Ensemble Difference Distance Matrix (eDDM) Analysis of GPCRs

Barry J. Grant, Lars Skjærven, and Xin-Qiu Yao

2020-07-01

Background

This document provides an example application of the eDDM method to G protein-coupled receptors (GPCRs). The embeded code below is indended for reproducing[1] the results described in the main text of the associated paper. The entire process consists of four major steps: **1) Structure set preparation**, **2) structure grouping**, **3) eDDM calculation**, and **4) significant changes identification and results visualization**.

Requirement

The latest developmental version of the **Bio3D core** package is required. It can be installed using the following command:[2]

```
devtools::install_bitbucket("Grantlab/bio3d/ver_devel/bio3d")
```

Also, the latest version of the **Bio3D-eddm** package is required:

```
devtools::install_bitbucket("Grantlab/bio3d-eddm", dependencies=TRUE)
```

1. Structure Set Preparation

```
library(bio3d)
library(bio3d.eddm)
library(ggplot2)
```

We will analyze structures of the beta adregenic receptor. Starting with the sequence of the human beta-2 adregenic receptor (UNIPROT: P07550), we search the PDB database to find structures with the same or similar sequence to the query, using the BLAST method implemented in Bio3D.

```
aa <- get.seq("P07550")
blast <- blast.pdb(aa)
```

The identified structures are sorted in the descending order of sequence similarity to the query. A similarity threshold (on the E-value) is required to cut the searching result and determine the

[1] Results are subject to the dynamic status of online servers, dabases, etc. and may not be exactly same as the reported.

[2] The **devtools** R package needs to be installed in advance.

structure set for subsequent analyses. The Bio3D function `plot.blast()` or simply `plot()` can help find a suitable threshold by examining the distribution of the similarity scores.

```
hits <- plot(blast, cutoff=301)
```

```
##    * Possible cutoff values:    301 -3
##              Yielding Nhits:    24 138
##
##    * Chosen cutoff value of:    301
##              Yielding Nhits:    24
```

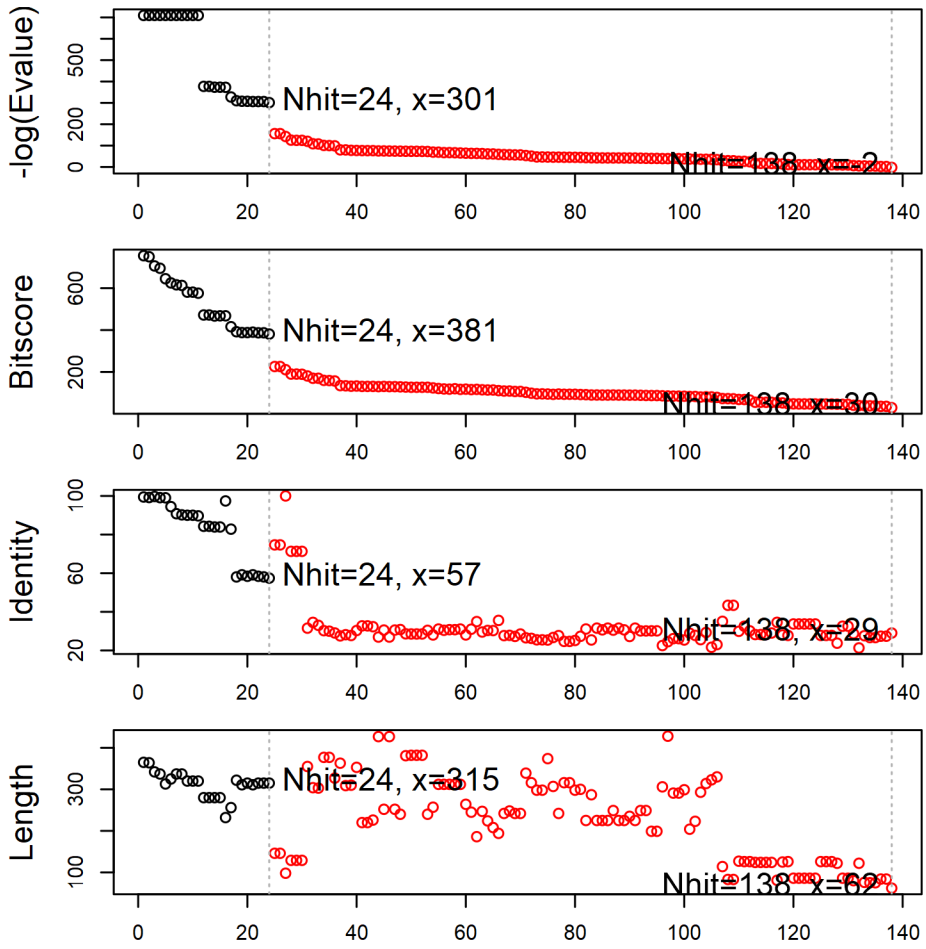

Figure 1: **BLAST report regarding the search for beta adregenic receptor structures.**

The structures can be further filtered by structural resolution (for crystallographic and cryo-EM structures). Such information can be automatically obtained using the Bio3D function, `pdb.annotate()`.

```
annotation <- pdb.annotate(hits)
pdb.id <- with(annotation, subset(hits$pdb.id, resolution<= 3.5))
annotation <- subset(annotation, resolution <= 3.5)
```

Selected structrues are then downloaded through the Bio3D function, `get.pdb()`. The optional **split=TRUE** splits the downloaded files into individual chains to facilitate subsequent analyses.

```
files <- get.pdb(pdb.id, path="pdbs", split=TRUE)
```

All selected structures are **aligned**. This is to facilitate structural comparisons between "equivalent" or aligned residues.

```
pdbs <- pdbaln(files)
```

Side-note:

A good practice is to save the generated `pdbs` and `annotation` for future uses. This can be done by following commands:

```
gpcr <- list(pdbs=pdbs, annotation=annotation)
save(gpcr, file="gpcr.RData")
```

To load a saved dataset, type following commands:

```
load("gpcr.RData")
attach(gpcr)
```

2. Structure Grouping

The eDDM analysis compares structural ensembles under distinct ligation, activation, etc. conditions. **At least two groups** of structures are required. The grouping of structures can be either from available structural annotations (e.g., the ligand identity bound with each structure; such information is available in the above prepared `annotation` object) or from a structural clustering analysis. The latter approach is more general and requires minimal prior knowledge about the system. In the following example, we use PCA of the distance matrices followed by a conventional hierachical clustering in the PC1-PC2 subspace. **Three major clusters** are identified.

```
# Update the aligned structures for including all heavy atoms.
pdbs.aa <- read.all(pdbs)
```

```
# Calculate distance matrices.
dm <- dm(pdbs.aa, all.atom=TRUE)
```

```
##   |                                                                      |
```

```
# Perform PCA of distance matrices.
pc <- pca.array(dm)
```

```
## NOTE: Removing 178047 upper triangular gap cells with missing data
##       retaining 17578 upper triangular non-gap cells for analysis.
```

```
plot.pca.scree(pc)
```

```
# Perform structural clustering in the PC1-PC2 subspace.
hc <- hclust(dist(pc$z[, 1:2]))
grps <- cutree(hc, k=3)
```

```
plot(pc, pc.axes=c(1,2), col=grps)
```

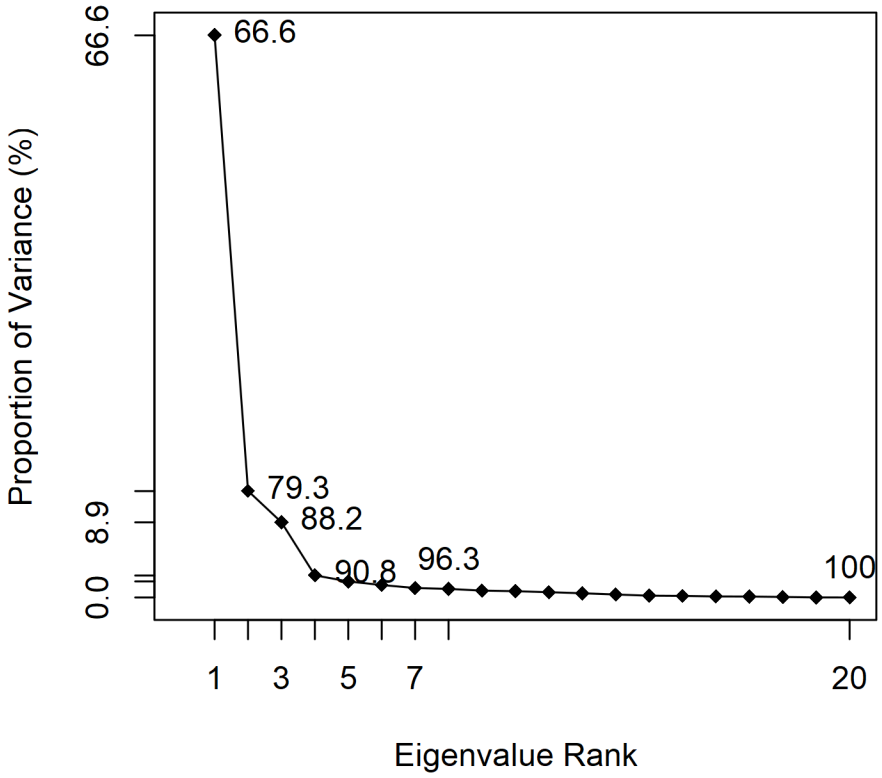

Figure 2: **Scree plot of PCA.**

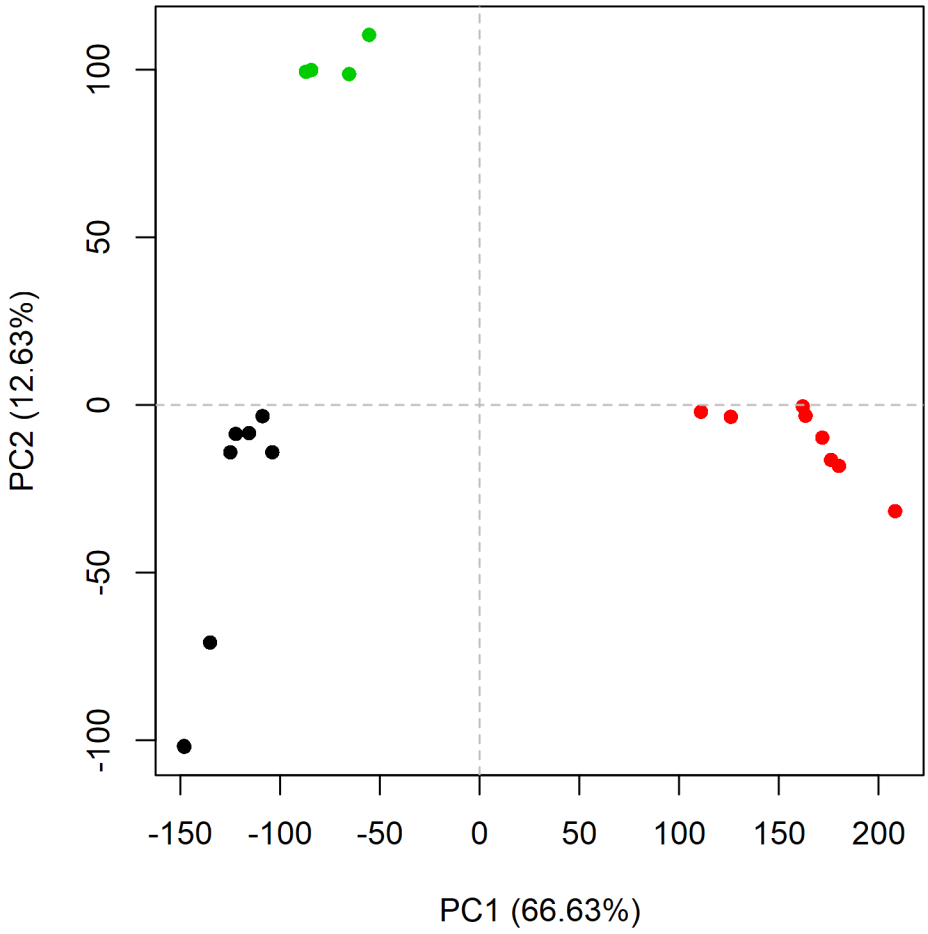

Figure 3: **Conformer plot.** Each point represents a structure with point color indicating the membership ID of the clustering.

3. eDDM calculation

The `eddm()` function calculates the difference mean distance between groups for each residue pair and statistical significance assessed using a two-sample Wilcoxon test. Long-distance pairs in all structures are omitted. This step is controled by the `mask` option of the function. In the following, we use `mask="cmap"`, meaning internally calculated contact maps are used to exclude residue pairs of (constantly) long distance (i.e., pairs that do not show a stable contact in any structural group).

```
tbl <- eddm(pdbs.aa, grps=grps, dm=dm, mask="cmap")
```

4. Significant Changes Identification and Results Visualization

In this step, **significant** distance changes are identified. A significant change is defined by a p-value of the statistical test lower than the threshold **alpha** and the absolute mean distance change is above the threshold **beta**. Below, we use **alpha=0.005** and **beta=1.0 (angstrom)**. Also, only **"switching"** residues are returned, i.e., residues showing rearranged contact networks in different groups. For clarity, only Group 1 ("inactive") and Group 2 ("active") are compared.

```
keys <- subset(tbl, alpha=0.005, beta=1.0, switch.only=TRUE)
keys
```

```
##
## Class:
##   eddm, data.frame
##
## # Residue pairs:
##   36 (34 unique residues)
##
## # Groups:
##   2 (1, 2)
##
##                 a         b   i   j c.1 c.2  d.1   d.2 sd.1 sd.2 dm.1_2 z.1_2
## 212-274  THR68(A) ASP130(A) 212 274  NA   1 4.23  2.85 0.48 0.34   1.38  2.86
## 274-285 ASP130(A) TYR141(A) 274 285  NA   1 7.10  2.84 0.43 0.27   4.26  9.86
## 274-287 ASP130(A) SER143(A) 274 287   1  NA 2.67 10.34 0.20 0.42   7.66 37.74
## 284-287 LYS140(A) SER143(A) 284 287  NA   1 6.20  3.27 0.60 0.17   2.93  4.92
## 212-285  THR68(A) TYR141(A) 212 285  NA   1 6.15  3.40 0.44 0.41   2.75  6.25
## 275-285 ARG131(A) TYR141(A) 275 285   1  NA 3.42  4.76 0.72 0.69   1.33  1.85
##          pv.1_2
## 212-274 0.00031
## 274-285 0.00016
## 274-287 0.00016
## 284-287 0.00016
## 212-285 0.00016
## 275-285 0.00470
## <...>
```

Results can be viewed as a 2D plot showing the distribution and magnitude of significant changes along the protein sequence.

```
plot(keys, pdbs=pdbs, full=TRUE, resno=NULL, sse=pdbs$sse[1, ], type="tile", labels=TRUE,
     labels.ind=c(1:3, 17:19))
```

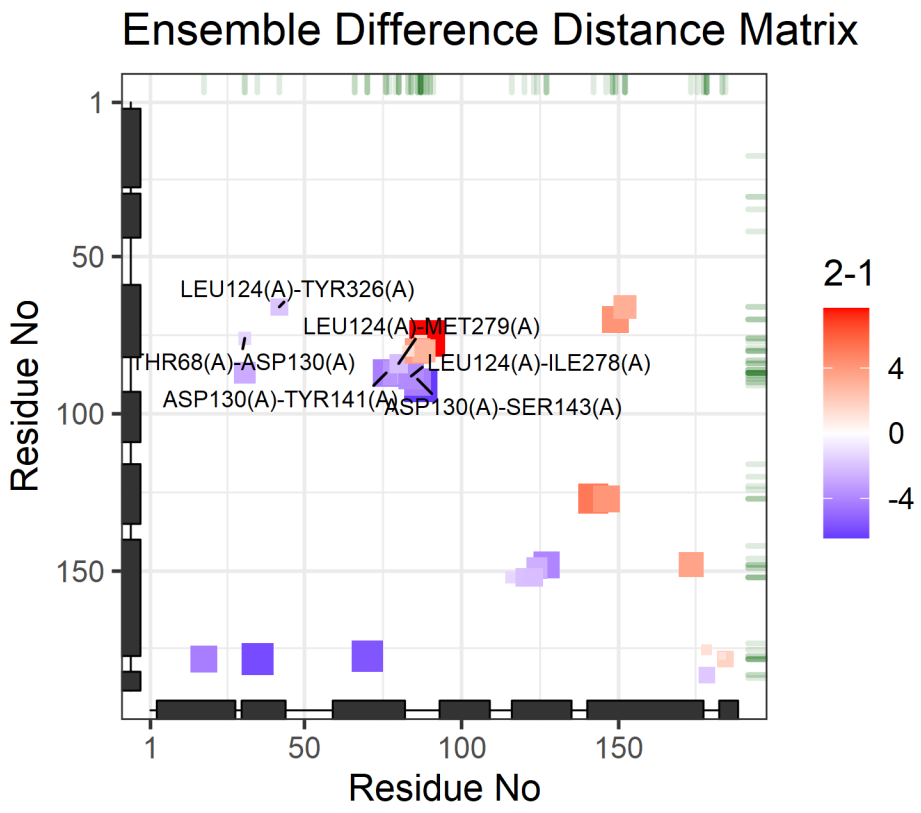

Figure 4: **The 'tile' plot of identified significant distance changes.**

Also, box-whisker plots of select pairs are displayed. Here, we focus on the two regions mentioned in the main text, i.e., the region involving ICL2 and the region in the middle of TM helices.

```
p <- boxplot(keys, dm, grps, inds = c(1:3,17:19))
p <- p + scale_color_manual(values=c("gray30", "#F8766D"), name="Group")
```

```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
```

```
print(p)
```

Finally, 3D views of identified residue pairs mapped onto GPCR structures are generated. This is done by the `pymol.eddm()` Bio3D function. Before that, structural fitting is performed to facilitate the visual comparision.[3]

[3]Structural fitting is not a required step for the eDDM calculation itself.

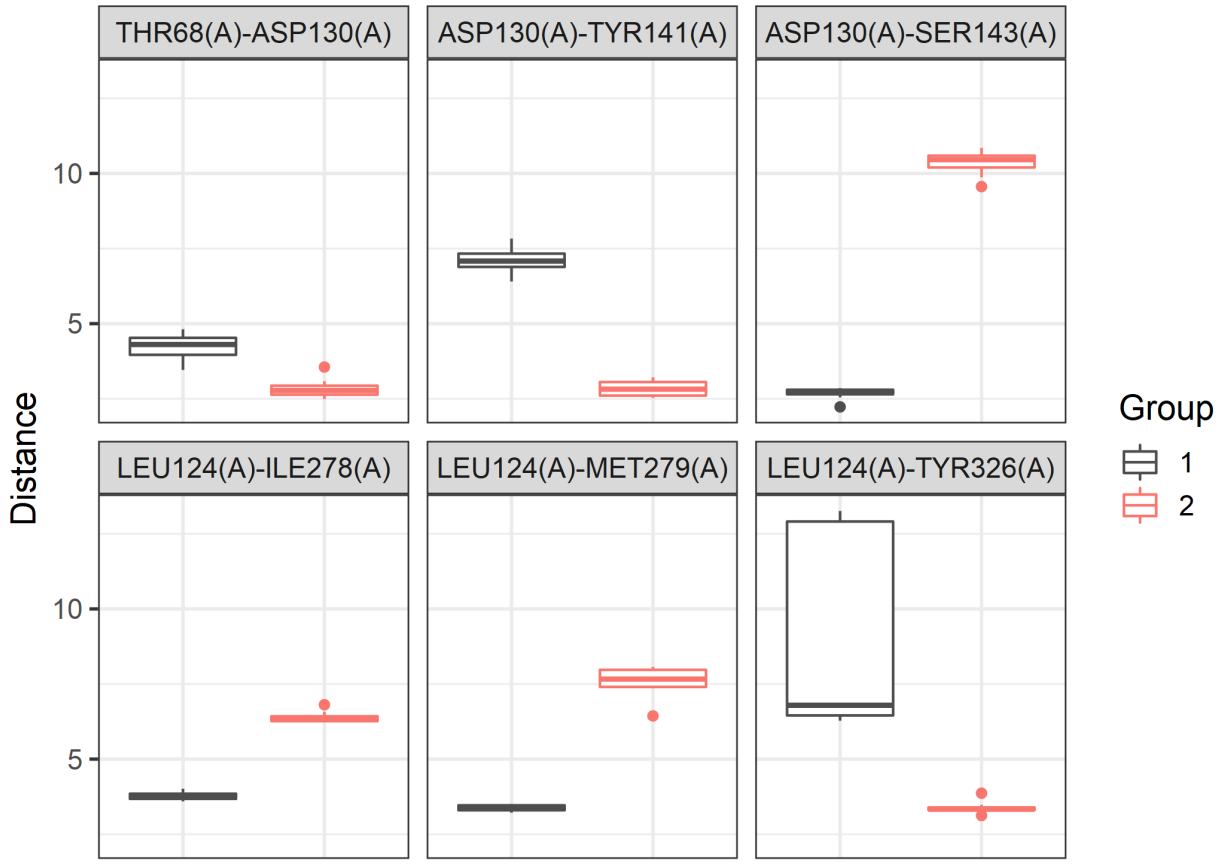

Figure 5: **Box-whisker plots of select residue pairs.**

```
core <- core.find(pdbs)
xyz <- pdbfit(pdbs, inds=core, outpath="fitlsq")
pdbs$xyz <- xyz
```

```
pymol(keys, pdbs=pdbs, grps=grps, as="sticks")
```

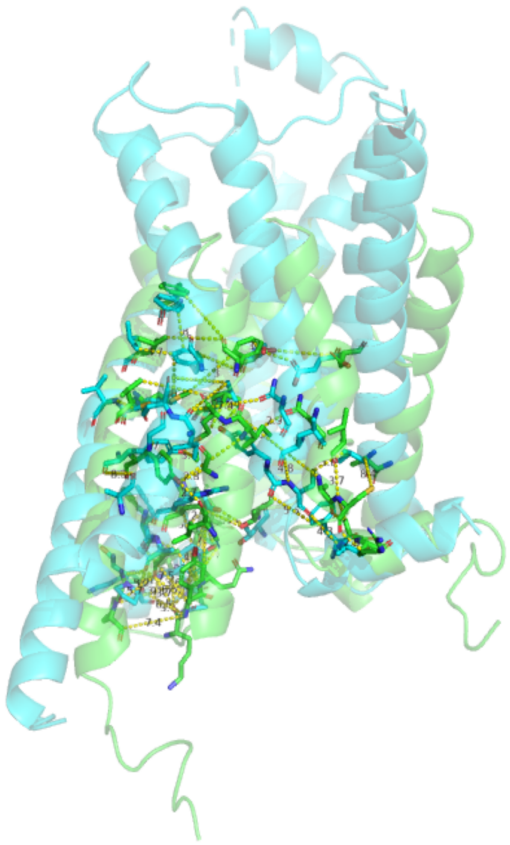

Figure 6: **Use of `pymol.eddm()` to visualize all identified residue pairs.**

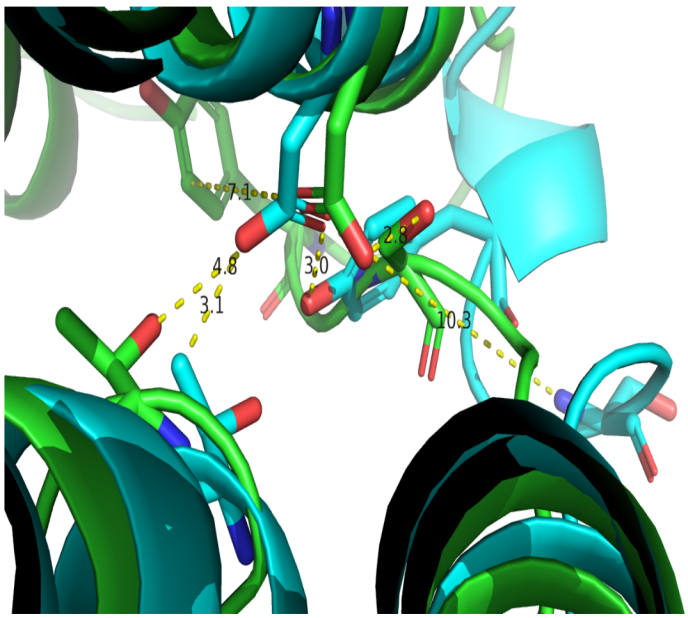

Figure 7: **A close view of the switching region near ICL2.**

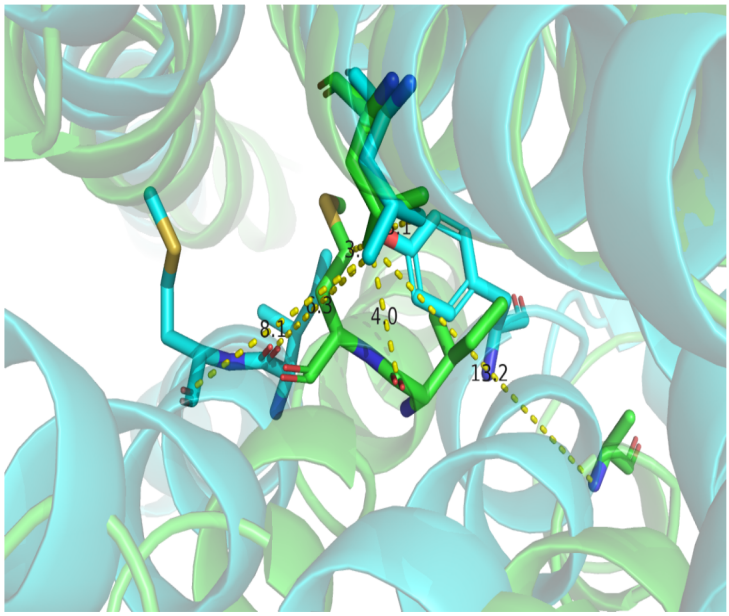

Figure 8: **A close view of the switching region in the middle of TM helices.**

About this document

This document is generated by the **rmarkdown** R package. To reproduce[4] it, simply type following commands:[5]

```
library(rmarkdown)
render("eddm_gpcr.r", "pdf_document")
```

Information About the Current Bio3D Session

```
print(sessionInfo(), FALSE)
```

```
## R version 3.6.1 (2019-07-05)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 18362)
##
## Matrix products: default
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] png_0.1-7             ggplot2_3.3.2.9000    bio3d.eddm_0.1.0.9000
## [4] bio3d_2.4-1.9000      rmarkdown_2.2
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.4.6     plyr_1.8.6       compiler_3.6.1   pillar_1.4.4
##  [5] highr_0.8        bitops_1.0-6     tools_3.6.1      digest_0.6.25
##  [9] evaluate_0.14    lifecycle_0.2.0  tibble_3.0.1     gtable_0.3.0
## [13] pkgconfig_2.0.3  rlang_0.4.6      ggrepel_0.8.2    yaml_2.2.1
## [17] parallel_3.6.1   xfun_0.15        withr_2.2.0      stringr_1.4.0
## [21] dplyr_1.0.0      knitr_1.29       generics_0.0.2   vctrs_0.3.1
## [25] grid_3.6.1       tidyselect_1.1.0 glue_1.4.1       R6_2.4.1
## [29] XML_3.99-0.3     reshape2_1.4.4   farver_2.0.3     purrr_0.3.4
## [33] magrittr_1.5     scales_1.1.1     ellipsis_0.3.1   htmltools_0.4.0
## [37] colorspace_1.4-1 labeling_0.3     stringi_1.4.6    RCurl_1.98-1.2
## [41] munsell_0.5.0    crayon_1.3.4
```

[4]Results are subject to the dynamic status of online servers, dabases, etc. and may not be exactly same as the reported.

[5]The PyMol images are inserted manually.