df$ESM1v, na.rm = TRUE)
  resid_limit <- max(abs(test_df$residuals_pred), na.rm = TRUE)

  # Main plot
  p <- ggplot(test_df, aes(x = ddG_pred , y = ESM1v, color = residuals_pred)) +
    geom_point(size = 2, alpha = 0.35) +
    geom_line(data = fit_line_df, aes(x = ddG_pred, y = ESM1v),
              inherit.aes = FALSE, color = "black", linewidth = 0.6) +
    labs(
      title = paste0(protein_name, ": ", nrow(test_df), " mutations"),
      subtitle = paste0("Spearman's rho = ", round(spearman_rho, 2)),
      x = "ThermoMPNN predicted ddGf",
      y = "ESM1v Pathogenicity",
      color = "ESM1v-ddGf residuals"
    ) +
    theme_classic() +
    xlim(xlim_vals) +
    ylim(ylim_vals) +
    scale_color_gradient2(
      low = "red", mid = "grey", high = "blue", midpoint = 0,
      limits = c(-resid_limit, resid_limit), name = "Residuals"
    ) +
    theme(legend.position = "left")

  # Add marginal density plots
```

```
  p_marginal <- ggMarginal(
    p,
    type = "density",
    margins = "both",
    groupColour = FALSE,
    groupFill = FALSE,
    size = 10,
    colour = "grey",
    fill = "lightgrey"
  )

  return(list(
    plot = p_marginal,
    data = test_df
  ))
}
head(gck_pred$merged_data)
```

```
## Key: <variant>
##    variant    V1 Mutation ddG_pred   pos   wtAA  mutAA new_position     ESM1v
##     <char> <int>   <char>    <num> <int> <char> <char>        <num>     <num>
## 1:    A10C   181      A9C   0.0486     9      A      C           10 -6.882101
## 2:    A10D   182      A9D   0.3884     9      A      D           10 -5.333317
## 3:    A10E   183      A9E   0.0151     9      A      E           10 -4.009192
## 4:    A10F   184      A9F   0.3177     9      A      F           10 -6.812537
## 5:    A10G   185      A9G   0.5904     9      A      G           10 -4.622459
## 6:    A10H   186      A9H   0.2476     9      A      H           10 -6.842433
```

```
gck_pred_residual <- plot_loess_residuals(gck_pred$merged_data, active_site_positions = c(80, 151, 152,
p1 <- gck_pred_residual$plot
ggsave("/Users/xl7/Documents/0.Projects/01.protein-seq-evo-v1/figs/panels/fig2_gck_p1.pdf",
       plot = p1, width = 5, height = 4, dpi = 300)
p1
```




```r
map_loess_residuals_to_pdb <- function(test_df, pdb_path, output_pdb_path) {

  # 1. Compute median residuals per position
  median_residuals <- test_df %>%
    group_by(new_position) %>%
    summarise(median_residuals = median(residuals_pred, na.rm = TRUE), .groups = "drop")

  # 2. Read in PDB
  pdb <- read.pdb(pdb_path)

  # 3. Initialize new B-factor vector
  new_b_factors <- pdb$atom$b

  # 4. Map residuals to matching residue numbers in the PDB
  for (i in seq_len(nrow(median_residuals))) {
    pos <- median_residuals$new_position[i]
    val <- median_residuals$median_residuals[i]
    matching_indices <- which(pdb$atom$resno == pos)
    new_b_factors[matching_indices] <- val
  }

  # 5. Replace non-matching indices with outlier value (e.g., 999)
  matched_positions <- unique(median_residuals$new_position)
```

```
  non_matching_indices <- which(!(pdb$atom$resno %in% matched_positions))
  new_b_factors[non_matching_indices] <- 999

  # 6. Assign and save new PDB
  pdb$atom$b <- new_b_factors
  write.pdb(pdb, file = output_pdb_path)

  # Optional: return summary
  return(list(
    min_residual = min(median_residuals$median_residuals, na.rm = TRUE),
    max_residual = max(median_residuals$median_residuals, na.rm = TRUE),
    length(non_matching_indices),
    output_file = output_pdb_path
  ))
}

pdb_residual <- map_loess_residuals_to_pdb(
  test_df = gck_pred_residual$data,
  pdb_path = "~/Documents/0.Projects/01.protein-seq-evo-v1/data/decay_pdb/GCK/1v4s.pdb",
  output_pdb_path = "~/Documents/0.Projects/01.protein-seq-evo-v1/data/decay_pdb/GCK/1v4s_loess_residual
)

print(pdb_residual)
```

```
## $min_residual
## [1] -6.138725
##
## $max_residual
## [1] 6.545383
##
## [[3]]
## [1] 185
##
## $output_file
## [1] "~/Documents/0.Projects/01.protein-seq-evo-v1/data/decay_pdb/GCK/1v4s_loess_residual.pdb"
```

```
#chimerax
#color byattribute a:bfactor #2 & sel target csab palette -6.2,red:0,white:7,blue
```

```
# --- Read PDB and extract protein/ligand atoms ---
pdb <- read.pdb("~/Documents/0.Projects/01.protein-seq-evo-v1/data/decay_pdb/GCK/1v4s.pdb", rm.alt = TRU

protein_ca <- pdb$atom %>%
  filter(elety == "CA", !resid %in% c("GLC", "NA", "MRK", "HOH"))

ligand_atoms <- pdb$atom %>%
  filter(resid == "GLC", type == "HETATM")

# --- Compute minimum distance to ligand for each CA atom ---
protein_ca$min_dist_to_ligand <- apply(protein_ca, 1, function(atom) {
  dists <- sqrt((as.numeric(atom["x"]) - ligand_atoms$x)^2 +
                (as.numeric(atom["y"]) - ligand_atoms$y)^2 +
                (as.numeric(atom["z"]) - ligand_atoms$z)^2)
```

```r
  min(dists)
})

# --- Merge with prediction data ---
merged_df <- merge(gck_pred_residual$data, protein_ca, by.x = "new_position", by.y = "resno") %>%
  filter(residuals_pred <= 0)

# --- Residue-level median residuals ---
merged_df_residue <- merged_df %>%
  group_by(new_position) %>%
  summarise(loess_residual_avg = median(residuals_pred, na.rm = TRUE), .groups = "drop") %>%
  left_join(protein_ca, by = c("new_position" = "resno"))

# --- Exclude orthosteric sites for fitting ---
orthosteric_sites <- c(80, 151, 152, 153, 168, 169, 204, 205, 206, 225, 229, 230, 231, 256, 258) # base
ortho_cutoff <- max(merged_df_residue %>% filter(new_position %in% orthosteric_sites) %>% pull(min_dist_
#merged_df_residue %>% filter(min_dist_to_ligand <= ortho_cutoff) %>% pull(new_position)
#80 151 152 153 168 169 204 205 206 225 229 230 231 256 258

merged_df_residue_fil <- merged_df_residue %>%
  filter(!new_position %in% orthosteric_sites)

# --- Fit exponential model ---
exp_model <- nlsLM(
  abs(loess_residual_avg) ~ a * exp(-b * min_dist_to_ligand),
  data = merged_df_residue,
  start = list(a = 1, b = 0.1)
)
exp_model
```

```
## Nonlinear regression model
##   model: abs(loess_residual_avg) ~ a * exp(-b * min_dist_to_ligand)
##    data: merged_df_residue
##       a       b
## 4.47423 0.04405
##  residual sum-of-squares: 496.1
##
## Number of iterations to convergence: 7
## Achieved convergence tolerance: 1.49e-08
```

```r
# --- Prediction grid ---
x_vals <- seq(min(merged_df_residue$min_dist_to_ligand, na.rm = TRUE),
              max(merged_df_residue$min_dist_to_ligand, na.rm = TRUE), length.out = 200)

# --- Bootstrapping for confidence intervals ---
set.seed(11)
boot_params <- replicate(1000, {
  samp <- merged_df_residue[sample(nrow(merged_df_residue), replace = TRUE), ]
  fit <- try(nlsLM(abs(loess_residual_avg) ~ a * exp(-b * min_dist_to_ligand),
                   data = samp, start = list(a = 1, b = 0.1)), silent = TRUE)
  if (inherits(fit, "try-error")) c(NA, NA) else coef(fit)
})
boot_params <- t(boot_params)[complete.cases(t(boot_params)), ]
```

```r
boot_preds <- apply(boot_params, 1, function(p) p[1] * exp(-p[2] * x_vals))
fit_df_residue <- data.frame(
  min_dist_to_ligand = x_vals,
  loess_residual_pred = predict(exp_model, newdata = data.frame(min_dist_to_ligand = x_vals)),
  lower = apply(boot_preds, 1, quantile, probs = 0.025),
  upper = apply(boot_preds, 1, quantile, probs = 0.975)
)

# --- Model parameter extraction and derived quantity ---
model_summary <- summary(exp_model)
coefs <- coef(exp_model)
se <- model_summary$coefficients[, "Std. Error"]

residue_a <- c(coefs["a"],
               coefs["a"] - 1.96 * se["a"],
               coefs["a"] + 1.96 * se["a"])

residue_b <- c(coefs["b"],
               coefs["b"] - 1.96 * se["b"],
               coefs["b"] + 1.96 * se["b"])

half_d <- log(2) / coefs["b"]
half_d_ci <- quantile(log(2) / boot_params[, "b"], probs = c(0.025, 0.975))
residue_half_d <- c(half_d, half_d_ci)

cat("Parameter a (intercept):", residue_a, "\n")
```

```
## Parameter a (intercept): 4.474228 3.934935 5.01352
```

```r
cat("Parameter b (decay rate):", residue_b, "\n")
```

```
## Parameter b (decay rate): 0.04405008 0.03660082 0.05149934
```

```r
cat("Half-distance (log(2)/b):", residue_half_d, "\n")
```

```
## Half-distance (log(2)/b): 15.73543 13.459 19.21691
```

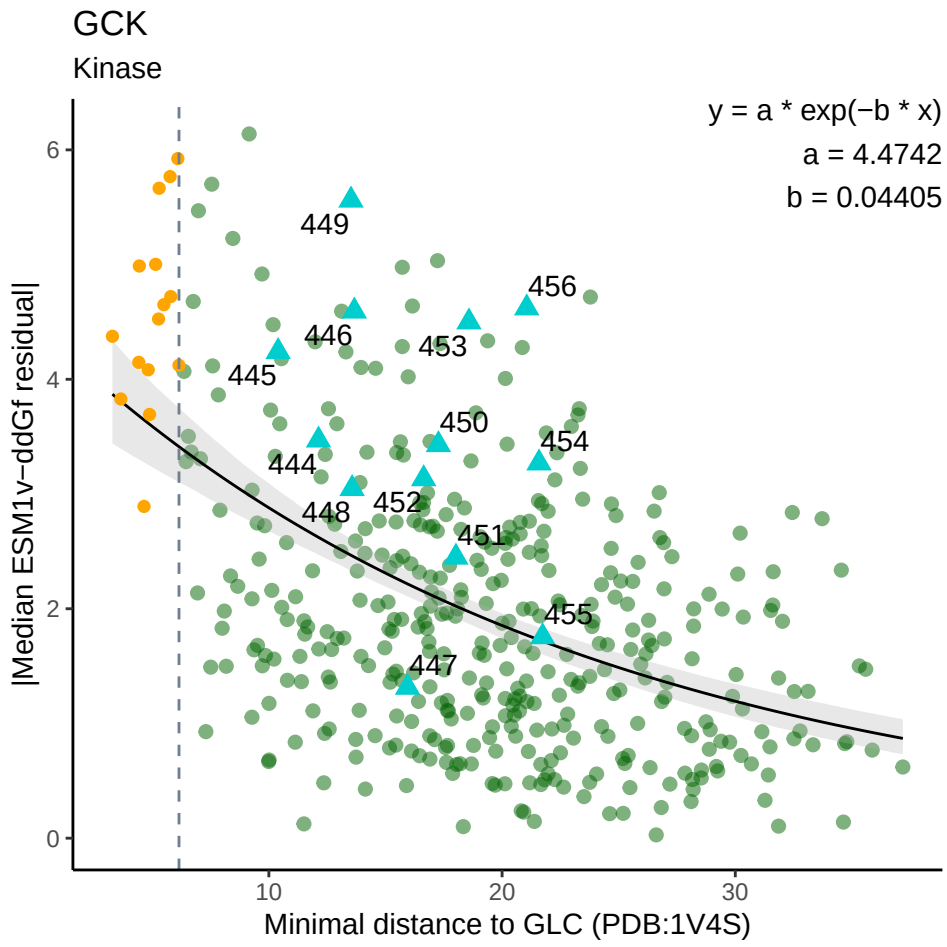
```r
# Number of iterations to convergence: 7
# Achieved convergence tolerance: 1.49e-08
# Parameter a (intercept): 4.474228 3.934935 5.01352
# Parameter b (decay rate): 0.04405008 0.03660082 0.05149934
# Half-distance (log(2)/b): 15.73543 13.459 19.21691
```

```r
# --- Annotate site types ---
merged_df_residue <- merged_df_residue %>%
  mutate(site_type = if_else(new_position %in% orthosteric_sites, "orthosteric", "non-orthosteric"))

# --- Plot ---
orange_labs <- orthosteric_sites
cyan_labs <- c(444:456)
```

```r
all_labs <- union(orange_labs, cyan_labs)

p2 <- ggplot(merged_df_residue, aes(x = min_dist_to_ligand, y = abs(loess_residual_avg))) +

  # Unlabeled points
  geom_point(data = subset(merged_df_residue, !new_position %in% all_labs),
             aes(color = site_type), size = 2, alpha = 0.5) +

  # CI ribbon
  geom_ribbon(
  data = fit_df_residue,
  aes(x = min_dist_to_ligand, ymin = lower, ymax = upper),
  inherit.aes = FALSE,
  fill = "grey70",
  alpha = 0.3) +

  # Main fit line
  geom_line(
  data = fit_df_residue,
  aes(x = min_dist_to_ligand, y = loess_residual_pred),
  inherit.aes = FALSE,
  color = "black")+


  # Labeled orange points
  geom_point(data = subset(merged_df_residue, new_position %in% orange_labs),
             shape = 16, size = 2, color = "orange") +
  # Labeled cyan points
  geom_point(data = subset(merged_df_residue, new_position %in% cyan_labs),
             shape = 17, size = 3, color = "cyan3") +
  geom_text_repel(data = subset(merged_df_residue, new_position %in% cyan_labs),
                  aes(label = new_position), color = "black") +

  # Reference line
  geom_vline(xintercept = max(merged_df_residue %>% filter(site_type == "orthosteric") %>% pull(min_dis
             linetype = "dashed", color = "slategrey") +

  # Labels and theme
  labs(
    title = "GCK",
    subtitle = "Kinase",
    x = "Minimal distance to GLC (PDB:1V4S)",
    y = "|Median ESM1v-ddGf residual|"
  ) +
  theme_classic() +
  theme(legend.position = "none") +
  scale_color_manual(values = c("non-orthosteric" = "darkgreen", "orthosteric" = "orange")) +

  annotate("text", x = Inf, y = Inf, hjust = 1, vjust = 1,
           label = sprintf("y = a * exp(-b * x)\na = %.4f\nb = %.5f", coefs["a"], coefs["b"]),
           size = 4, color = "black", hjust = 0)
```

```
## Warning: Duplicated aesthetics after name standardisation: hjust
```

```
ggsave("/Users/xl7/Documents/0.Projects/01.protein-seq-evo-v1/figs/panels/fig2_gck_p2.pdf",
       plot = p2, width = 4, height = 4, dpi = 300)
p2
```




```
lm_model <- lm(log(abs(loess_residual_avg)) ~ min_dist_to_ligand, data = merged_df_residue)
summary(lm_model)
```

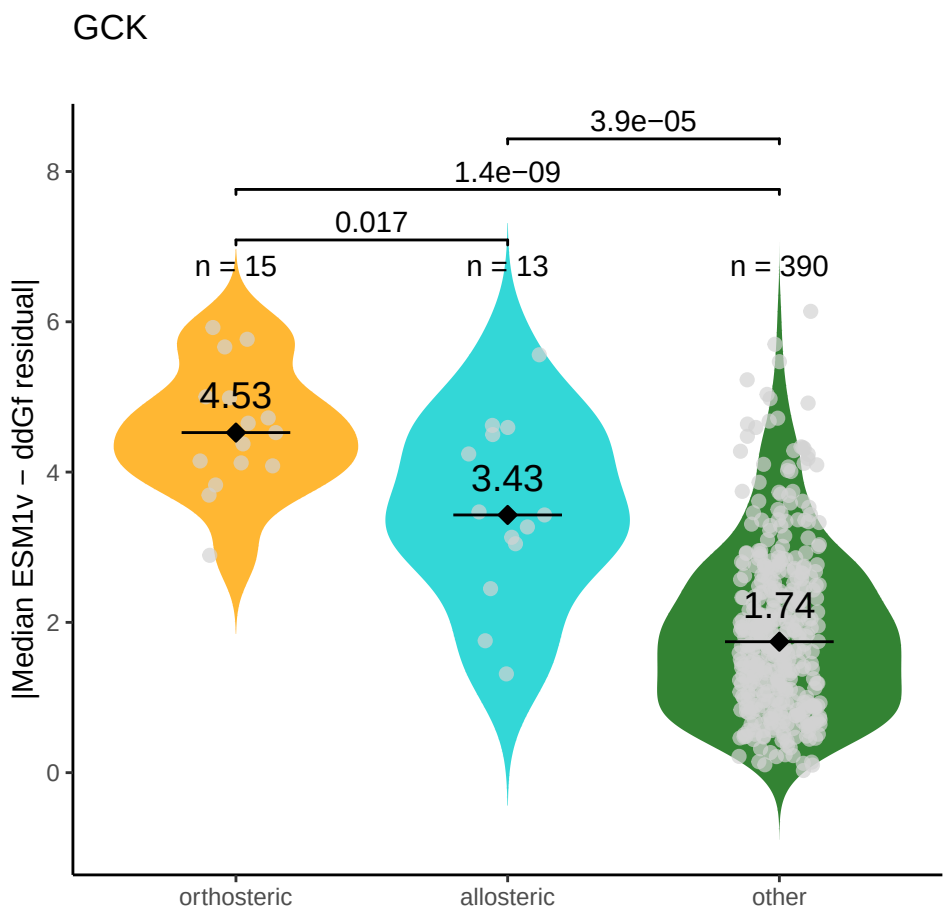
```
##
## Call:
## lm(formula = log(abs(loess_residual_avg)) ~ min_dist_to_ligand,
##     data = merged_df_residue)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7150 -0.3620  0.1186  0.4890  1.2697
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.307726   0.095559  13.685   <2e-16 ***
## min_dist_to_ligand -0.043143   0.004717  -9.147   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.686 on 416 degrees of freedom
## Multiple R-squared:  0.1674, Adjusted R-squared:  0.1654
## F-statistic: 83.67 on 1 and 416 DF,  p-value: < 2.2e-16
```

```
# Call:
# lm(formula = log(abs(loess_residual_avg)) ~ min_dist_to_ligand,
#     data = merged_df_residue)
#
# Residuals:
#     Min      1Q  Median      3Q     Max
# -3.7150 -0.3620  0.1186  0.4890  1.2697
#
# Coefficients:
#                     Estimate Std. Error t value Pr(>|t|)
# (Intercept)         1.307726   0.095559  13.685   <2e-16 ***
# min_dist_to_ligand -0.043143   0.004717  -9.147   <2e-16 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 0.686 on 416 degrees of freedom
# Multiple R-squared:  0.1674,  Adjusted R-squared:  0.1654
# F-statistic: 83.67 on 1 and 416 DF,  p-value: < 2.2e-16
```

```
allosteric_sites <- c(444:456)

merged_df_residue <- merged_df_residue %>%
  mutate(site_class = case_when(
    new_position %in% orthosteric_sites ~ "orthosteric",
    new_position %in% allosteric_sites ~ "allosteric",
    TRUE ~ "other"
  ))

label_df <- merged_df_residue %>%
  group_by(site_class) %>%
  summarise(
    n = n(),
    median_val = median(abs(loess_residual_avg), na.rm = TRUE),
    .groups = "drop"
  )

merged_df_residue$site_class <- factor(merged_df_residue$site_class, levels = c("orthosteric", "alloste

p3 <- ggplot(merged_df_residue, aes(x = site_class, y = abs(loess_residual_avg), fill = site_class)) +
  geom_violin(trim = FALSE, scale = "width", alpha = 0.8, color = NA)+
  geom_jitter(width = 0.15, size = 2, alpha = 0.7, color = "lightgrey") +
  stat_summary(fun = median, geom = "crossbar", width = 0.4, color = "black", fatten = 1) +
  stat_summary(fun = median, geom = "point", shape = 23, size = 2, fill = "black", color = "black", str
  # Add sample size n=xxx above each group
  geom_text(
    data = label_df,
    aes(x = site_class, y = max(abs(merged_df_residue$loess_residual_avg)) * 1.1,
        label = paste0("n = ", n)),
    inherit.aes = FALSE,
```

```
    size = 4
  ) +
  geom_text(
  data = label_df,
  aes(x = site_class, y = median_val + 0.5, label = sprintf("%.2f", median_val)),
  inherit.aes = FALSE,
  size = 5
) +
  # Significance bars
  geom_signif(
    comparisons = list(
      c("orthosteric", "allosteric"),
      c("orthosteric", "other"),
      c("allosteric", "other")
    ),
    map_signif_level = FALSE,
    test = "wilcox.test",
    step_increase = 0.1,
    tip_length = 0.01
  ) +

  # Labels and theme
  labs(
    title = "GCK",
    subtitle = "",
    x = "",
    y = "|Median ESM1v - ddGf residual|"
  ) +
  scale_fill_manual(values = c(
    "orthosteric" = "orange",
    "allosteric" = "cyan3",
    "other" = "darkgreen"
  )) +
  theme_classic() +
  theme(legend.position = "none")
ggsave("/Users/xl7/Documents/0.Projects/01.protein-seq-evo-v1/figs/panels/fig2_gck_p3.pdf",
       plot = p3, width = 3, height = 4, dpi = 300)
p3
```




```r
set.seed(11)

# Get fixed per-residue medians
fixed_df <- merged_df_residue %>%
  filter(site_class %in% c("orthosteric", "allosteric")) %>%
  mutate(median_resid = abs(loess_residual_avg)) %>%
  dplyr::select(site_class, median_resid)

# Bootstrap medians for 'other' group
n_sample <- sum(merged_df_residue$site_class == "orthosteric")

boot_medians_other <- map_dfr(1:1000, function(i) {
  sampled <- merged_df_residue %>%
    filter(site_class == "other") %>%
    slice_sample(n = n_sample)

  tibble(
    site_class = "other",
    median_resid = median(abs(sampled$loess_residual_avg), na.rm = TRUE),
    replicate = i
  )
})
```

```r
# Combine all
plot_df <- bind_rows(
  fixed_df %>% mutate(replicate = NA),
  boot_medians_other
)
head(plot_df)
```

```
## # A tibble: 6 x 3
##   site_class  median_resid replicate
##   <chr>              <dbl>     <int>
## 1 orthosteric         5.92        NA
## 2 orthosteric         4.15        NA
## 3 orthosteric         4.38        NA
## 4 orthosteric         4.99        NA
## 5 orthosteric         5.00        NA
## 6 orthosteric         4.12        NA
```

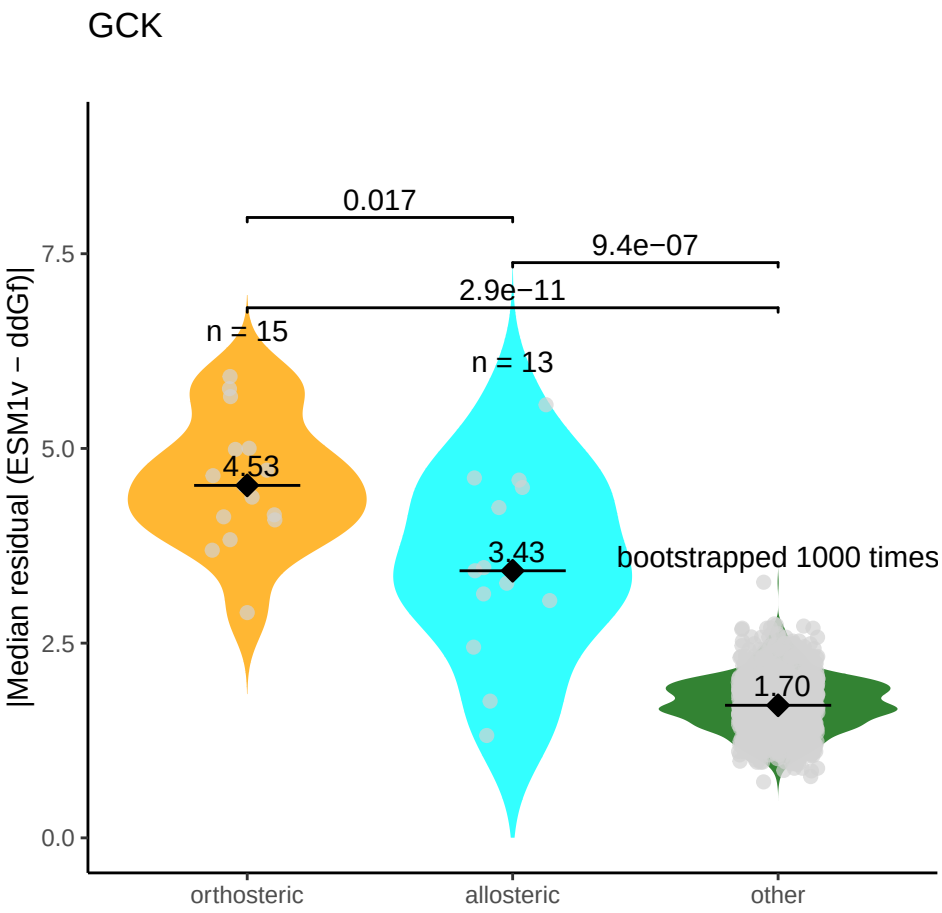
```r
# Labels
label_df <- plot_df %>%
  group_by(site_class) %>%
  summarise(
    n = n(),
    median_val = median(median_resid),
    y_max = max(median_resid),
    .groups = "drop"
  )

label_df <- label_df %>%
  mutate(n_label = case_when(
    site_class == "other" ~ "bootstrapped 1000 times",
    TRUE ~ paste0("n = ", n)
  ))

plot_df$site_class <- factor(plot_df$site_class, levels = c("orthosteric", "allosteric", "other"))

# Plot
p4 <- ggplot(plot_df, aes(x = site_class, y = median_resid, fill = site_class)) +
  geom_violin(data = plot_df,
              trim = FALSE, scale = "width", alpha = 0.8, color = NA) +

  geom_jitter(data = plot_df ,
              width = 0.15, size = 2, alpha = 0.7, color = "lightgrey") +

  stat_summary(fun = median, geom = "crossbar", width = 0.4, color = "black", fatten = 1) +
  stat_summary(fun = median, geom = "point", shape = 23, size = 2.5,
               fill = "black", color = "black", stroke = 0.7) +

 geom_text(
  data = label_df,
  aes(x = site_class, y = y_max * 1.1, label = n_label),
  inherit.aes = FALSE,
  size = 4) +
```

```
  geom_text(
    data = label_df,
    aes(x = site_class, y = median_val + 0.25, label = sprintf(" %.2f", median_val)),
    inherit.aes = FALSE,
    size = 4
  ) +

  geom_signif(
    comparisons = list(
      c("orthosteric", "other"),
      c("allosteric", "other"),
       c("allosteric", "orthosteric")
    ),
    test = "wilcox.test",
    map_signif_level = FALSE,
    step_increase = 0.1,
    tip_length = 0.01
  ) +

  labs(
    title = "GCK",
    subtitle = "",
    x = "",
    y = "|Median residual (ESM1v - ddGf)|"
  ) +
  scale_fill_manual(values = c(
    "orthosteric" = "orange",
    "allosteric" = "cyan",
    "other" = "darkgreen"
  )) +
  theme_classic() +
  theme(legend.position = "none") + ylim(0,9)

ggsave("/Users/xl7/Documents/0.Projects/01.protein-seq-evo-v1/figs/panels/fig2_gck_p4.pdf",
       plot = p4, width = 3, height = 4, dpi = 300)
```

```
## Warning: Removed 29 rows containing missing values or values outside the scale range
## (`geom_violin()`).
```

```
p4
```

```
## Warning: Removed 29 rows containing missing values or values outside the scale range
## (`geom_violin()`).
```




```
gck_abundance <- read.csv("/Users/xl7/Documents/0.Projects/01.protein-seq-evo-v1/data/vampseq/vampseq_d
gck_activity <- read.csv("/Users/xl7/Documents/0.Projects/01.protein-seq-evo-v1/data/vampseq/vampseq_dd
nrow(gck_abundance) #8396
```

```
## [1] 8396
```

```
nrow(gck_activity) #8570
```

```
## [1] 8570
```

```
gck_df <- gck_abundance %>% dplyr::select (mutant, DMS_score, DMS_score_bin) %>%
  dplyr::rename(DMS_score_abundance = DMS_score,
                DMS_score_bin_abundance = DMS_score_bin)
gck_df <- merge(gck_df, gck_activity, by = "mutant")
gck_df <- gck_df %>% dplyr::rename(DMS_score_activity = DMS_score,
                                   DMS_score_bin_activity = DMS_score_bin) %>%
  dplyr::select (mutant, DMS_score_abundance, DMS_score_bin_abundance,
                 DMS_score_activity, DMS_score_bin_activity)

gck_df <- gck_df %>%
  mutate(mutation_position = as.numeric(str_extract(mutant, "(?<=\\D)(\\d+)(?=\\D)")))
```

```
test_merged_df <- merge(gck_df, gck_pred$merged_data, by.x="mutant", by.y="variant")
nrow(test_merged_df) #8255
```

```
## [1] 8255
```

```
head(test_merged_df)
```

```
##   mutant DMS_score_abundance DMS_score_bin_abundance DMS_score_activity
## 1   A10C           1.0930201                       1          1.8238964
## 2   A10D           1.0294388                       1          0.5802219
## 3   A10E           1.1407103                       1          0.8284387
## 4   A10F           0.9642829                       1          2.0494065
## 5   A10G           1.2073064                       1          1.0260442
## 6   A10H           0.8375215                       1          0.6990930
##   DMS_score_bin_activity mutation_position  V1 Mutation ddG_pred pos wtAA mutAA
## 1                      1                10 181      A9C   0.0486   9    A     C
## 2                      1                10 182      A9D   0.3884   9    A     D
## 3                      1                10 183      A9E   0.0151   9    A     E
## 4                      1                10 184      A9F   0.3177   9    A     F
## 5                      1                10 185      A9G   0.5904   9    A     G
## 6                      1                10 186      A9H   0.2476   9    A     H
##   new_position     ESM1v
## 1           10 -6.882101
## 2           10 -5.333317
## 3           10 -4.009192
## 4           10 -6.812537
## 5           10 -4.622459
## 6           10 -6.842433
```

GCK exp residual

```
active_positions <- c(151:179, # disordered loop
                      151-153, 168-169, 204-206, 225-231, 254-258, 287, 290, # glucose-binding
                      78:85, 151, 169, 205, 225:229, 295:296, 331:333, 336, 410:416 # ATP-binding
)

fil_test_merged_df <- test_merged_df %>%
  filter(!mutation_position %in% active_positions)

# Fit a loess model using the filtered data
loess_fit <- loess(DMS_score_activity ~ DMS_score_abundance, data = fil_test_merged_df, span = 0.7, fami

# Predict fitted values for ALL data points using the loess model trained on fil_gck_df
test_merged_df$fitted_exp <- predict(loess_fit, newdata = test_merged_df)

# Calculate residuals for ALL points
test_merged_df$residuals_exp <- test_merged_df$fitted_exp - test_merged_df$DMS_score_activity
range(test_merged_df$residuals_exp) #-6.101563  1.887485
```

```
## [1] -6.102024  1.891487
```

```
# Fit a loess model using the filtered data
loess_fit_comp <- loess(ESM1v ~ ddG_pred, data = fil_test_merged_df, span = 0.7, family = "symmetric")

# Predict fitted values for ALL data points using the loess model trained on fil_gck_df
test_merged_df$fitted_comp <- predict(loess_fit_comp, newdata = test_merged_df)

# Calculate residuals for ALL points
test_merged_df$residuals_comp <- test_merged_df$ESM1v - test_merged_df$fitted_comp
sum(is.na(test_merged_df$residuals_comp)) #2
```

```
## [1] 2
```

```
head(test_merged_df)
```

```
##   mutant DMS_score_abundance DMS_score_bin_abundance DMS_score_activity
## 1   A10C           1.0930201                       1          1.8238964
## 2   A10D           1.0294388                       1          0.5802219
## 3   A10E           1.1407103                       1          0.8284387
## 4   A10F           0.9642829                       1          2.0494065
## 5   A10G           1.2073064                       1          1.0260442
## 6   A10H           0.8375215                       1          0.6990930
##   DMS_score_bin_activity mutation_position  V1 Mutation ddG_pred pos wtAA mutAA
## 1                      1                10 181      A9C   0.0486   9    A     C
## 2                      1                10 182      A9D   0.3884   9    A     D
## 3                      1                10 183      A9E   0.0151   9    A     E
## 4                      1                10 184      A9F   0.3177   9    A     F
## 5                      1                10 185      A9G   0.5904   9    A     G
## 6                      1                10 186      A9H   0.2476   9    A     H
##   new_position     ESM1v fitted_exp residuals_exp fitted_comp residuals_comp
## 1           10 -6.882101  0.8575707   -0.96632571   -9.261323       2.379223
## 2           10 -5.333317  0.8371116    0.25688971   -9.790254       4.456937
## 3           10 -4.009192  0.8742179    0.04577921   -9.241057       5.231865
## 4           10 -6.812537  0.8155700   -1.23383659   -9.629733       2.817196
## 5           10 -4.622459  0.8991870   -0.12685722  -10.401659       5.779200
## 6           10 -6.842433  0.7598041    0.06071114   -9.505593       2.663160
```

```
test_merged_df_residue <- test_merged_df %>%
  group_by(new_position) %>%
  summarise(
    comp_residual_avg = median(residuals_comp, na.rm = TRUE),
    exp_residual_avg = median(residuals_exp, na.rm = TRUE))

#cor.test(test_merged_df_residue$comp_residual_avg, test_merged_df_residue$exp_residual_avg, method="sp
#   Spearman's rank correlation rho
#
# data:  test_merged_df_residue$comp_residual_avg and test_merged_df_residue$exp_residual_avg
# S = 24507826, p-value < 2.2e-16
# alternative hypothesis: true rho is not equal to 0
# sample estimates:
#       rho
```

```
# -0.4815458

active_sites <- c(151:179, # disordered loop
                  78:85, 151, 169, 205, 225:229, 295:296, 331:333, 336, 410:416) # ATP-binding

binding_sites <- c(151:153, 168:169, 204:206, 225:231, 254:258, 287, 290) # glucose-binding)

test_merged_df_residue$site_type <- "Non-orthosteric site"
test_merged_df_residue$site_type[test_merged_df_residue$new_position %in% active_sites] <- "ATP-binding
test_merged_df_residue$site_type[test_merged_df_residue$new_position %in% binding_sites] <- "Glucose-bi

test_merged_df_residue_active <- test_merged_df_residue %>% filter(site_type == "ATP-binding site")
nrow(test_merged_df_residue_active) #45
```

```
## [1] 45
```

```
#cor.test(test_merged_df_residue_active$comp_residual_avg, test_merged_df_residue_active$exp_residual_a
#   Spearman's rank correlation rho
#
# data:  test_merged_df_residue_active$comp_residual_avg and test_merged_df_residue_active$exp_residual
# S = 15928, p-value = 0.7472
# alternative hypothesis: true rho is not equal to 0
# sample estimates:
#         rho
# -0.04927536

test_merged_df_residue_glu <- test_merged_df_residue %>% filter(site_type == "Glucose-binding site")
nrow(test_merged_df_residue_glu) #22
```

```
## [1] 22
```

```
#cor.test(test_merged_df_residue_glu$comp_residual_avg, test_merged_df_residue_glu$exp_residual_avg, me
#   Spearman's rank correlation rho
#
# data:  test_merged_df_residue_glu$comp_residual_avg and test_merged_df_residue_glu$exp_residual_avg
# S = 2836, p-value = 0.003688
# alternative hypothesis: true rho is not equal to 0
# sample estimates:
#       rho
# -0.6013552

test_merged_df_residue_other <- test_merged_df_residue %>% filter(site_type == "Non-orthosteric site")
nrow(test_merged_df_residue_other) #396
```

```
## [1] 396
```

```
#cor.test(test_merged_df_residue_other$comp_residual_avg, test_merged_df_residue_other$exp_residual_avg
#   Spearman's rank correlation rho
#
# data:  test_merged_df_residue_other$comp_residual_avg and test_merged_df_residue_other$exp_residual_a
# S = 14588154, p-value < 2.2e-16
```

```
# alternative hypothesis: true rho is not equal to 0
# sample estimates:
#        rho
# -0.4095121
```

```
#nrow(test_merged_df_residue) #463
#cor.test(test_merged_df_residue$comp_residual_avg, test_merged_df_residue$exp_residual_avg, method="sp
p1 <- ggplot(test_merged_df_residue, aes(x = exp_residual_avg , y = comp_residual_avg, color = site_typ
  geom_point(size = 2, alpha = 0.5) +
  labs(
    title = "GCK: 463 residues",
    subtitle = "Spearman's rho = -0.48",
    x = "Experimental median activity-abundance residual",
    y = "Predicted median ESM1v-TMPNN ddGf residual",
    color = "") +
  theme_classic() +
    scale_color_manual(values = c(
    "Non-orthosteric site" = "darkgreen",
    "ATP-binding site" = "cyan",
    "Glucose-binding site" = "orange"
  )) + theme(legend.position = "none") +
  geom_vline(xintercept = 0, linetype = "dashed", linewidth = 0.5, color = "grey") +
  geom_hline(yintercept = 0, linetype = "dashed", linewidth = 0.5, color = "grey") +
  xlim(-3,2) + ylim(-5,5)


#nrow(test_merged_df_residue_active) #45
#cor.test(test_merged_df_residue_active$comp_residual_avg, test_merged_df_residue_active$exp_residual_a
p2 <- ggplot(test_merged_df_residue_active, aes(x = exp_residual_avg , y = comp_residual_avg, color = si
  geom_point(size = 2, alpha = 0.7) +
  labs(
    title = "45 ATP-binding sites",
    subtitle = "Spearman's rho = -0.05",
    x = "Experimental median activity-abundance residual",
    y = "Predicted median ESM1v-TMPNN ddGf residual",
    color = "") +
  theme_classic() +
    scale_color_manual(values = c(
    "Non-orthosteric site" = "darkgreen",
    "ATP-binding site" = "cyan",
    "Glucose-binding site" = "orange"
  )) + theme(legend.position = "none") +
  geom_text_repel(aes(label = new_position)) +
  geom_vline(xintercept = 0, linetype = "dashed", linewidth = 0.5, color = "grey") +
  geom_hline(yintercept = 0, linetype = "dashed", linewidth = 0.5, color = "grey") +
  xlim(-3,2) + ylim(-5,5)


#nrow(test_merged_df_residue_glu) #22
#cor.test(test_merged_df_residue_glu$comp_residual_avg, test_merged_df_residue_glu$exp_residual_avg, me
p3 <- ggplot(test_merged_df_residue_glu, aes(x = exp_residual_avg , y = comp_residual_avg, color = site_
  geom_point(size = 2, alpha = 0.7) +
  labs(
```

```
    title = "22 glucose-binding sites",
    subtitle = "Spearman's rho = -0.60",
    x = "Experimental median activity-abundance residual",
    y = "Predicted median ESM1v-TMPNN ddGf residual",
    color = "") +
  theme_classic() +
    scale_color_manual(values = c(
    "Non-orthosteric site" = "darkgreen",
    "ATP-binding site" = "cyan",
    "Glucose-binding site" = "orange"
  )) + theme(legend.position = "none") +
  geom_text_repel(aes(label = new_position)) +
  geom_vline(xintercept = 0, linetype = "dashed", linewidth = 0.5, color = "grey") +
  geom_hline(yintercept = 0, linetype = "dashed", linewidth = 0.5, color = "grey") +
  xlim(-3,2) + ylim(-5,5)

#nrow(test_merged_df_residue_other) #396
cor.test(test_merged_df_residue_other$comp_residual_avg, test_merged_df_residue_other$exp_residual_avg,
```

```
##
##  Spearman's rank correlation rho
##
## data:  test_merged_df_residue_other$comp_residual_avg and test_merged_df_residue_other$exp_residual_a
## S = 14588154, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.4095121
```

```
# helix 13: https://cspec.genome.network/cspec/ui/svi/doc/GN086
# 444-456
p4 <- ggplot(test_merged_df_residue_other, aes(x = exp_residual_avg , y = comp_residual_avg, color = si
  geom_point(size = 2, alpha = 0.3) +
  labs(
    title = "396 non-orthosteric sites",
    subtitle = "Spearman's rho = -0.41",
    x = "Experimental median activity-abundance residual",
    y = "Predicted median ESM1v-TMPNN ddGf residual",
    color = "") +
  theme_classic() +
    scale_color_manual(values = c(
    "Non-orthosteric site" = "darkgreen",
    "ATP-binding site" = "cyan",
    "Glucose-binding site" = "orange"
  )) + theme(legend.position = "none") +
  geom_text_repel(data = test_merged_df_residue_other %>% filter(new_position %in% c(444:456)), aes(labe
  geom_vline(xintercept = 0, linetype = "dashed", linewidth = 0.5, color = "grey") +
  geom_hline(yintercept = 0, linetype = "dashed", linewidth = 0.5, color = "grey") +
  xlim(-3,2) + ylim(-5,5)

p1 <- ggMarginal(
  p1,
  type = "density",
```

```
  margins = "both",
  groupColour = FALSE,
  groupFill = FALSE,
  size = 10,
  colour = "grey",
  fill = "lightgrey"
)
```

```
## Warning: Removed 26 rows containing missing values or values outside the scale range
## (`geom_point()`).
## Removed 26 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
p2 <- ggMarginal(
  p2,
  type = "density",
  margins = "both",
  groupColour = FALSE,
  groupFill = FALSE,
  size = 10,
  colour = "grey",
  fill = "lightgrey"
)
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).

## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_text_repel()`).

## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).

## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_text_repel()`).
```

```
p3 <- ggMarginal(
  p3,
  type = "density",
  margins = "both",
  groupColour = FALSE,
  groupFill = FALSE,
  size = 10,
  colour = "grey",
  fill = "lightgrey"
)
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_point()`).

## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_text_repel()`).
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_text_repel()`).
```

```
p4 <- ggMarginal(
  p4,
  type = "density",
  margins = "both",
  groupColour = FALSE,
  groupFill = FALSE,
  size = 10,
  colour = "grey",
  fill = "lightgrey"
)
```

```
## Warning: Removed 20 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_text_repel()`).
```

```
## Warning: Removed 20 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_text_repel()`).
```

```
p5 <- plot_grid(p1,p2,p3,p4, ncol=4,nrow=1)
ggsave("/Users/xl7/Documents/0.Projects/01.protein-seq-evo-v1/figs/panels/fig2_gck_p5.pdf",
       plot = p5, width = 12, height = 3, dpi = 300)
```

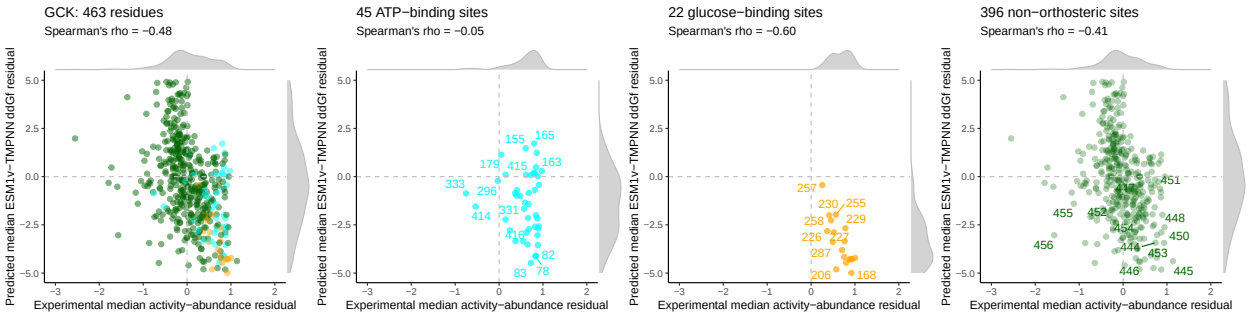
```
## Warning: ggrepel: 38 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

```
## Warning: ggrepel: 15 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

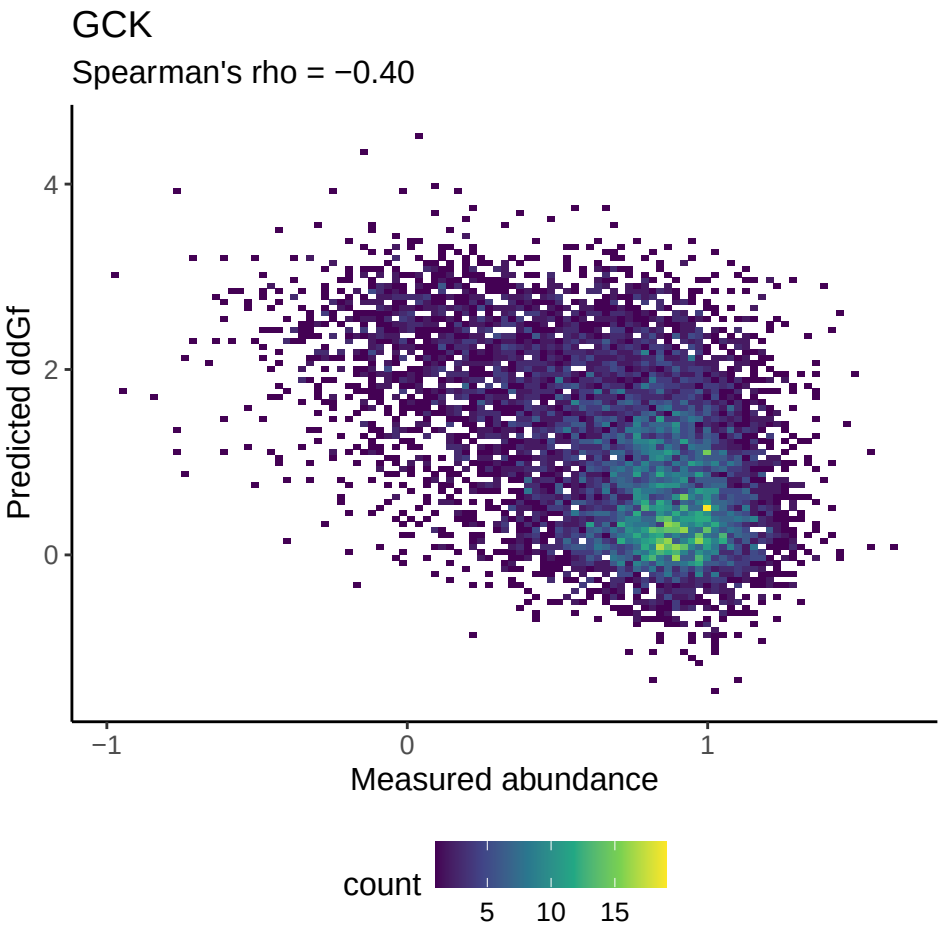
```
p5
```

```
## Warning: ggrepel: 30 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

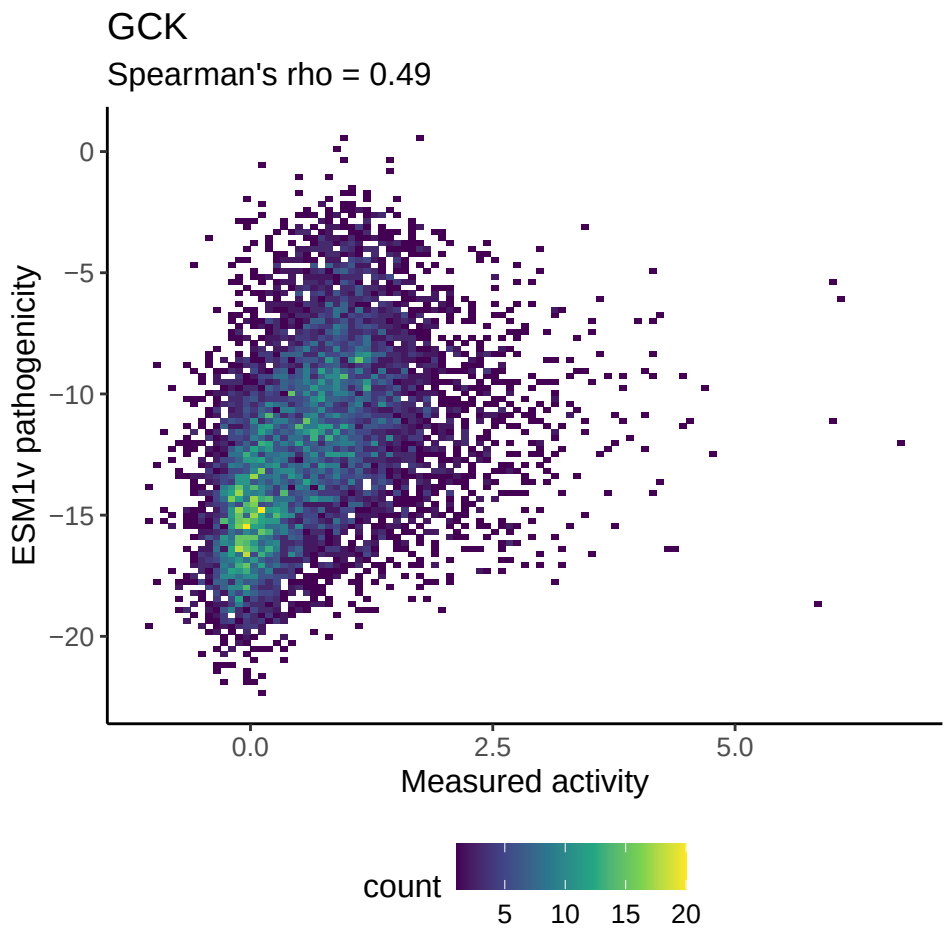
```
## Warning: ggrepel: 8 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```




```
#cor.test(test_merged_df$ddG_pred, test_merged_df$DMS_score_abundance, method = "spearman")#-0.404267

p6 <- ggplot(test_merged_df, aes(x = DMS_score_abundance, y = ddG_pred) ) +
  geom_bin2d(bins = 100) +
  scale_fill_continuous(type = "viridis") +
  theme_classic() +
  labs(
    x = "Measured abundance",
    y = "Predicted ddGf",
    title = "GCK",
    subtitle = "Spearman's rho = -0.40"
  ) +
  theme(
    text = element_text(size = 12),
    legend.position = ("bottom")
  )
ggsave("/Users/xl7/Documents/0.Projects/01.protein-seq-evo-v1/figs/panels/fig2_gck_p6.pdf",
       plot = p6, width = 3, height = 4, dpi = 300)
p6
```




```
#cor.test(test_merged_df$ESM1v, test_merged_df$DMS_score_activity, method = "spearman")#0.4867936

p7 <- ggplot(test_merged_df, aes(x = DMS_score_activity, y = ESM1v) ) +
  geom_bin2d(bins = 100) +
  scale_fill_continuous(type = "viridis") +
  theme_classic() +
  labs(
    x = "Measured activity",
    y = "ESM1v pathogenicity",
    title = "GCK",
    subtitle = "Spearman's rho = 0.49"
  ) +
  theme(
    text = element_text(size = 12),
    legend.position = ("bottom")
  )
ggsave("/Users/xl7/Documents/0.Projects/01.protein-seq-evo-v1/figs/panels/fig2_gck_p7.pdf",
       plot = p7, width = 3, height = 4, dpi = 300)
p7
```


GCK
Spearman's rho = 0.49
ESM1v pathogenicity
0
−5
−10
−15
−20
0.0
2.5
5.0
Measured activity
count
5
10
15
20