deep_blast_plots

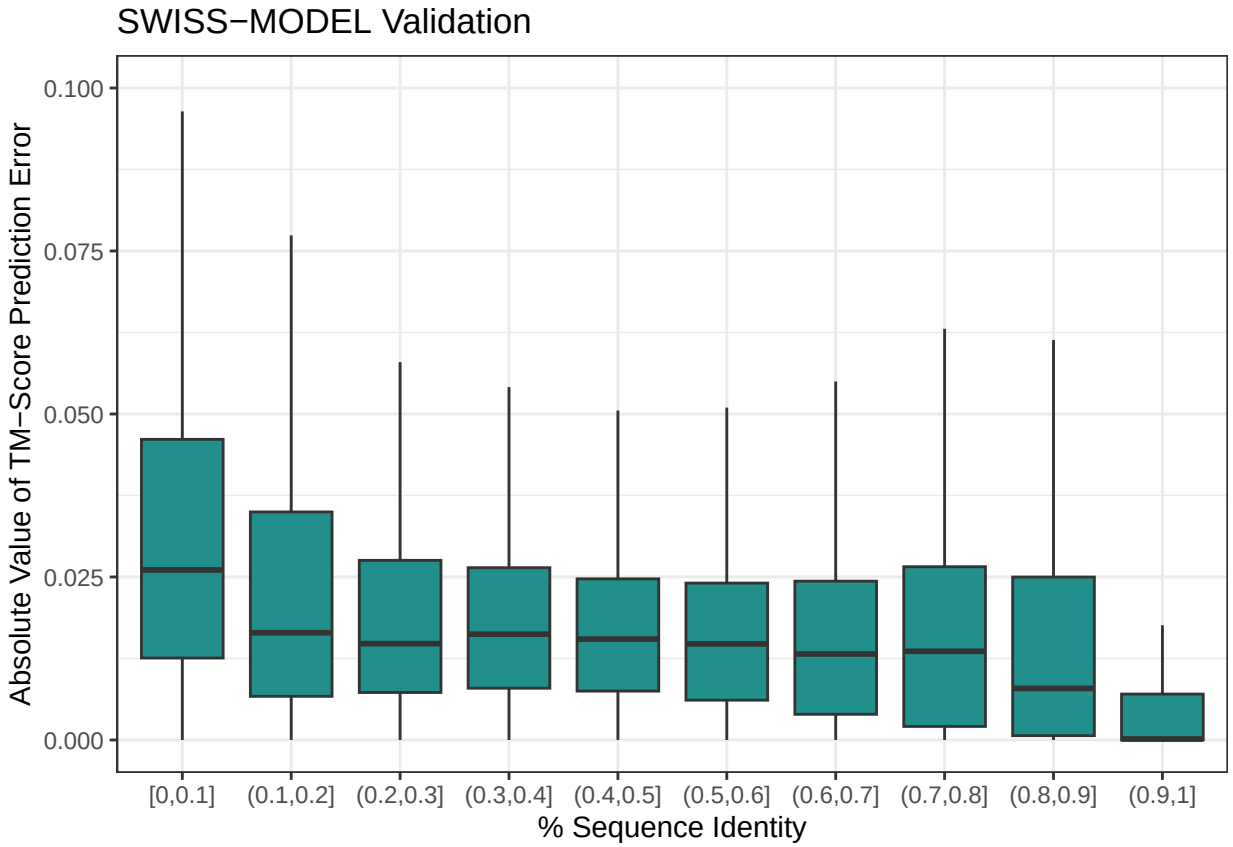
```
library(tidyverse)
library(corrr)
library(viridis)
library(tidyr)
library(cowplot)
library(Rtsne)

set.seed(42)
```

```
#Read in files

#embedding vector scores
embedding_scores <- read_tsv("~/deepblast_data_rplots/scores.tsv")

#TM score predictions with sequence similarity: SWISS
tm_preds_swiss <- read_csv("~/deepblast_data_rplots/predicted_tms_swiss_test_set_1M_w_seqid.csv")

#TM score predictions out of fold, CATH
tm_preds_out_of_fold_cath <- read_csv("~/deepblast_data_rplots/predicted_tms_cath_left_out_folds_500K.cs

#TM score predictions pairs left out, CATH
tm_preds_pairs_left_out_cath <- read_csv("~/deepblast_data_rplots/predicted_tms_cath_left_pairs_out_scra

#TM score predictions domains left out, CATH
tm_preds_domains_left_out_cath <- read_csv("~/deepblast_data_rplots/predicted_tms_cath_left_domains_out_

#Malidup predictions
malidup_predictions_cath <- read_csv("~/deepblast_data_rplots/malidup_tm_predictions_with_ground_file_ca

#malidup benchmarks
malidup_benchmarks <- read_csv("~/deepblast_data_rplots/tm_alignment_scores.csv")

#tsne data
tsne_meta_bench <- read_csv("~/deepblast_data_rplots/tsne_w_meta_cath_dan_test_cath_model.csv")

#Bacteriocin: Colabfold
colabfold_benchmarking <- read_csv('~/deepblast_data_rplots/tm_scores_with_predictions_and_meta_colabfol

#Bacteriocin: Omegafold
omegafold_benchmarking <- read_tsv("~/deepblast_data_rplots/tm_align_outputs_omega_processed.tsv")
omegafold_benchmarking <-
  omegafold_benchmarking %>%
  mutate(omega_fold_tm_max = pmax(omegafold_benchmarking$tm_score_1, omegafold_benchmarking$tm_score_2))
  select(chain_1, chain_2, omega_fold_tm_max)

#Bacteriocin: ESM
esm_benchmarking <- read_tsv("~/deepblast_data_rplots/tm_align_outputs_ESM_processed.tsv")
```

```
esm_benchmarking  <-
  esm_benchmarking  %>%
  mutate(esm_tm_max = pmax(esm_benchmarking$tm_score_1, esm_benchmarking$tm_score_2)) %>%
  select(chain_1, chain_2, esm_tm_max)

#TM-Vec performance
deep_blast_speed_encoding <- readxl::read_xlsx("~/deepblast_data_rplots/deep_blast_search_speed.xlsx", s
deep_blast_speed_query_database <- readxl::read_xlsx("~/deepblast_data_rplots/deep_blast_search_speed.xl

#TM-Vec embedding vectors
tsne_data <- read_csv("~/deepblast_data_rplots/embeddings_cath_dan_test_cath_model.csv")
meta_data_tsne <- read_csv("~/deepblast_data_rplots/meta_cath_dan_test_cath_model.csv")

#ProtTrans embedding vectors
protrans_names <- read_csv("~/deepblast_data_rplots/protrans_protein_names.csv")
protrans_vecs <- read_csv("~/deepblast_data_rplots/protrans_protein_vectors.csv")
```

SwissProt Validation: % Sequence Identity versus Absolute Value of TM-Score Prediction Error

```
tm_preds_swiss_matrix_seq_tm_info <-
  tm_preds_swiss %>%
  mutate(prediction_error = abs(Predicted_TM_score - Ground_truth_TM_score))

tm_preds_swiss_matrix_seq_tm_info_validation <-
  tm_preds_swiss_matrix_seq_tm_info %>%
  mutate(
    seq_id_breaks = cut(seq_id, breaks = seq(from = 0, to = 1, by = .1), include.lowest = TRUE),
    filling = "filling"
  ) %>%
  ggplot(aes(seq_id_breaks, prediction_error, fill = filling)) +
  geom_boxplot(outlier.shape = NA) +
  coord_cartesian(ylim =  c(0, .1)) +
  labs(x = "% Sequence Identity", y = "Absolute Value of TM-Score Prediction Error" , title = "SWISS-MOD
  theme_bw() +
  theme(legend.position="none") +
  scale_fill_viridis_d(begin=.5)

tm_preds_swiss_matrix_seq_tm_info_validation
```



SwissProt Validation: Correlation over different sequence similarity thresholds

```
'
library(tidymodels)

all_cors = tibble()
seqid_chunks <- seq(from = 0, to = 1, by = .01)

for(i in seqid_chunks){
  table_i <-
    tm_preds_swiss_matrix_seq_tm_info %>%
    filter(seq_id <= i) %>%
    select(Ground_truth_TM_score, Predicted_TM_score)

  cor_i <-
    cor.test(table_i$Ground_truth_TM_score, table_i$Predicted_TM_score, method = "pearson") %>%
    tidy()

  all_cors <- rbind(all_cors, cor_i)
}

combined_cors <-
  all_cors %>%
  mutate(sequence_similarity_threshold = seqid_chunks)

combined_cors_plot <-
  combined_cors %>%
```

```
  ggplot(aes(sequence_similarity_threshold, estimate)) +
  geom_line() +
  geom_point(size = .5) +
  geom_ribbon(aes(ymin = conf.low, ymax = conf.high), alpha = 0.1) +
  coord_cartesian(ylim = c(0.0, 1.0)) +
  labs(x = "Sequence similarity threshold", y = "Correlation") +
  theme_bw() +
  scale_colour_viridis_d()
'
```

SwissProt validation: TM-score prediction errors (quartiles)

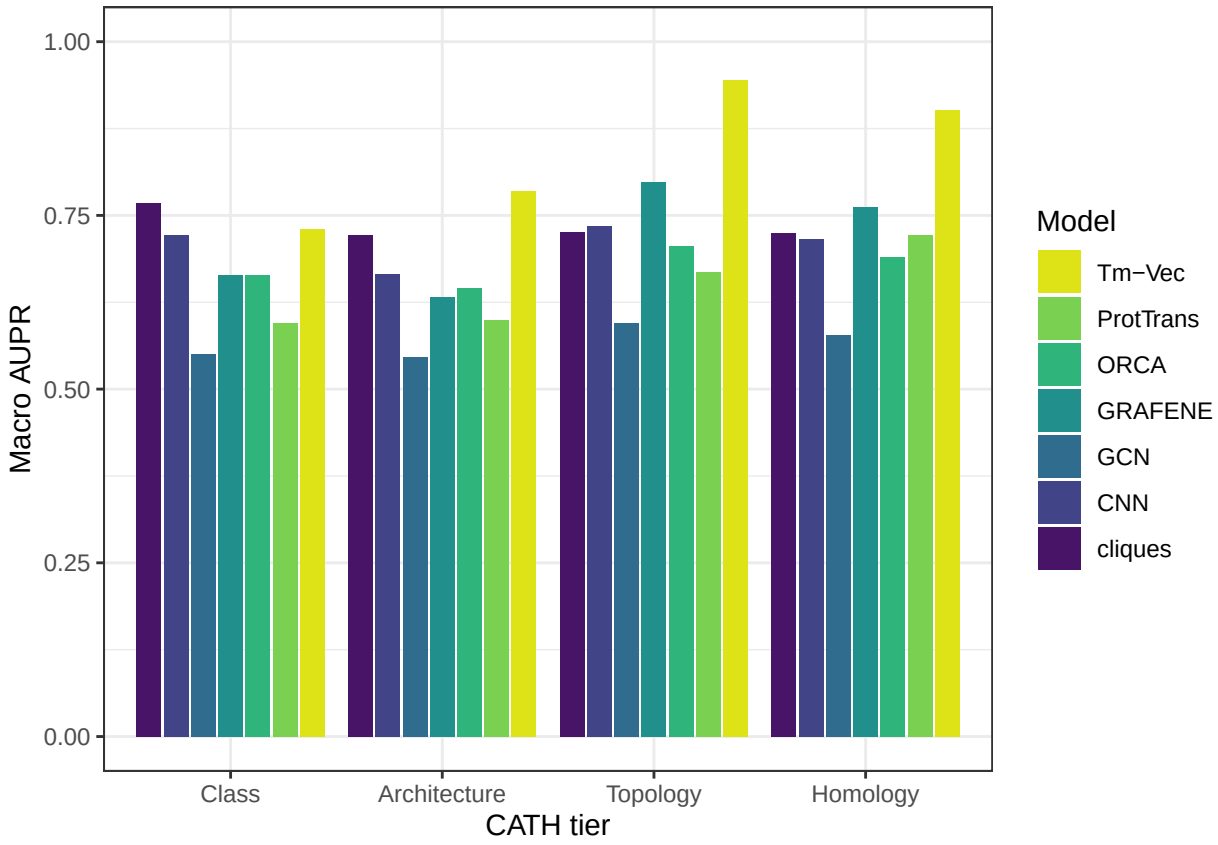
```
tm_preds_swiss_matrix_seq_tm_info %>%
  pull(prediction_error) %>%
  summary()
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000000 0.008896 0.019962 0.028525 0.037536 0.838238
```

Benchmarking TM-Vec against other structure/sequence embedding methods: Macro AUPR

```
tier_system <-
  tibble(tier = c("CLASS", "ARCH", "TOPOL", "HOMOL")) %>%
  mutate(model_order = row_number())

aupr_plot <-
  embedding_scores %>%
  filter(
    metric == "macro_aupr",
    !model %in% c("cnn6", "gcn6", "random")
  ) %>%
  left_join(tier_system, by = "tier") %>%
  mutate(
    tier = if_else(tier == "CLASS", "Class", tier),
    tier = if_else(tier == "ARCH", "Architecture", tier),
    tier = if_else(tier == "TOPOL", "Topology", tier),
    tier = if_else(tier == "HOMOL", "Homology", tier)
  ) %>%
  mutate(model = if_else(model == "deepblast", "TM-Vec", model)) %>%
  mutate(
    model = fct_reorder(model, value),
    tier = fct_reorder(tier, model_order)
  ) %>%
  mutate(
    model = str_to_title(model)
  ) %>%
  rename(Model = model) %>%
  mutate(
    Model = if_else(Model == "Cnn10", "CNN",
            if_else(Model == "Grafene6", "GRAFENE",
            if_else(Model == "Cliques6", "cliques",
            if_else(Model == "Gcn10", "GCN",
            if_else(Model == "Protrans", "ProtTrans",
            if_else(Model == "Orca6", "ORCA", Model)))))) %>%
  ggplot(aes(tier, value, fill = Model)) +
  geom_bar(stat="Identity", position = "dodge2") +
```

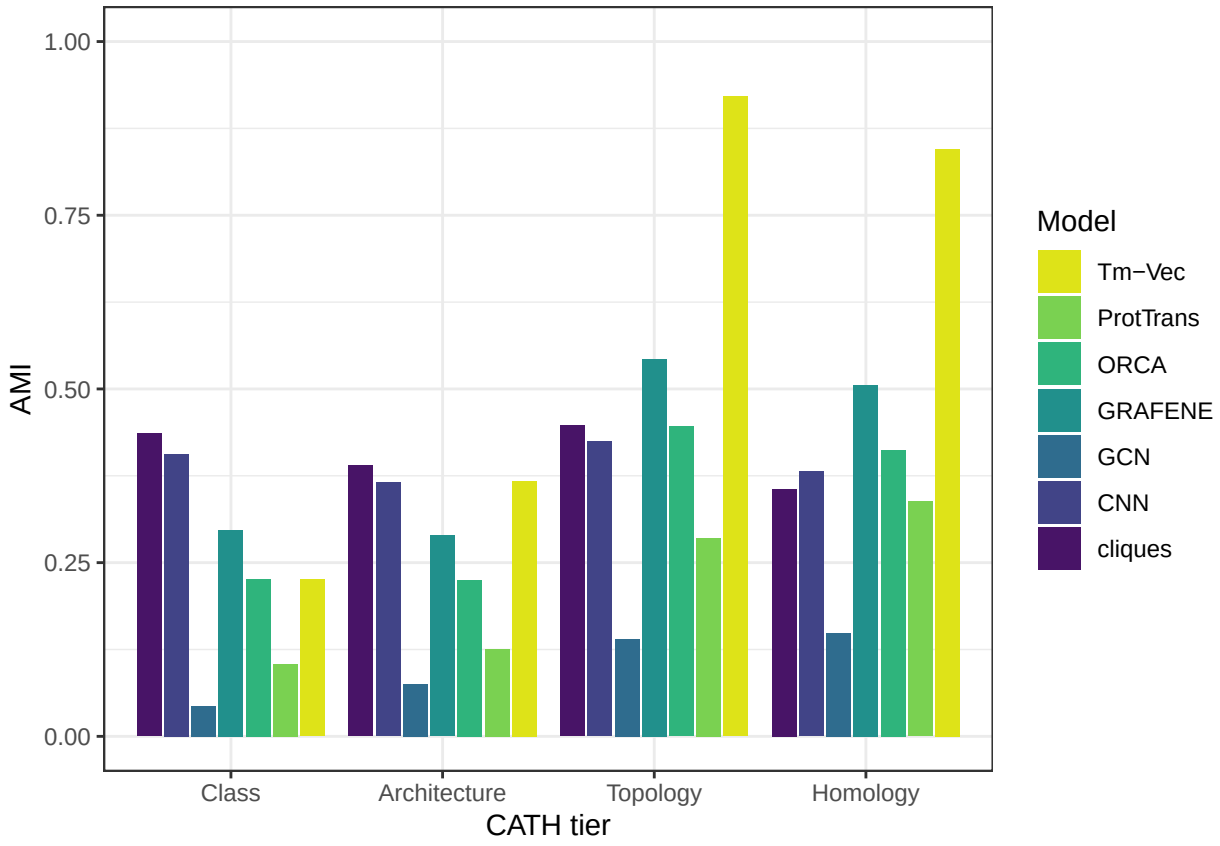
```
  theme_bw() +
  scale_fill_viridis_d(begin = .05, end=.95) +
  guides(fill=guide_legend(reverse = TRUE)) +
  labs(x = "CATH tier", y = "Macro AUPR") +
  coord_cartesian(ylim=c(0,1.0))


aupr_plot
```



Benchmarking TM-Vec against other structure/sequence embedding methods: AMI

```
tier_system <-
  tibble(tier = c("CLASS", "ARCH", "TOPOL", "HOMOL")) %>%
  mutate(model_order = row_number())

embedding_scores_plot <-
  embedding_scores %>%
  filter(
    metric == "ami",
    !model %in% c("cnn6", "gcn6", "random")
  ) %>%
  left_join(tier_system, by = "tier") %>%
    mutate(
    tier = if_else(tier == "CLASS", "Class", tier),
    tier = if_else(tier == "ARCH", "Architecture", tier),
    tier = if_else(tier == "TOPOL", "Topology", tier),
    tier = if_else(tier == "HOMOL", "Homology", tier)
```

```
  ) %>%
  mutate(model = if_else(model == "deepblast", "TM-Vec", model)) %>%
  mutate(
    model = fct_reorder(model, value),
    tier = fct_reorder(tier, model_order)
  ) %>%
  mutate(
    model = str_to_title(model)
  ) %>%
  rename(Model = model) %>%
  mutate(
    Model = if_else(Model == "Cnn10", "CNN",
            if_else(Model == "Grafene6", "GRAFENE",
            if_else(Model == "Cliques6", "cliques",
            if_else(Model == "Gcn10", "GCN",
            if_else(Model == "Protrans", "ProtTrans",
            if_else(Model == "Orca6", "ORCA", Model)))))) %>%
  ggplot(aes(tier, value, fill = Model)) +
  geom_bar(stat="Identity", position = "dodge2") +
  theme_bw() +
  scale_fill_viridis_d(begin = .05, end=.95) +
   guides(fill=guide_legend(reverse = TRUE)) +
  labs(x = "CATH tier", y = "AMI") +
  coord_cartesian(ylim=c(0,1.0))

embedding_scores_plot
```



Compare AUPR of TM-Vec versus other methods

```
embedding_scores %>%
  filter(
    metric == "macro_aupr",
    !model %in% c("cnn6", "gcn6", "random")
  ) %>%
  left_join(tier_system, by = "tier") %>%
  mutate(
    tier = if_else(tier == "CLASS", "Class", tier),
    tier = if_else(tier == "ARCH", "Architecture", tier),
    tier = if_else(tier == "TOPOL", "Topology", tier),
    tier = if_else(tier == "HOMOL", "Homology", tier)
  ) %>%
  mutate(
    model = fct_reorder(model, value),
    tier = fct_reorder(tier, model_order)
  ) %>%
  mutate(encoding_data = if_else(model %in% c("protrans", "deepblast"), "Sequence", "Structure")) %>%
  filter(model %in% c("protrans", "deepblast", "grafene6")) %>%
  filter(tier == "Topology")
```

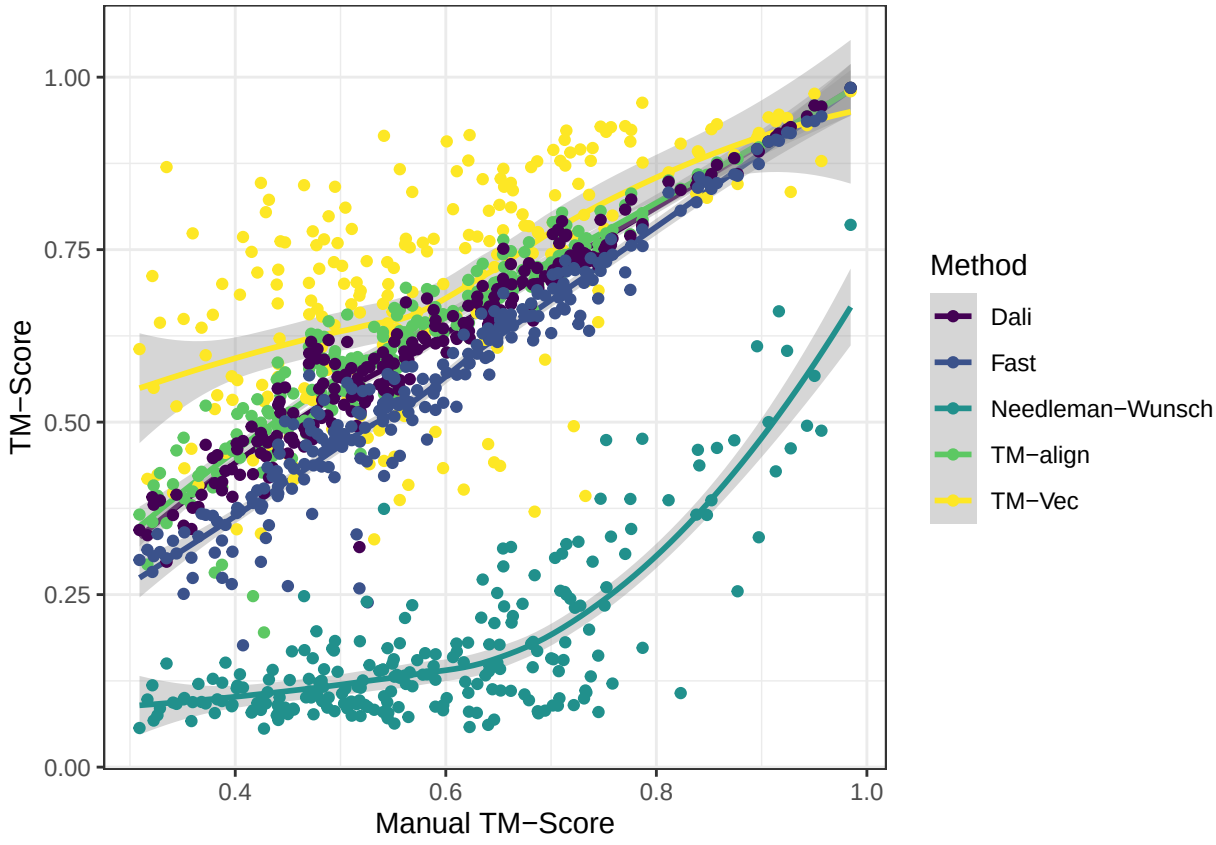
```
## # A tibble: 3 x 9
##   index benchmark_name     tier     metric   model channel value model~1 encod~2
##   <dbl> <chr>              <fct>    <chr>    <fct> <chr>   <dbl>   <int> <chr>
## 1   132 TripletScoringAUPR Topology macro_a~ deep~ featur~ 0.944       3 Sequen~
## 2   156 TripletScoringAUPR Topology macro_a~ graf~ normed  0.798       3 Struct~
## 3   204 TripletScoringAUPR Topology macro_a~ prot~ featur~ 0.668       3 Sequen~
## # ... with abbreviated variable names 1: model_order, 2: encoding_data
```

Malidup predictions

```
mal_preds_fig <-
  malidup_predictions_cath %>%
  select(Name, Predicted_TM) %>%
  left_join(malidup_benchmarks, by = c("Name" = "index")) %>%
  gather(-Name, -manual, key = "key", value = "value") %>%
  mutate(
    key = str_to_title(key),
    key = if_else(key == "Predicted_tm", "DeepBlast", key),
    key = if_else(key == "Deepblast", "DeepBlast aligner", key),
    key = if_else(key == "DeepBlast", "TM-Vec", key),
    key = if_else(key == "Tm", "TM-align", key)
  ) %>%
  filter(key != "DeepBlast aligner") %>%
  ggplot(aes(manual, value, color = key)) +
  geom_smooth() +
  geom_point() +
  theme_bw() +
  labs(color = "Method", x = "Manual TM-Score", y = "TM-Score") +
  scale_color_viridis_d()

mal_preds_fig
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

```
## Warning: Removed 35 rows containing non-finite values (`stat_smooth()`).
## Warning: Removed 35 rows containing missing values (`geom_point()`).
```



Plot alongside DeepBLAST

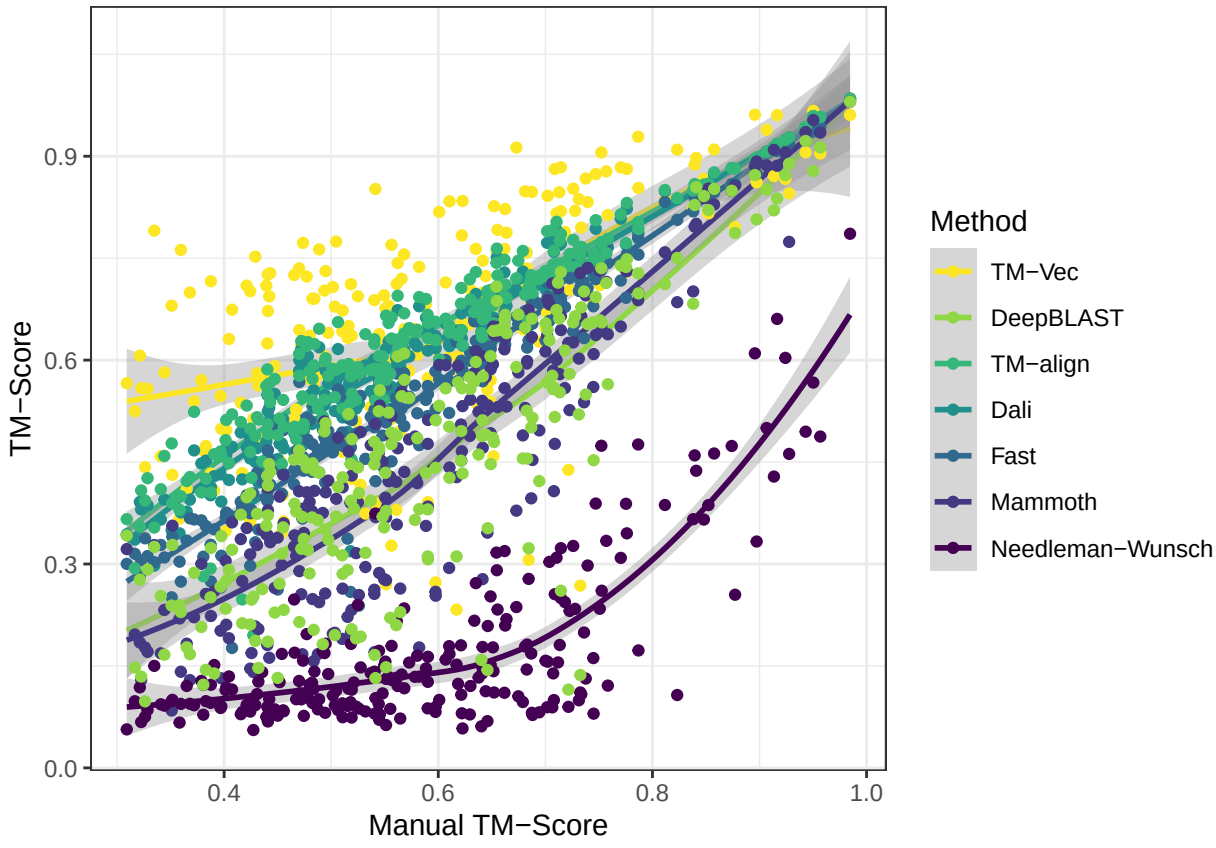
```
mal_preds_fig <-
  read_csv("~/malidup_sequences_and_tm_scores.csv") %>%
  select(Name, Predicted_TM = tm_score_predictions) %>%
  left_join(
    read_csv("~/Downloads/tm_scores_deepblast.csv"), by = c("Name" = "index")
  ) %>%
  gather(-Name, -manual, key = "key", value = "value") %>%
  mutate(
    key = str_to_title(key),
    key = if_else(key == "Predicted_tm", "TM-Vec", key),
    key = if_else(key == "Deepblast", "DeepBLAST", key),
    key = if_else(key == "Tm", "TM-align", key)
  ) %>%
  mutate(
    key = factor(key,
    levels = c("TM-Vec", "DeepBLAST", "TM-align", "Dali", "Fast", "Mammoth", "Needleman-Wunsch"))
  ) %>%
  filter(!is.na(manual)) %>%
  ggplot(aes(manual, value, color = key)) +
  geom_smooth() +
  geom_point() +
  theme_bw() +
```

```
  labs(color = "Method", x = "Manual TM-Score", y = "TM-Score") +
  scale_color_viridis_d(direction = -1)
```

```
## Rows: 241 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): Name, Sequence 1, Sequence 2, Protein_pair
## dbl (7): Tm_score1, TM_score2, tm_max, tm_score_predictions, diff, diff_cath...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 234 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): index
## dbl (7): manual, dali, fast, tm, mammoth, needleman-wunsch, deepblast
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
mal_preds_fig
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Plot with SW

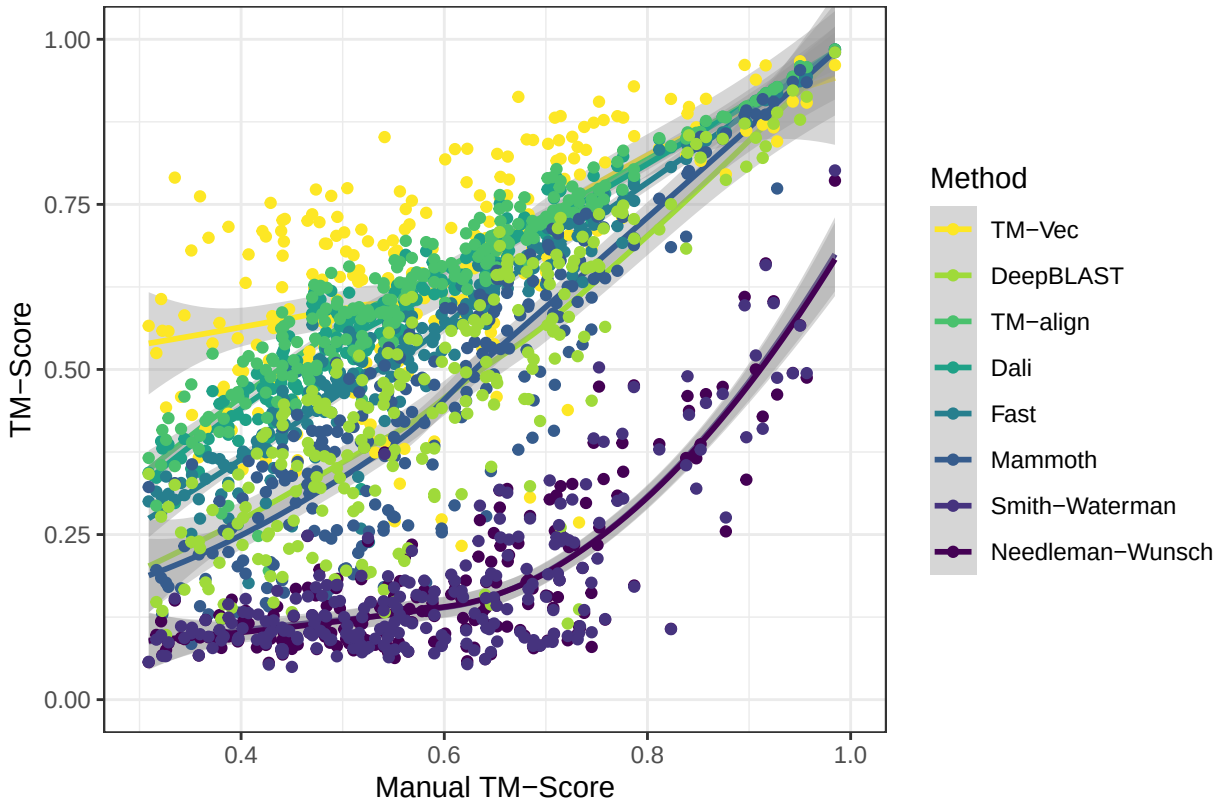
```
mal_preds_fig_sw <-
  read_csv("~/deepblast_data_rplots/malidup_sequences_and_tm_scores.csv") %>%
```

```
  select(Name, Predicted_TM = tm_score_predictions) %>%
  left_join(
    read_csv("~/deepblast_data_rplots/tm_scores_deepblast.csv"), by = c("Name" = "index")
  ) %>%
  left_join(
    read_csv("~/deepblast_data_rplots/smithwaterman.csv") %>%
      select(`Smith-Waterman` = TM, dir),
    by = c("Name" = "dir")
  ) %>%
  gather(-Name, -manual, key = "key", value = "value") %>%
  mutate(
    key = str_to_title(key),
    key = if_else(key == "Predicted_tm", "TM-Vec", key),
    key = if_else(key == "Deepblast", "DeepBLAST", key),
    key = if_else(key == "Tm", "TM-align", key)
  ) %>%
  mutate(
    key = factor(key,
    levels = c("TM-Vec", "DeepBLAST", "TM-align", "Dali", "Fast", "Mammoth", "Smith-Waterman", "Needlema
  ) %>%
  filter(!is.na(manual)) %>%
  ggplot(aes(manual, value, color = key)) +
  geom_smooth() +
  geom_point() +
  theme_bw() +
  coord_fixed(ratio = .65, ylim = c(0, 1), xlim = c(.3, 1.00)) +
  labs(color = "Method", x = "Manual TM-Score", y = "TM-Score") +
  scale_color_viridis_d(direction = -1)
```

```
## Rows: 241 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): Name, Sequence 1, Sequence 2, Protein_pair
## dbl (7): Tm_score1, TM_score2, tm_max, tm_score_predictions, diff, diff_cath...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 234 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): index
## dbl (7): manual, dali, fast, tm, mammoth, needleman-wunsch, deepblast
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## New names:
## Rows: 234 Columns: 28
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (12): Unnamed: 0, 1, manual, mammoth, dir, file1, file2, fast, tm, dali,...
## dbl (16): ...1, TM, PSI, aPSI, oPSI, rPSI, cRMS, aRMS, oRMS, aSeq_ident, oSe...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
mal_preds_fig_sw
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

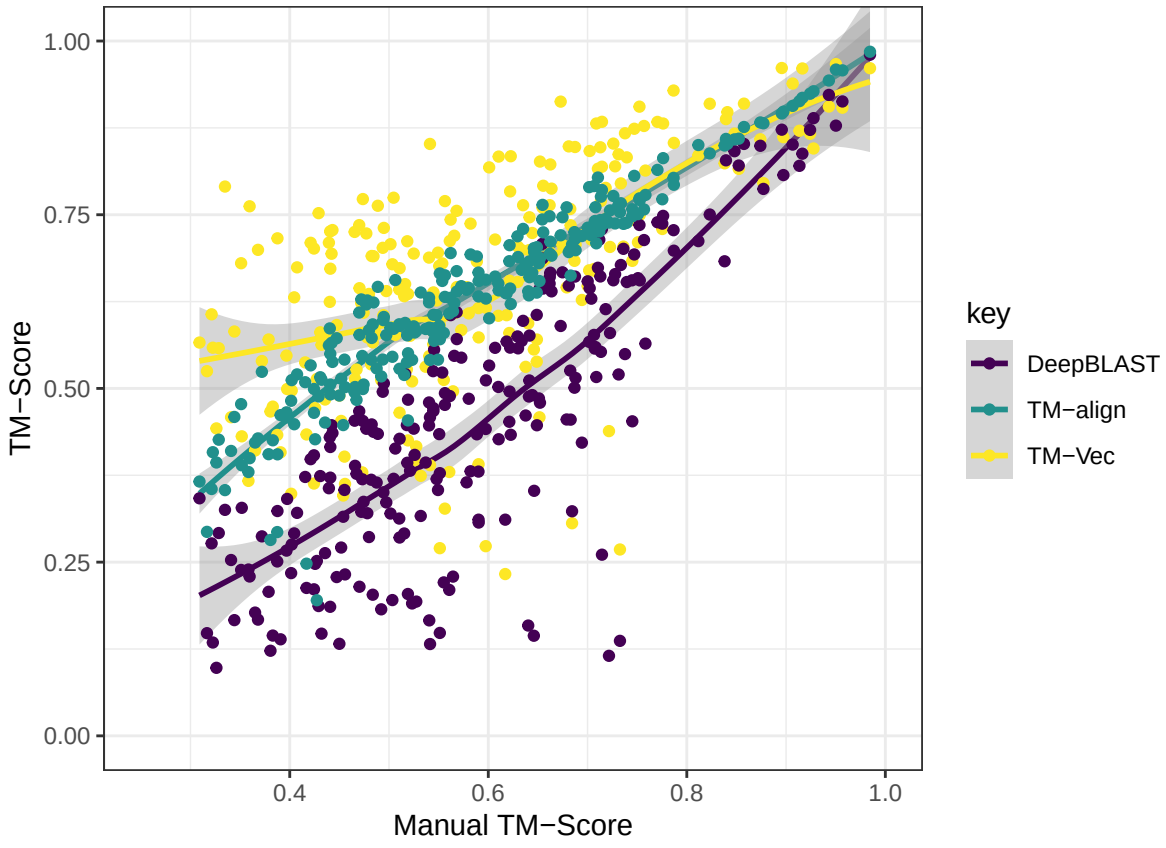


Plot only DeepBLAST and TM-Vec in this plot

```
read_csv("~/deepblast_data_rplots/malidup_sequences_and_tm_scores.csv") %>%
  select(Name, 'TM-Vec' = tm_score_predictions) %>%
  left_join(
    read_csv("~/deepblast_data_rplots/tm_scores_deepblast.csv"), by = c("Name" = "index")
  ) %>%
  select(Name, 'TM-Vec', 'DeepBLAST' = deepblast, 'TM-align' = tm, manual) %>%
  gather(-Name, -manual, key = "key", value = "value") %>%
  filter(!is.na(manual)) %>%
  ggplot(aes(manual, value, color = key)) +
  geom_smooth() +
  geom_point() +
  theme_bw() +
  coord_fixed(ratio = .7, xlim = c(0.25, 1), ylim = c(0,1)) +
  labs(x = "Manual TM-Score", y = "TM-Score", color = "key") +
  scale_color_viridis_d(direction = 1)
```

```
## Rows: 241 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): Name, Sequence 1, Sequence 2, Protein_pair
## dbl (7): Tm_score1, TM_score2, tm_max, tm_score_predictions, diff, diff_cath...
##
## i Use `spec()` to retrieve the full column specification for this data.
```

```
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 234 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): index
## dbl (7): manual, dali, fast, tm, mammoth, needleman-wunsch, deepblast
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Correlation matrix

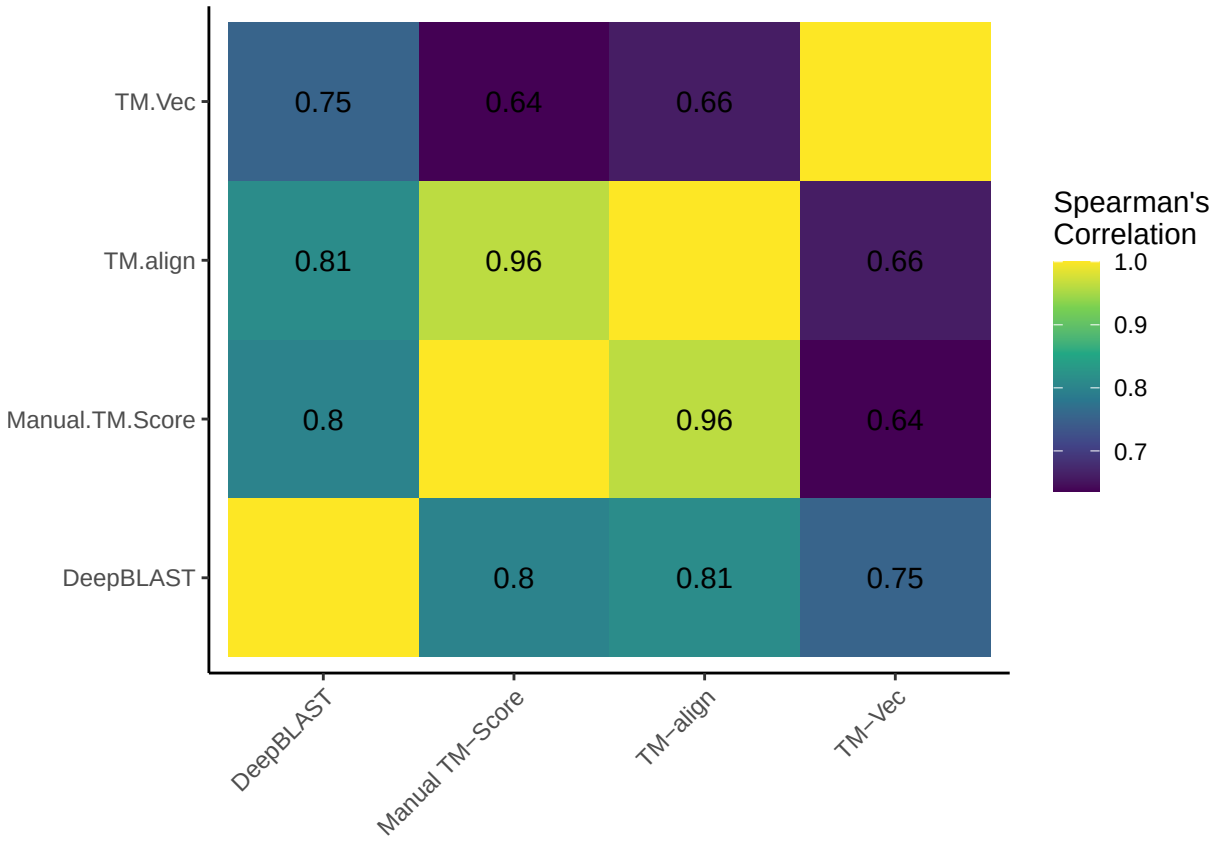
```
tm_prediction_matrix <-
  read_csv("~/malidup_sequences_and_tm_scores.csv") %>%
  select(Name, Predicted_TM = tm_score_predictions) %>%
  left_join(
    read_csv("~/Downloads/tm_scores_deepblast.csv"), by = c("Name" = "index")
  ) %>%
  select(-Name)
```

```
## Rows: 241 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): Name, Sequence 1, Sequence 2, Protein_pair
## dbl (7): Tm_score1, TM_score2, tm_max, tm_score_predictions, diff, diff_cath...
##
## i Use `spec()` to retrieve the full column specification for this data.
```

```
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 234 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): index
## dbl (7): manual, dali, fast, tm, mammoth, needleman-wunsch, deepblast
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
cors <- function(df) {
   M <- Hmisc::rcorr(as.matrix(df), type="spearman")
   Mdf <- map(M, ~data.frame(.x))
  }


formatted_cors <- function(df){
 cors(df) %>%
 map(~rownames_to_column(.x, var="measure1")) %>%
 map(~pivot_longer(.x, -measure1, "measure2")) %>%
 bind_rows(.id = "id") %>%
 pivot_wider(names_from = id, values_from = value) %>%
 mutate(sig_p = ifelse(P < .05, T, F), p_if_sig = ifelse(P <.05, P, NA), r_if_sig = ifelse(P <.05, r, NA
}

tm_prediction_matrix %>%
  select(`TM-Vec`=Predicted_TM, `Manual TM-Score`=manual, `TM-align`=tm, `DeepBLAST`=deepblast) %>%
  formatted_cors() %>%
   ggplot(aes(measure1, measure2, fill=r, label=round(r_if_sig,2))) +
   geom_tile() +
   labs(
     x = NULL, y = NULL,
     fill = "Spearman's\nCorrelation"
   ) +
  scale_fill_viridis_c(direction = 1) +
  geom_text() +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Removed 4 rows containing missing values (`geom_text()`).
```



CATH validation: Combined CATH plot

```
combined_cath_tests <-
  tm_preds_domains_left_out_cath %>%
  mutate(test_dataset = "Domains") %>%
  rbind(
    tm_preds_pairs_left_out_cath %>%
    mutate(test_dataset = "Pairs")
  ) %>%
  rbind(
    tm_preds_out_of_fold_cath %>%
    select(Predicted_TM_score, Ground_truth_TM_score) %>%
    mutate(test_dataset = "Folds")
  ) %>%
  mutate(test_dataset = factor(test_dataset, levels=c("Pairs", "Domains", "Folds")))

combined_cath_tests_validation <-
  combined_cath_tests %>%
  mutate(
    Ground_truth_TM_score_breaks = cut(Ground_truth_TM_score, breaks = seq(from = 0, to = 1, by = .25),
    filling = "filling",
    prediction_error = abs(Predicted_TM_score - Ground_truth_TM_score)
  ) %>%
  ggplot(aes(Ground_truth_TM_score_breaks, prediction_error, fill = test_dataset)) +
  geom_boxplot(outlier.shape = NA) +
  coord_cartesian(ylim =  c(0, .3)) +
  labs(x = "TM-Score", y = "Absolute Value of TM-Score Prediction Error", fill = "Test Dataset", title =
  theme_bw() +
```

```
  scale_fill_viridis_d(begin=.25)
```

CATH validation: Median error for held outs Pairs, Domains, and Folds

```
combined_cath_tests %>%
  mutate(
    Ground_truth_TM_score_breaks = cut(Ground_truth_TM_score, breaks = seq(from = 0, to = 1, by = .25),
    filling = "filling",
    prediction_error = abs(Predicted_TM_score - Ground_truth_TM_score)
  ) %>%
  group_by(test_dataset) %>%
  summarise(median_for_dataset = median(prediction_error)) %>%
  ungroup()
```

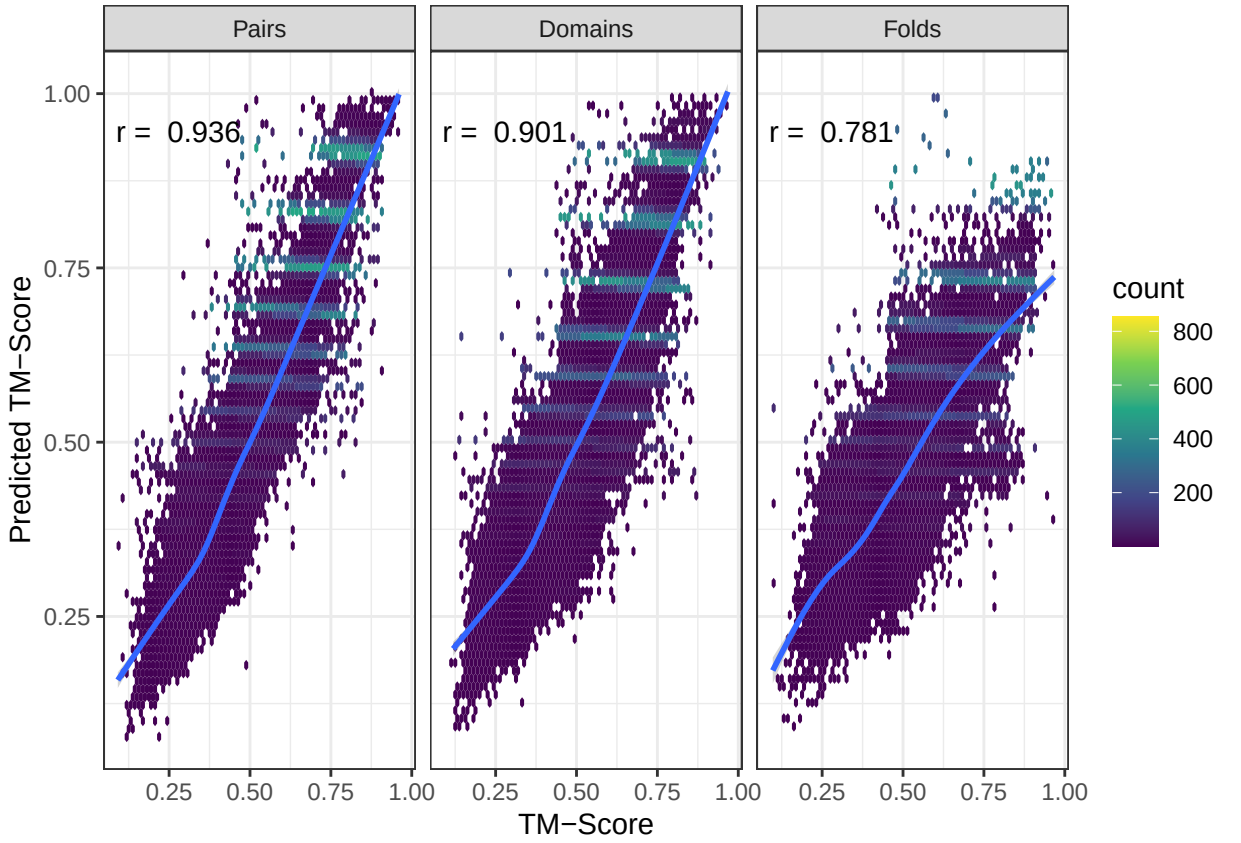
```
## # A tibble: 3 x 2
##   test_dataset median_for_dataset
##   <fct>                     <dbl>
## 1 Pairs                    0.0232
## 2 Domains                  0.0291
## 3 Folds                    0.0427
```

Plot CATH correlation and scatterplot of TM-scores versus Predicted TM-scores

```
combined_cath_correlations <-
  combined_cath_tests %>%
  group_by(test_dataset) %>%
  summarise(corr = cor(Ground_truth_TM_score, Predicted_TM_score)) %>%
  ungroup() %>%
  mutate(corr = paste("r = ", round(corr, digits = 3)))


combined_figs <-
  combined_cath_tests %>%
  mutate(
    Ground_truth_TM_score_breaks = cut(Ground_truth_TM_score, breaks = seq(from = 0, to = 1, by = .25),
    filling = "filling",
    prediction_error = abs(Predicted_TM_score - Ground_truth_TM_score)
  ) %>%
  group_by(test_dataset) %>%
  sample_n(75000) %>%
  ungroup() %>%
  ggplot(aes(Ground_truth_TM_score, Predicted_TM_score)) +
  geom_hex(bins = 70) +
  scale_fill_viridis_c() +
  geom_smooth() +
  facet_grid(cols = vars(test_dataset)) +
  geom_text(
    data = combined_cath_correlations,
    mapping = aes(x = -Inf, y = .90, label = corr),
    hjust   = -0.1,
    vjust   = -1
  ) +
  labs(
    x = "TM-Score",
    y = "Predicted TM-Score"
```

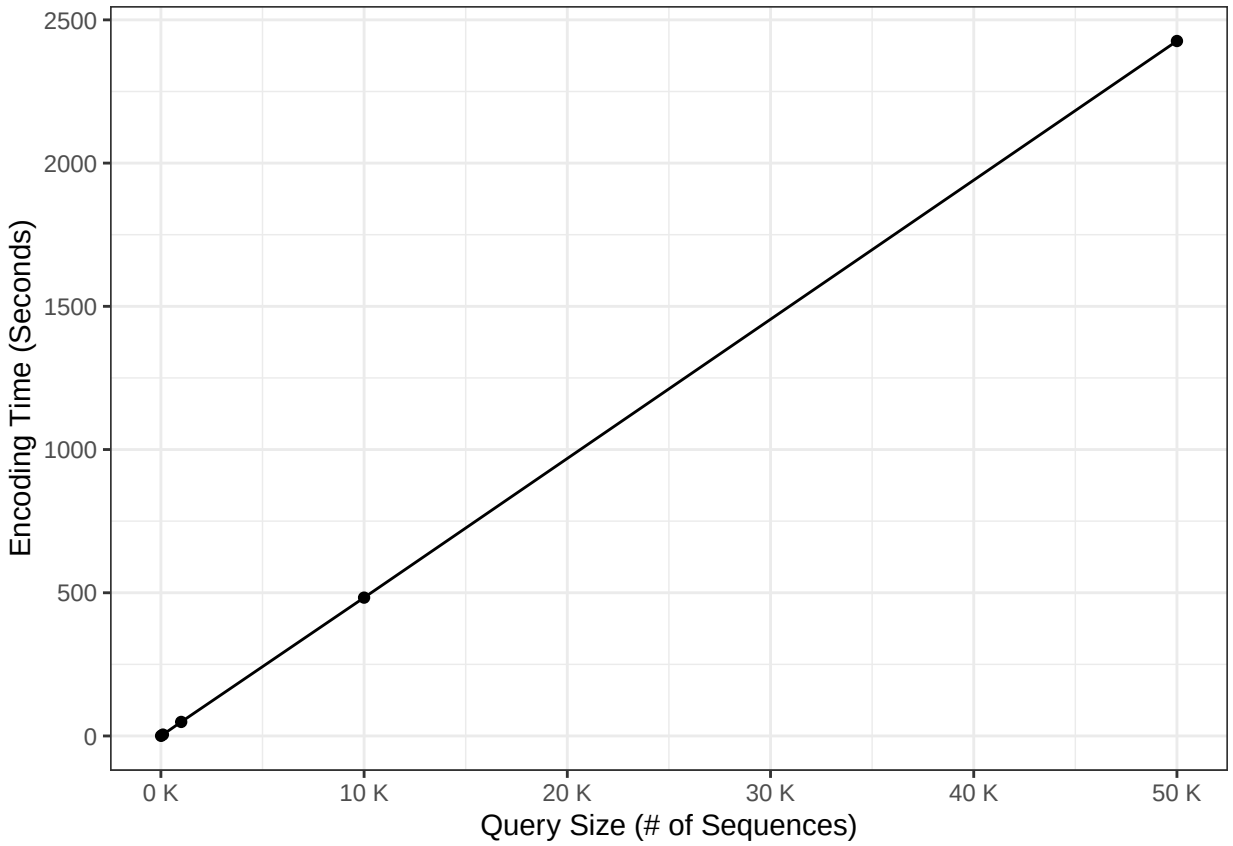
```
  ) +
  theme_bw()

combined_figs
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



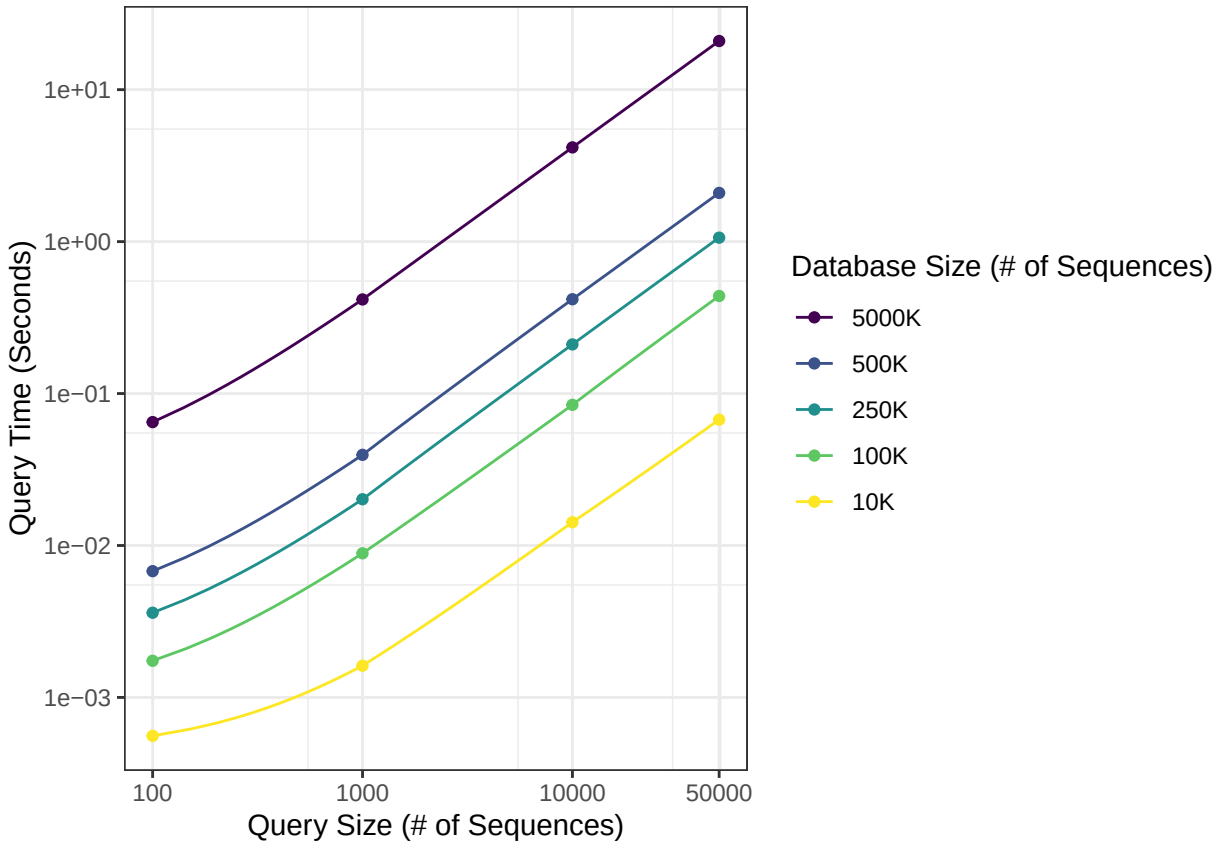
Visualize performance of TM-Vec:Encoding

```
deep_blast_speed_encoding %>%
  ggplot(aes(`Query Size`, `Encoding Time`)) +
  geom_point() +
  geom_line() +
  scale_x_continuous(labels = scales::unit_format(unit = "K", scale = 1e-3)) +
  labs(x = "Query Size (# of Sequences)", y = "Encoding Time (Seconds)") +
  theme_bw()
```



Visualize performance of TM-Vec: query time for different database sizes

```
deep_blast_speed_query_database <-
  deep_blast_speed_query_database %>%
  mutate(`Query Size` = as.integer(`Query Size`))

deep_blast_speed_query_database %>%
  filter(`Query Size` != 10) %>%
  mutate(
    `Database Size K` = paste(`Database Size`/1e3, sep="", "K"),
    `Database Size K` = fct_reorder(`Database Size K`, -`Database Size`)
  ) %>%
  ggplot(aes(`Query Size`, `Time`, colour = `Database Size K`)) +
  geom_point() +
  geom_line() +
  coord_trans(y='log10', x = 'log10') +
  scale_y_continuous(breaks = c(.001, .01, .1, 1, 10)) +
  scale_x_continuous(breaks = unique(deep_blast_speed_query_database$`Query Size`) %>% as.integer()) +
  scale_colour_viridis(discrete = TRUE) +
  labs(x = "Query Size (# of Sequences)", y = "Query Time (Seconds)", color= "Database Size (# of Sequer
  theme_bw()
```



CATH TSNE plots of TM-Vec embedding vectors

```
tsne_data_all <-
  tsne_data %>%
  as_tibble() %>%
  cbind(meta_data_tsne)

tsne_data_for_class <-
  tsne_data_all %>%
  filter(CLASS != "Few Secondary Structures")

tsne_data_matrix_class <- as.matrix(tsne_data_for_class %>% select('0':'511'))
tsne_data_rtsne_class <- Rtsne(tsne_data_matrix_class, dims = 2)

#TSNE TM-Vec:  ARCH
top_archs <- meta_data_tsne %>% count(ARCH, sort = T) %>% slice(1:5) %>% pull(ARCH)

tsne_data_for_arch <-
  tsne_data_all %>%
  filter(ARCH %in% top_archs)

tsne_data_matrix_arch <- as.matrix(tsne_data_for_arch %>% select('0':'511'))
tsne_data_rtsne_arch <- Rtsne(tsne_data_matrix_arch, dims = 2)

#TSNE TM-Vec: TOPOL
top_topols <- meta_data_tsne %>% count(TOPOL, sort = T) %>% slice(1:5) %>% pull(TOPOL)
```

```
tsne_data_for_topol <-
  tsne_data_all %>%
  filter(TOPOL %in% top_topols)

tsne_data_matrix_topol <- as.matrix(tsne_data_for_topol %>% select('0':'511'))
tsne_data_rtsne_topol <- Rtsne(tsne_data_matrix_topol, dims = 2)

#TSNE TM-Vec: HOMOL
top_homols <- meta_data_tsne %>% count(HOMOL, sort = T) %>% slice(1:5) %>% pull(HOMOL)

tsne_data_for_homol <-
  tsne_data_all %>%
  filter(HOMOL %in% top_homols)

tsne_data_matrix_homol <- as.matrix(tsne_data_for_homol %>% select('0':'511'))
tsne_data_rtsne_homol <- Rtsne(tsne_data_matrix_homol, dims = 2)
```

CATH TSNE plots of ProtTrans embedding vectors

```
protrans_meta_data <-
  protrans_names %>%
  select(DOMAIN = `0`) %>%
  left_join(meta_data_tsne, by = "DOMAIN")

tsne_data_all_protrans <-
  protrans_vecs %>%
  as_tibble() %>%
  cbind(protrans_meta_data)

#TSNE CLASS: ProtTrans
tsne_data_protrans_for_class <-
  tsne_data_all_protrans %>%
  filter(CLASS != "Few Secondary Structures")

tsne_data_protrans_matrix_class <- as.matrix(tsne_data_protrans_for_class %>% select('0':'1023'))
tsne_data_protrans_rtsne_class <- Rtsne(tsne_data_protrans_matrix_class, dims = 2)

#TSNE ProtTrans: ARCH
tsne_data_protrans_for_arch <-
  tsne_data_all_protrans %>%
  filter(ARCH %in% top_archs)

tsne_data_protrans_matrix_arch <- as.matrix(tsne_data_protrans_for_arch %>% select('0':'1023'))
tsne_data_protrans_rtsne_arch <- Rtsne(tsne_data_protrans_matrix_arch, dims = 2)

#TSNE ProtTrans: TOPOLOGY
tsne_data_protrans_for_topol <-
  tsne_data_all_protrans %>%
  filter(TOPOL %in% top_topols)

tsne_data_protrans_matrix_topol <- as.matrix(tsne_data_protrans_for_topol %>% select('0':'1023'))
tsne_data_protrans_rtsne_topol <- Rtsne(tsne_data_protrans_matrix_topol, dims = 2)

#TSNE ProtTrans: Homology
```

```
tsne_data_protrans_for_homol <-
  tsne_data_all_protrans %>%
  filter(HOMOL %in% top_homols)

tsne_data_protrans_matrix_homol <- as.matrix(tsne_data_protrans_for_homol %>% select('0':'1023'))
tsne_data_protrans_rtsne_homol <- Rtsne(tsne_data_protrans_matrix_homol, dims = 2)
```

Make figures (ProtTrans and TM-Vec TSNE side by side)

```
#Homology figure combined
fig_homology <-
  tsne_data_protrans_rtsne_homol$Y %>%
  as_tibble() %>%
  cbind(tsne_data_protrans_for_homol %>% select(-('0':'1023'))) %>%
  mutate(method = "ProtTrans") %>%
  rbind(
    tsne_data_rtsne_homol$Y %>%
    as_tibble() %>%
    cbind(tsne_data_for_homol %>% select(-('0':'511'))) %>%
    mutate(method = "TM-Vec")
  ) %>%
  ggplot(aes(V1, V2, color = HOMOL)) +
  geom_point() +
  labs(color = "Homology", x="Dimension 1", y="Dimension 2") +
  theme_bw() +
  theme(
    axis.text.x=element_blank(),
    axis.ticks.x=element_blank(),
    axis.text.y=element_blank(),
    axis.ticks.y=element_blank(),
    #legend.key.width = unit(1, 'cm')
  ) +
  scale_colour_viridis(discrete = TRUE, labels = function(x) str_wrap(x, width = 20)) +
  facet_grid(cols = vars(method))
```

```
## Warning: The `x` argument of `as_tibble.matrix()` must have unique column names if
## `.name_repair` is omitted as of tibble 2.0.0.
## i Using compatibility `.name_repair`.
```

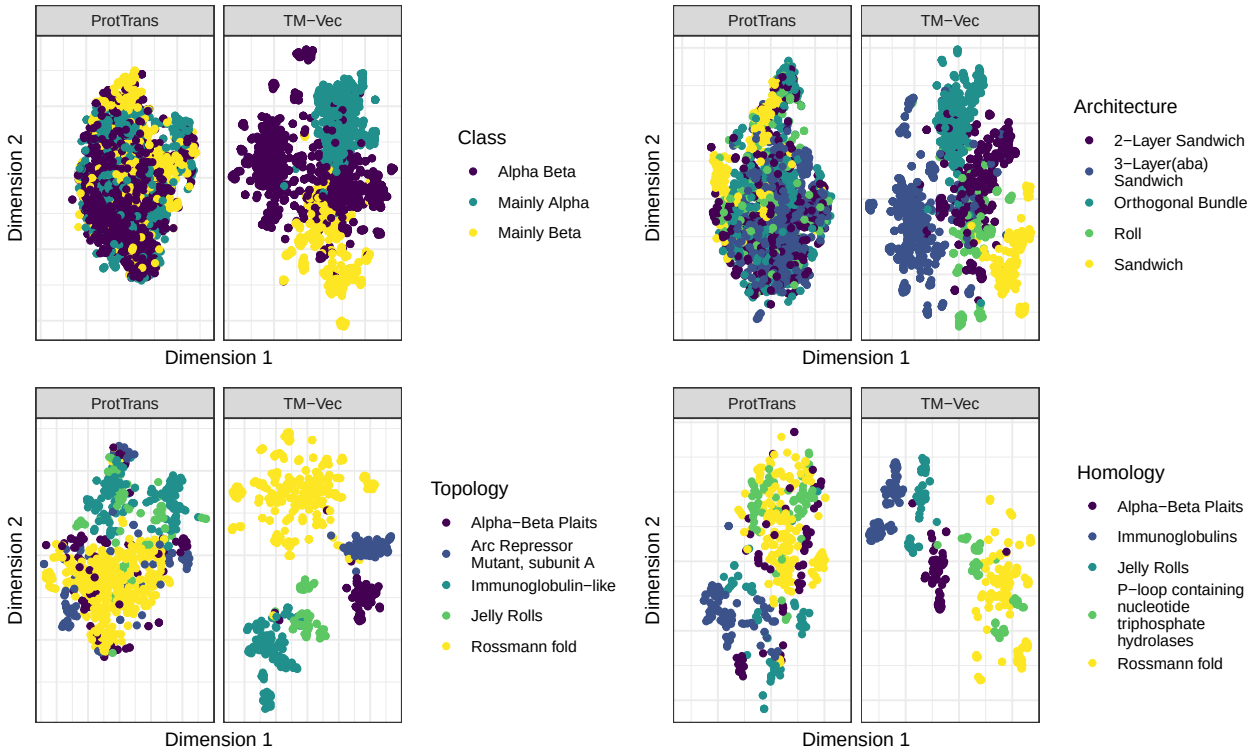
```
#Topology figure combined
fig_topology <-
  tsne_data_protrans_rtsne_topol$Y %>%
  as_tibble() %>%
  cbind(tsne_data_protrans_for_topol %>% select(-('0':'1023'))) %>%
  mutate(method = "ProtTrans") %>%
  rbind(
    tsne_data_rtsne_topol$Y %>%
    as_tibble() %>%
    cbind(tsne_data_for_topol %>% select(-('0':'511'))) %>%
    mutate(method = "TM-Vec")
  ) %>%
  ggplot(aes(V1, V2, color = TOPOL)) +
  geom_point() +
  labs(color = "Topology", x="Dimension 1", y="Dimension 2") +
  theme_bw() +
```

```
  theme(
    axis.text.x=element_blank(),
    axis.ticks.x=element_blank(),
    axis.text.y=element_blank(),
    axis.ticks.y=element_blank(),
    #legend.key.width = unit(1, 'cm')
  ) +
  scale_colour_viridis(discrete = TRUE, labels = function(x) str_wrap(x, width = 20)) +
  facet_grid(cols = vars(method))

#Class figure combined
fig_class <-
  tsne_data_protrans_rtsne_class$Y %>%
  as_tibble() %>%
  cbind(tsne_data_protrans_for_class %>% select(-('0':'1023'))) %>%
  mutate(method = "ProtTrans") %>%
  rbind(
    tsne_data_rtsne_class$Y %>%
    as_tibble() %>%
    cbind(tsne_data_for_class %>% select(-('0':'511'))) %>%
    mutate(method = "TM-Vec")
  ) %>%
  ggplot(aes(V1, V2, color = CLASS)) +
  geom_point() +
  labs(color = "Class", x="Dimension 1", y="Dimension 2") +
  theme_bw() +
  theme(
    axis.text.x=element_blank(),
    axis.ticks.x=element_blank(),
    axis.text.y=element_blank(),
    axis.ticks.y=element_blank(),
    #legend.key.width = unit(1.5, 'cm')
  ) +
  scale_colour_viridis(discrete = TRUE, labels = function(x) str_wrap(x, width = 30)) +
  facet_grid(cols = vars(method))


#Architecture figure combined
fig_arch <-
  tsne_data_protrans_rtsne_arch$Y %>%
  as_tibble() %>%
  cbind(tsne_data_protrans_for_arch %>% select(-('0':'1023'))) %>%
  mutate(method = "ProtTrans") %>%
  rbind(
    tsne_data_rtsne_arch$Y %>%
    as_tibble() %>%
    cbind(tsne_data_for_arch %>% select(-('0':'511'))) %>%
    mutate(method = "TM-Vec")
  ) %>%
  ggplot(aes(V1, V2, color = ARCH)) +
  geom_point() +
  labs(color = "Architecture", x="Dimension 1", y="Dimension 2") +
  theme_bw() +
```

```
  theme(
    axis.text.x=element_blank(),
    axis.ticks.x=element_blank(),
    axis.text.y=element_blank(),
    axis.ticks.y=element_blank(),
    #legend.key.width = unit(1.5, 'cm')
  ) +
  scale_colour_viridis(discrete = TRUE, labels = function(x) str_wrap(x, width = 20)) +
  facet_grid(cols = vars(method))
```

Plot everything combined

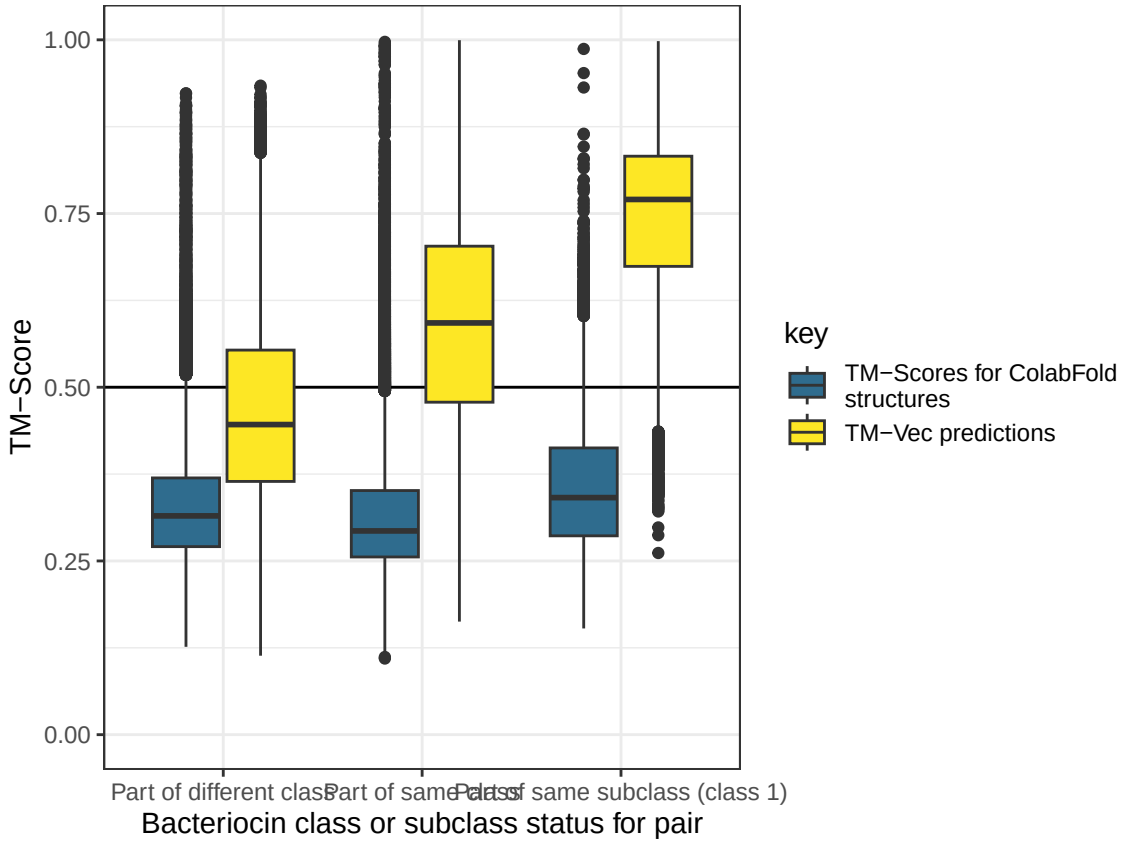
```
plot_grid(fig_class, fig_arch, fig_topology, fig_homology, ncol = 2, align = "v")
```



Colabfold benchmarking

```
colab_fig <-
  colabfold_benchmarking  %>%
  mutate(chain_comparison = if_else(class_x == class_y, "Part of same class", "Part of different class"
  rbind(
    colabfold_benchmarking  %>%
    filter(class_x == "Class_1", class_y == "Class_1") %>%
    mutate(chain_comparison = if_else(Subclass_x == Subclass_y, "Part of same subclass (class 1)", "Par
    filter(chain_comparison == "Part of same subclass (class 1)")
  ) %>%
  select(chain_comparison, tm_max, deepblast_predictions) %>%
  gather(-chain_comparison, key=key, value=value) %>%
  mutate(key = if_else(key == "tm_max", "TM-Scores for ColabFold structures", "TM-Vec predictions")) %>%
  mutate(key = as_factor(key), chain_comparison = as_factor(chain_comparison)) %>%
  ggplot(aes(chain_comparison,  value, fill=key)) +
  geom_hline(yintercept = .5, ) +
```

```
  geom_boxplot() +
  theme_bw() +
  coord_cartesian(ylim = c(0,1)) +
  labs(x = "Bacteriocin class or subclass status for pair", y = "TM-Score") +
  scale_fill_viridis_d(begin=.35, labels = function(x) str_wrap(x, width = 25))

colab_fig
```



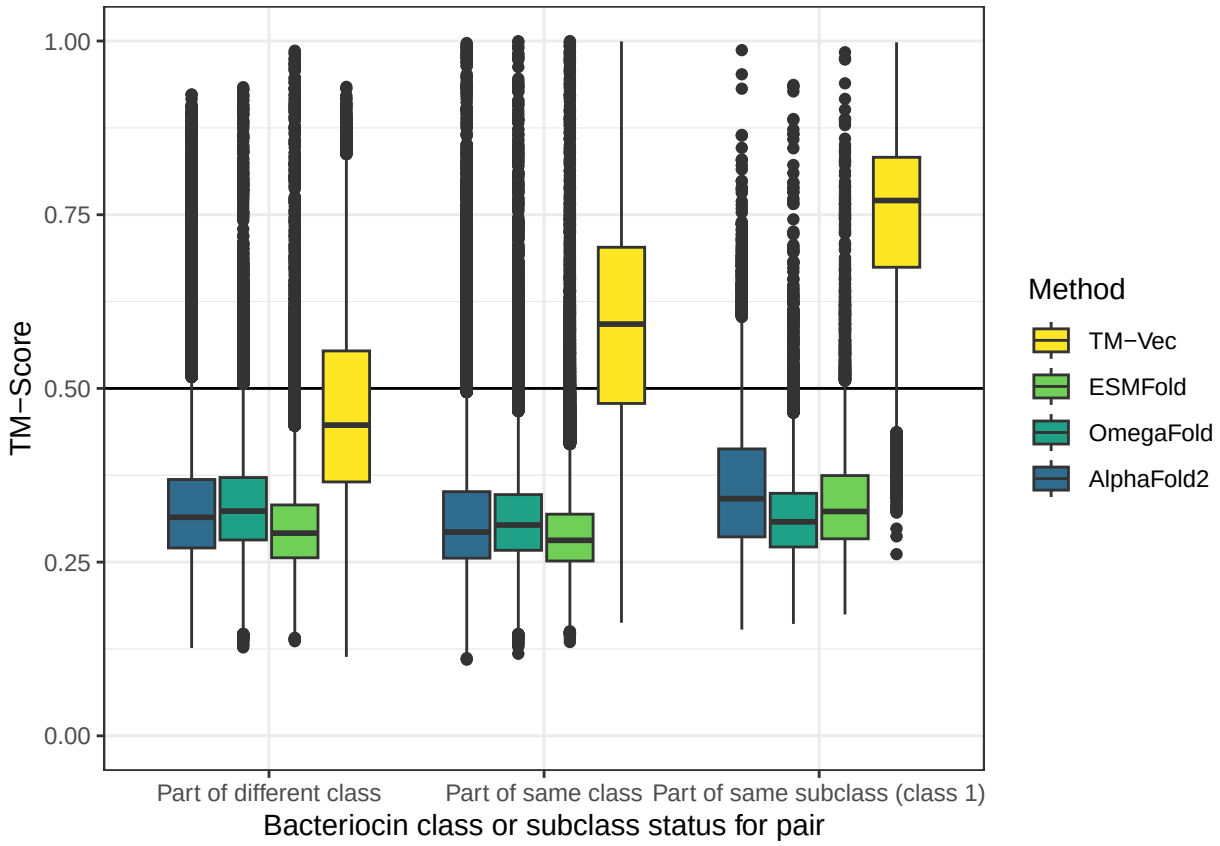

```
colabfold_benchmarking <-
  read_csv("~/tm_scores_with_predictions_and_meta_colabfold_cath_vs_swiss.csv") %>%
  left_join(omegafold_benchmarking, by=c("chain_1", "chain_2")) %>%
  left_join(esm_benchmarking, by=c("chain_1", "chain_2")) %>%
  filter(!is.na(omega_fold_tm_max))
```

```
## Rows: 238395 Columns: 19
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (6): class_x, Subclass_x, class_y, Subclass_y, Seq_x, Seq_y
## dbl (13): chain_1, chain_2, tm_1, tm_2, tm_max, deepblast_predictions, ID_x,...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
structure_prediction_benchmarking <-
  colabfold_benchmarking  %>%
  mutate(chain_comparison = if_else(class_x == class_y, "Part of same class", "Part of different class")
  rbind(
```

```
    colabfold_benchmarking  %>%
    filter(class_x == "Class_1", class_y == "Class_1") %>%
    mutate(chain_comparison = if_else(Subclass_x == Subclass_y, "Part of same subclass (class 1)", "Par
    filter(chain_comparison == "Part of same subclass (class 1)")
  ) %>%
  select(chain_comparison, tm_max, omega_fold_tm_max, esm_tm_max, deepblast_predictions, cath_2_layer, 
  gather(-chain_comparison, key=key, value=value) %>%
  mutate(key = if_else(key == "tm_max", "AlphaFold2", key)) %>%
  mutate(key = if_else(key == "esm_tm_max", "ESMFold", key)) %>%
  mutate(key = if_else(key == "omega_fold_tm_max", "OmegaFold", key)) %>%
  mutate(key = if_else(key == "deepblast_predictions", "TM-Vec", key)) %>%
  mutate(key = as_factor(key), chain_comparison = as_factor(chain_comparison)) %>%
  filter(!key %in% c("cath_2_layer", "swiss_4_layer_600", "cath_4_layer"))


structure_prediction_benchmarking_fig <-
  structure_prediction_benchmarking %>%
  ggplot(aes(chain_comparison,  value, fill=key)) +
  geom_hline(yintercept = .5, ) +
  geom_boxplot() +
  theme_bw() +
  coord_cartesian(ylim = c(0,1)) +
  labs(x = "Bacteriocin class or subclass status for pair", y = "TM-Score") +
  guides(fill=guide_legend(title="Method", reverse = TRUE)) +
  scale_fill_viridis_d(begin=.35, labels = function(x) str_wrap(x, width = 35))
```

```
structure_prediction_benchmarking_fig
```


1.00
0.75
0.50
0.25
0.00
TM−Score
Part of different class
Part of same class
Part of same subclass (class 1)
Bacteriocin class or subclass status for pair
Method
TM−Vec
ESMFold
OmegaFold
AlphaFold2