TCREMP and TCRpMHC structures

M.S.

2025-01-18

Fetch list of TCRs and their structures from VDJdb

```
download.file('https://raw.githubusercontent.com/antigenomics/vdjdb-db/refs/heads/master/chunks/PDB_Dat
              destfile = "v_tcrpmhc_raw.txt",
              method = "wget")

vpdb <- read_tsv("v_tcrpmhc_raw.txt") |>
  mutate(meta.structure.id = ifelse(meta.structure.id == '4E+41',
                                    "4e41",
                                    meta.structure.id))
```

```
## New names:
## Rows: 284 Columns: 33
## -- Column specification
## -------------------------------------------------------- Delimiter: "\t" chr
## (23): cdr3.alpha, v.alpha, j.alpha, cdr3.beta, v.beta, d.beta, j.beta, s... dbl
## (1): ...1 lgl (9): method.frequency, meta.study.id, meta.cell.subset,
## meta.subset.fre...
## 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.
## * ‘‘ -> ‘...1‘
```

```
vpdb$`...1` <- NULL

vpdb_meta <- vpdb |>
  select(mhc.class, clone_id = meta.structure.id) |>
  unique()

glimpse(vpdb)
```

```
## Rows: 284
## Columns: 32
## $ cdr3.alpha            <chr> "CAVTTDSWGKLQF", "CAAMEGAQKLVF", "CAATGSFNKLTF",~
## $ v.alpha               <chr> "TRAV12-2*01", "TRAV29DV5*01", "TRAV14D-2*01", "~
## $ j.alpha               <chr> "TRAJ24*01", "TRAJ54*01", "TRAJ4*01", "TRAJ32*01~
## $ cdr3.beta             <chr> "CASRPGLAGGRPEQYF", "CASSYPGGGFYEQYF", "CASGGQGR~
## $ v.beta                <chr> "TRBV6-5*01", "TRBV6-5*01", "TRBV13-2*01", "TRBV~
## $ d.beta                <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ j.beta                <chr> "TRBJ2-7*01", "TRBJ2-7*01", "TRBJ2-1*01", "TRBJ1~
## $ species               <chr> "HomoSapiens", "HomoSapiens", "MusMusculus", "Mu~
## $ mhc.a                 <chr> "HLA-A*02:01:59", "HLA-A*02:01:48", "H-2Aa", "H-~
## $ mhc.b                 <chr> "B2M", "B2M", "H-2Aa", "B2M", "HLA-DRB1*01:01:01~
## $ mhc.class             <chr> "MHCI", "MHCI", "MHCII", "MHCI", "MHCII", "MHCI"~
## $ antigen.epitope       <chr> "LLFGYPVYV", "LLFGYPVYV", "GNSHRGAIEWEGIESG", "I~
## $ antigen.gene          <chr> "Tax", "Tax", "Ovotransferrin", "Kctd20", "HA", ~
## $ antigen.species       <chr> "HTLV-1", "HTLV-1", "GallusGallus", "MusMusculus~
## $ reference.id          <chr> "PMID:8906788", "PMID:9586631", "PMID:10583947",~
## $ method.identification <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ method.frequency      <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ method.singlecell     <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ method.sequencing     <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ method.verification   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ meta.study.id         <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ meta.cell.subset      <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ meta.subset.frequency <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ meta.subject.cohort   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ meta.subject.id       <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ meta.replica.id       <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ meta.clone.id         <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ meta.epitope.id       <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ meta.tissue           <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ meta.donor.MHC        <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ meta.donor.MHC.method <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ meta.structure.id     <chr> "1ao7", "1bd2", "1d9k", "1fo0", "1fyt", "1g6r", ~
```

Format into AIRR

```
vpdb <- vpdb |>
  filter(species == "HomoSapiens") |>
  pivot_longer(cols = c(v.alpha, j.alpha, cdr3.alpha,
                        v.beta, j.beta, cdr3.beta)) |>
  select(meta.structure.id, name, value) |>
  separate(name, c("variable", "locus")) |>
  unique() |>
  pivot_wider(id_cols = c(meta.structure.id, locus),
              names_from = variable,
              values_from = value) |>
  rename(clone_id = meta.structure.id,
         v_call = v, j_call = j,
         junction_aa = cdr3)

write_tsv(vpdb, "v_tcrpmhc.txt")
glimpse(vpdb)
```

```
## Rows: 432
## Columns: 5
## $ clone_id    <chr> "1ao7", "1ao7", "1bd2", "1bd2", "1fyt", "1fyt", "1j8h", "1~
## $ locus       <chr> "alpha", "beta", "alpha", "beta", "alpha", "beta", "alpha"~
## $ v_call      <chr> "TRAV12-2*01", "TRBV6-5*01", "TRAV29DV5*01", "TRBV6-5*01",~
## $ j_call      <chr> "TRAJ24*01", "TRBJ2-7*01", "TRAJ54*01", "TRBJ2-7*01", "TRA~
## $ junction_aa <chr> "CAVTTDSWGKLQF", "CASRPGLAGGRPEQYF", "CAAMEGAQKLVF", "CASS~
```

Download PDB structures

```r
pdb_ids <- unique(vpdb$clone_id)
pdb_files <- get.pdb(pdb_ids, path = "tmp/")
pdb_files <- pdb_files[endsWith(pdb_files, ".pdb")]
```

Align, superimpose and get coords. Then compute RMSD

```r
struct_alns <- pdbaln(pdb_files, fit = F,
                      ncore = 12,
                      exefile = "msa",
                      maxiters = 256)
# how coords are organized:
# (struct_alns$xyz |> matrix(nrow = 4))[1,] |> matrix(nrow = 3) |> t()
struct_alns_ids <- tools::file_path_sans_ext(basename(struct_alns$id))
struct_alns_coords <- struct_alns$xyz |>
  matrix(nrow = length(struct_alns$id)) |>
  t()
aligned_residues <- apply(struct_alns_coords, 2, \(x) sum(!is.na(x)))
struct_alns_coords[is.na(struct_alns_coords)] <- 0
# pairwise distances
struct_alns_eucl <- struct_alns_coords |>
  t() |>
  dist() |>
  as.matrix()
# normalize for number of aligned residues
struct_alns_rmsd <- t(struct_alns_eucl / sqrt(aligned_residues)) / sqrt(aligned_residues)
rownames(struct_alns_rmsd) <- struct_alns_ids
colnames(struct_alns_rmsd) <- struct_alns_ids
struct_alns_rmsd <- struct_alns_rmsd |>
  melt()
colnames(struct_alns_rmsd) <- c("clone_id.from", "clone_id.to", "rmsd")
summary(struct_alns_rmsd)
```

Plot RMSD, compare MHCI and MHCII - should be not much difference here due to gaps

```r
struct_alns_rmsd.mhc <- struct_alns_rmsd |>
  filter(clone_id.from != clone_id.to) |>
  left_join(vpdb_meta |>
              rename(mhc.from = mhc.class, clone_id.from = clone_id)) |>
  left_join(vpdb_meta |>
              rename(mhc.to = mhc.class, clone_id.to = clone_id))
```

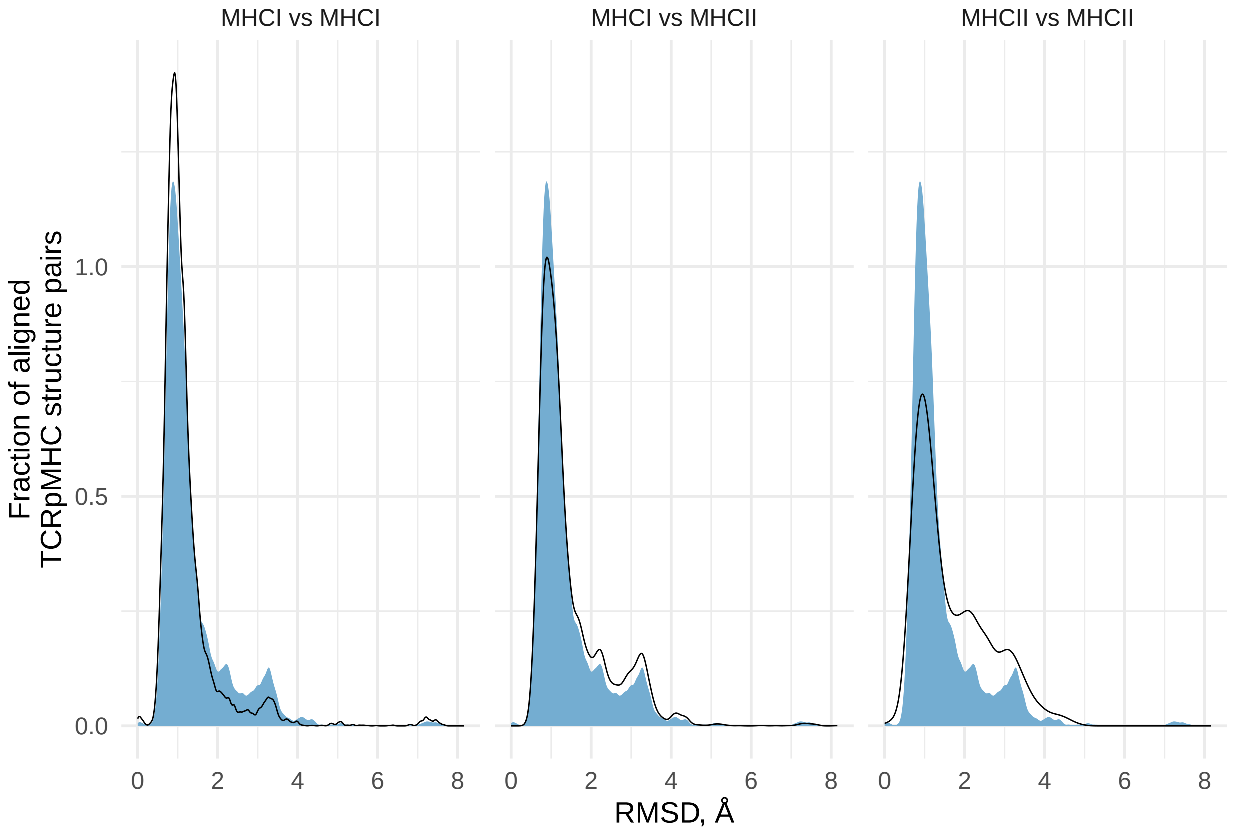
```
## Joining with ‘by = join_by(clone_id.from)‘
## Joining with ‘by = join_by(clone_id.to)‘
```

```r
struct_alns_rmsd.mhc <-
  rbind(struct_alns_rmsd.mhc,
        struct_alns_rmsd.mhc |>
          mutate(mhc.tmp = mhc.from,
                 mhc.from = mhc.to,
                 mhc.to = mhc.tmp) |>
          select(-mhc.tmp)) |>
  unique()
```

```r
plt1 <- ggplot(struct_alns_rmsd.mhc |>
                 mutate(mhc.fromto = paste(mhc.from, "vs", mhc.to)) |>
                 filter(mhc.fromto != "MHCII vs MHCI"),
       aes(x = rmsd)) +
  geom_density(data = struct_alns_rmsd.mhc |>
                 select(rmsd),
               color = NA, fill = "#74add1") +
  geom_density(color = "black", fill = NA, size = 0.25) +
  facet_wrap( ~ mhc.fromto) +
  xlab("RMSD, Å") +
  ylab("Fraction of aligned\nTCRpMHC structure pairs") +
  theme_minimal()
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```r
plt1
```



Distances from TCREMP where prototypes are TCRpMHC structures themselves

```r
tcremp_dists <- read_tsv("v_tcrpmhc_tcremp.tsv")
```

```
## New names:
## Rows: 208 Columns: 1250
## -- Column specification
## -------------------------------------------------------- Delimiter: "\t" chr
## (1): clone_id dbl (1249): ...1, 0_a_v, 0_a_j, 0_a_cdr3, 0_b_v, 0_b_j, 0_b_cdr3,
## 1_a_v, 1_a...
## 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.
## * ‘‘ -> ‘...1‘
```

```r
fetch_dist <- function(dist_type) {
  tcremp_dists |>
    select(ends_with(dist_type)) |>
    as.matrix() ->
    mat
  rownames(mat) <- tcremp_dists$clone_id
  mat |>
    dist() |>
    as.matrix() |>
    melt(value.name = paste0("tcremp_", dist_type)) |>
    rename(clone_id.from = Var1, clone_id.to = Var2)
}

struct_alns_rmsd.mhc.dist <- struct_alns_rmsd.mhc |>
  left_join(fetch_dist("a_v")) |>
  left_join(fetch_dist("a_cdr3")) |>
  left_join(fetch_dist("a_j")) |>
  left_join(fetch_dist("b_v")) |>
  left_join(fetch_dist("b_cdr3")) |>
  left_join(fetch_dist("b_j"))
```

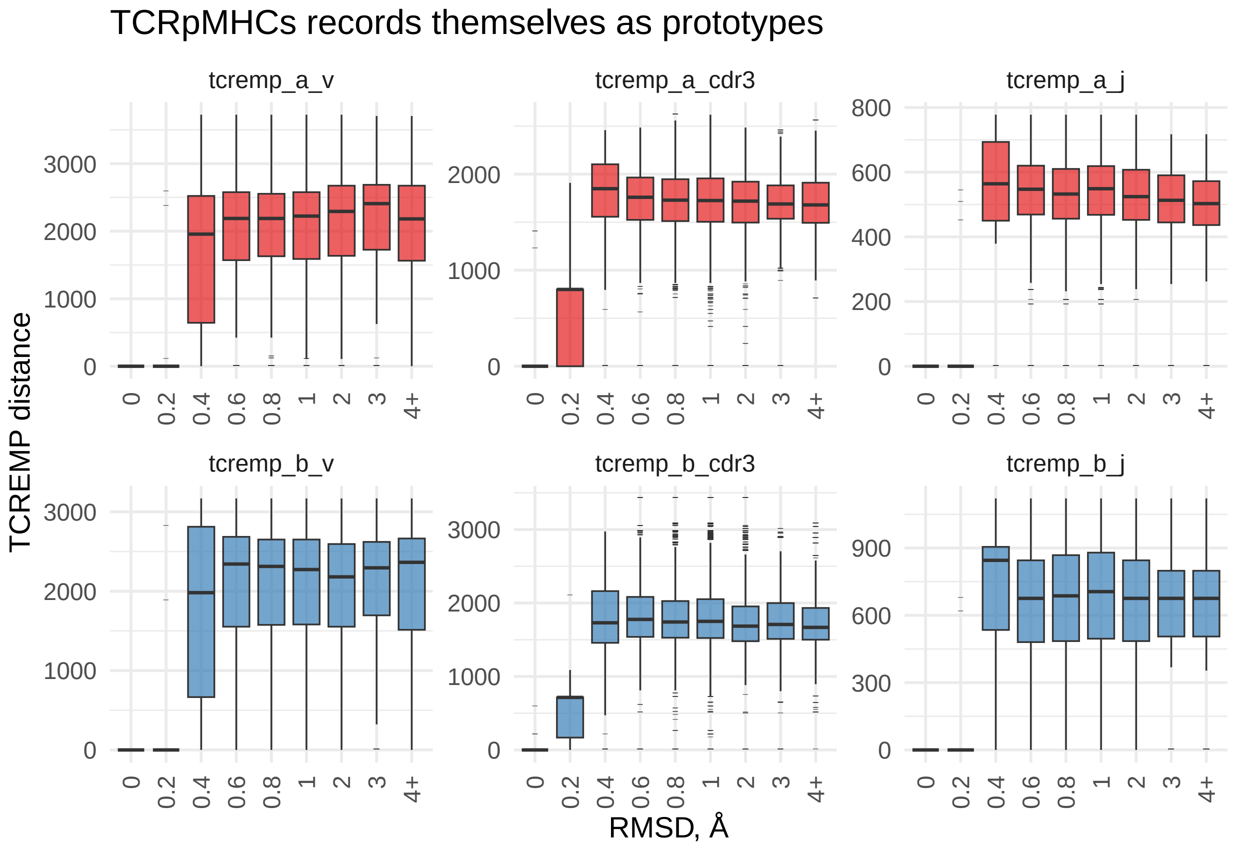
```
## Joining with ‘by = join_by(clone_id.from, clone_id.to)‘
## Joining with ‘by = join_by(clone_id.from, clone_id.to)‘
## Joining with ‘by = join_by(clone_id.from, clone_id.to)‘
## Joining with ‘by = join_by(clone_id.from, clone_id.to)‘
## Joining with ‘by = join_by(clone_id.from, clone_id.to)‘
## Joining with ‘by = join_by(clone_id.from, clone_id.to)‘
```

Plot them

```r
roundfactor <- 5.0
plt2 <- struct_alns_rmsd.mhc.dist |>
  melt(id.vars = 1:5,
       variable.name = "region",
       value.name = "distance") |>
  filter(clone_id.from > clone_id.to) |>
  mutate(chain = substr(region, 8, 8),
         rmsdr = round(rmsd * roundfactor, 0) / roundfactor) |>
  mutate(rmsdr = ifelse(rmsdr > 3, "4+", signif(rmsdr, 1)) |>
           as.factor()) |>
  ggplot(aes(x = rmsdr,
             group = rmsdr,
             y = distance,
             fill = chain)) +
  geom_boxplot(alpha = 0.7, size = 0.3, outlier.size = 1.0,
               outlier.shape = "-") +
  xlab("RMSD, Å") +
  ylab("TCREMP distance") +
  facet_wrap(. ~ region, scales = "free") +
  scale_fill_brewer(guide = F, palette = "Set1") +
  theme_minimal() +
  ggtitle("TCRpMHCs records themselves as prototypes") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
plt2
```

```
## Warning: Removed 13206 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

```
## Warning: The ‘guide‘ argument in ‘scale_*()‘ cannot be ‘FALSE‘. This was deprecated in
## ggplot2 3.3.4.
## i Please use "none" instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



Distances from TCREMP where prototypes are sampled from OLGA

```
tcremp_dists <- read_tsv("v_tcrpmhc_against_olga_tcremp.tsv.gz")
```

```
## Rows: 208 Columns: 18001
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr     (1): clone_id
## dbl (18000): 0_a_v, 0_a_j, 0_a_cdr3, 0_b_v, 0_b_j, 0_b_cdr3, 1_a_v, 1_a_j, 1...
##
## 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.
```

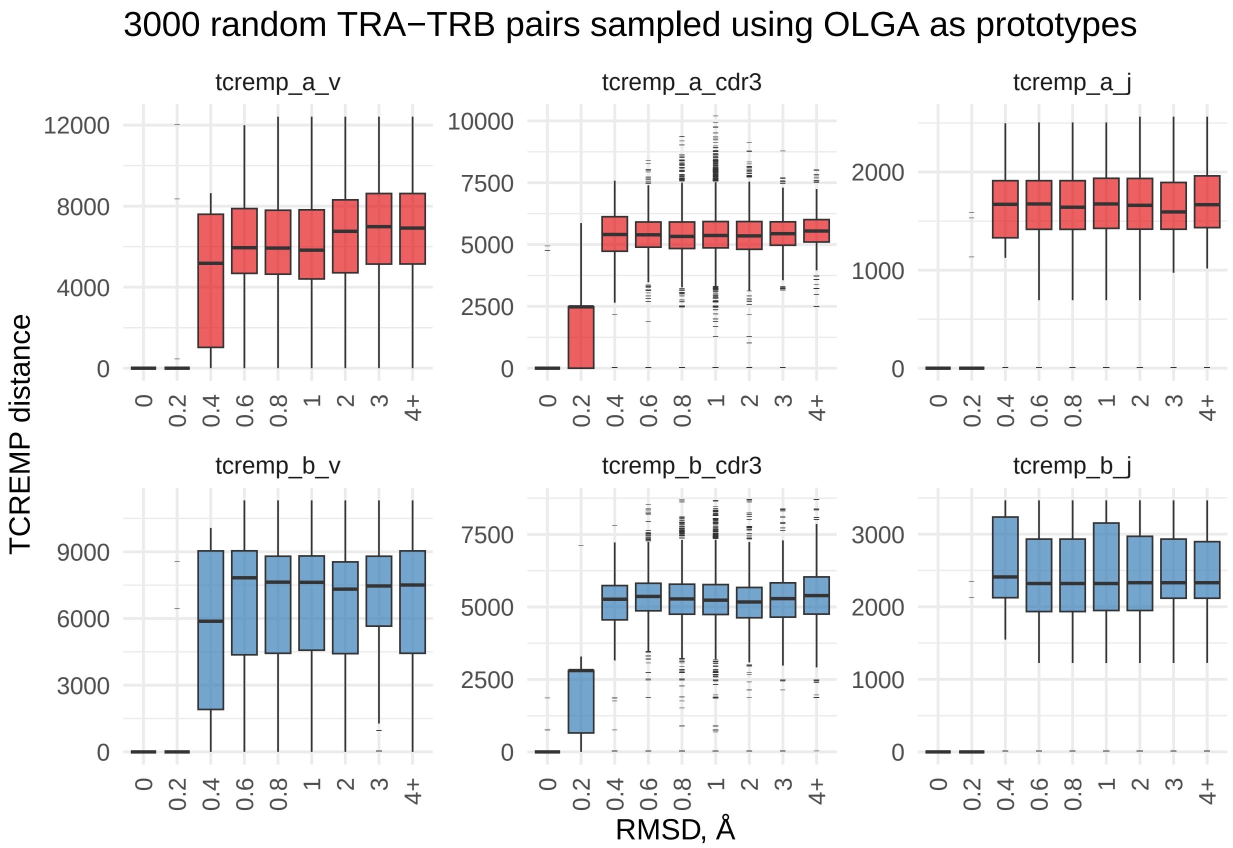
```
struct_alns_rmsd.mhc.dist2 <- struct_alns_rmsd.mhc |>
  left_join(fetch_dist("a_v")) |>
  left_join(fetch_dist("a_cdr3")) |>
  left_join(fetch_dist("a_j")) |>
  left_join(fetch_dist("b_v")) |>
  left_join(fetch_dist("b_cdr3")) |>
  left_join(fetch_dist("b_j"))
```

```
## Joining with ‘by = join_by(clone_id.from, clone_id.to)‘
## Joining with ‘by = join_by(clone_id.from, clone_id.to)‘
## Joining with ‘by = join_by(clone_id.from, clone_id.to)‘
## Joining with ‘by = join_by(clone_id.from, clone_id.to)‘
## Joining with ‘by = join_by(clone_id.from, clone_id.to)‘
## Joining with ‘by = join_by(clone_id.from, clone_id.to)‘
```

Plot them

```
plt3 <- struct_alns_rmsd.mhc.dist2 |>
  melt(id.vars = 1:5,
       variable.name = "region",
       value.name = "distance") |>
  filter(clone_id.from > clone_id.to) |>
  mutate(chain = substr(region, 8, 8),
         rmsdr = round(rmsd * roundfactor, 0) / roundfactor) |>
  mutate(rmsdr = ifelse(rmsdr > 3, "4+", signif(rmsdr, 1)) |>
           as.factor()) |>
  ggplot(aes(x = rmsdr,
             group = rmsdr,
             y = distance,
             fill = chain)) +
  geom_boxplot(alpha = 0.7, size = 0.3, outlier.size = 1.0,
               outlier.shape = "-") +
  xlab("RMSD, Å") +
  ylab("TCREMP distance") +
  facet_wrap(. ~ region, scales = "free") +
  scale_fill_brewer(guide = F, palette = "Set1") +
  ggtitle("3000 random TRA-TRB pairs sampled using OLGA as prototypes") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
plt3
```

```
## Warning: Removed 13206 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```



```
pdf("figs_5.pdf", width=7, height=11)
(plt1 +
   scale_x_continuous("", limits = c(0, 4)) +
   theme(plot.tag = element_text(size = 16, face="bold"))) /
  (plt2 +
     xlab("") +
     theme(plot.tag = element_text(size = 16, face="bold"))) /
  (plt3 + theme(plot.tag = element_text(size = 16, face="bold"))) +
  plot_annotation(tag_levels = 'a')
```

```
## Warning: Removed 2862 rows containing non-finite outside the scale range
## (‘stat_density()‘).
```

```
## Warning: Removed 656 rows containing non-finite outside the scale range
## (‘stat_density()‘).
```

```
## Warning: Removed 13206 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
## Removed 13206 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

```
dev.off()
```

```
## pdf
##   2
```

```
#END
```