Structural dynamics of the monoamine transporter homologue LeuT from accelerated conformational sampling and channel analysis

James R. Thomas[1,2], Patrick C. Gedeon[1,3], and Jeffry D. Madura[1]

[1]Department of Chemistry and Biochemistry and the Center for Computational Sciences, Duquesne University, Pittsburgh,PA, USA
[2]Mylan School of Pharmacy, Duquesne University, Pittsburgh,PA, USA
[3]Duke University, Department of Biomedical Engineering, Durham, NC, USA

January 27, 2014

Supplementary Information

Principal Component Analysis Results

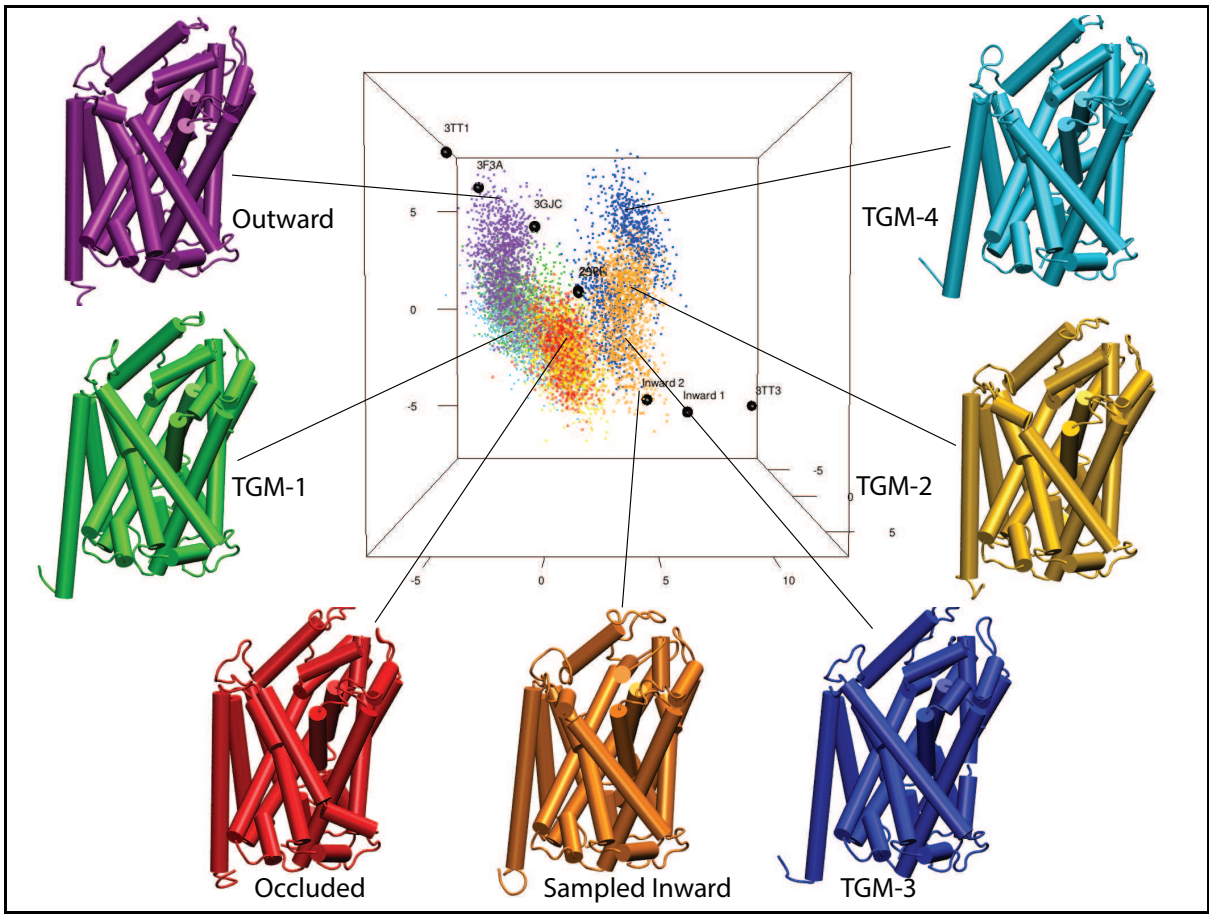

Figure S1: A three dimensional scatterplot of the PCA from all seven simulations started from the original publication[4] and updated. All data points are colored by simulation according to Table I in the main document. Each structure shown corresponds to one of the isolated LeuT conformations found from this analysis. The names next to each protein representation correspond to the convention used in this manuscript.

Intracellular R5–Y268 Cationic Stabilization Interaction

From the proposed mechanism, as the distance between R5-D369 increased, the R5-Y268 should have decreased as the positive nitrogenous side chain of R5 moved to stabilize its charge with the electron rich Y268 aromatic residue[5]. However, this behavior was not seen.

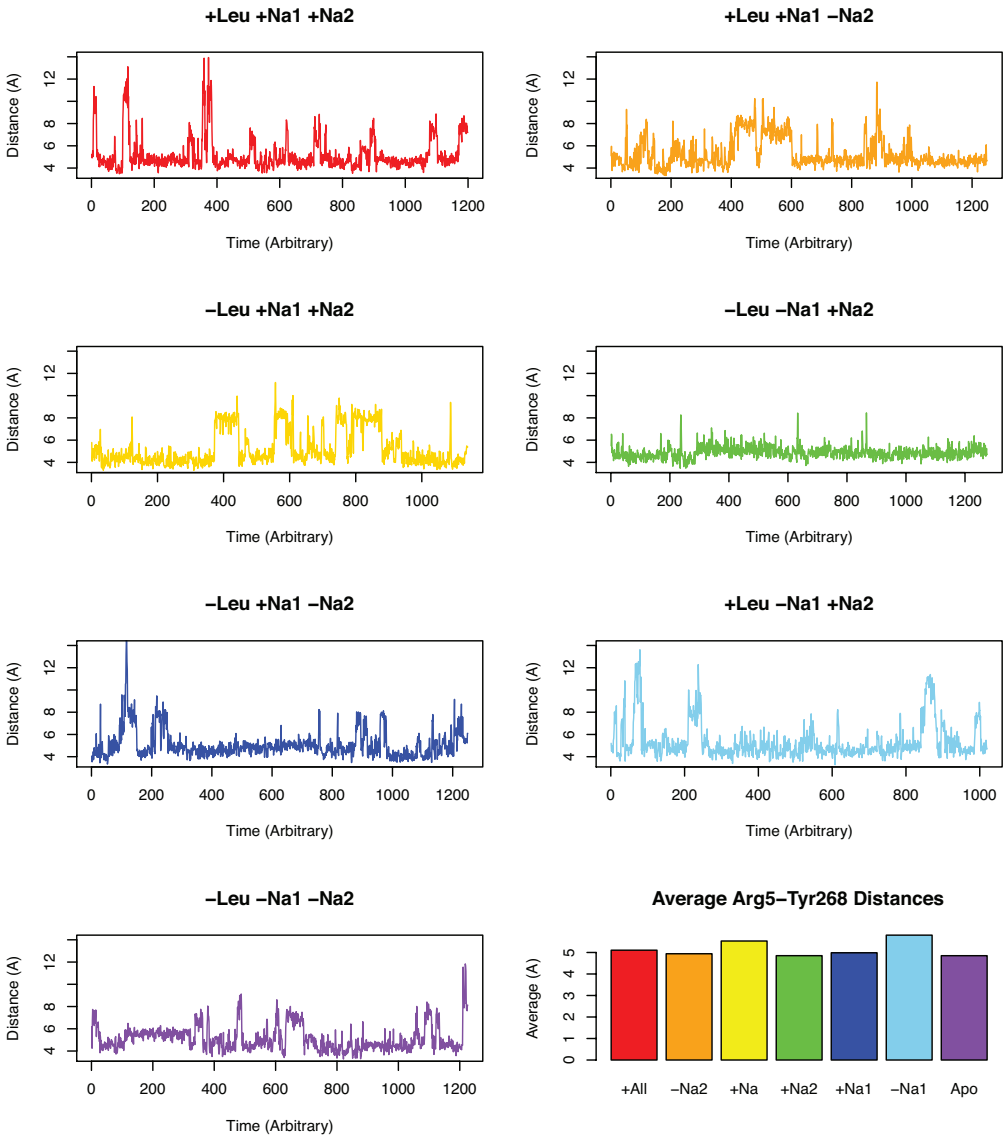

Figure S2: Distances observed between R5–Y268 involved in a proposed intracellular cation–π stabilization interaction[5]. The averages plot is the average after 150 ns in order to prevent sampling error from early post minimization dynamics.

TM10–EL4 Interaction Graph

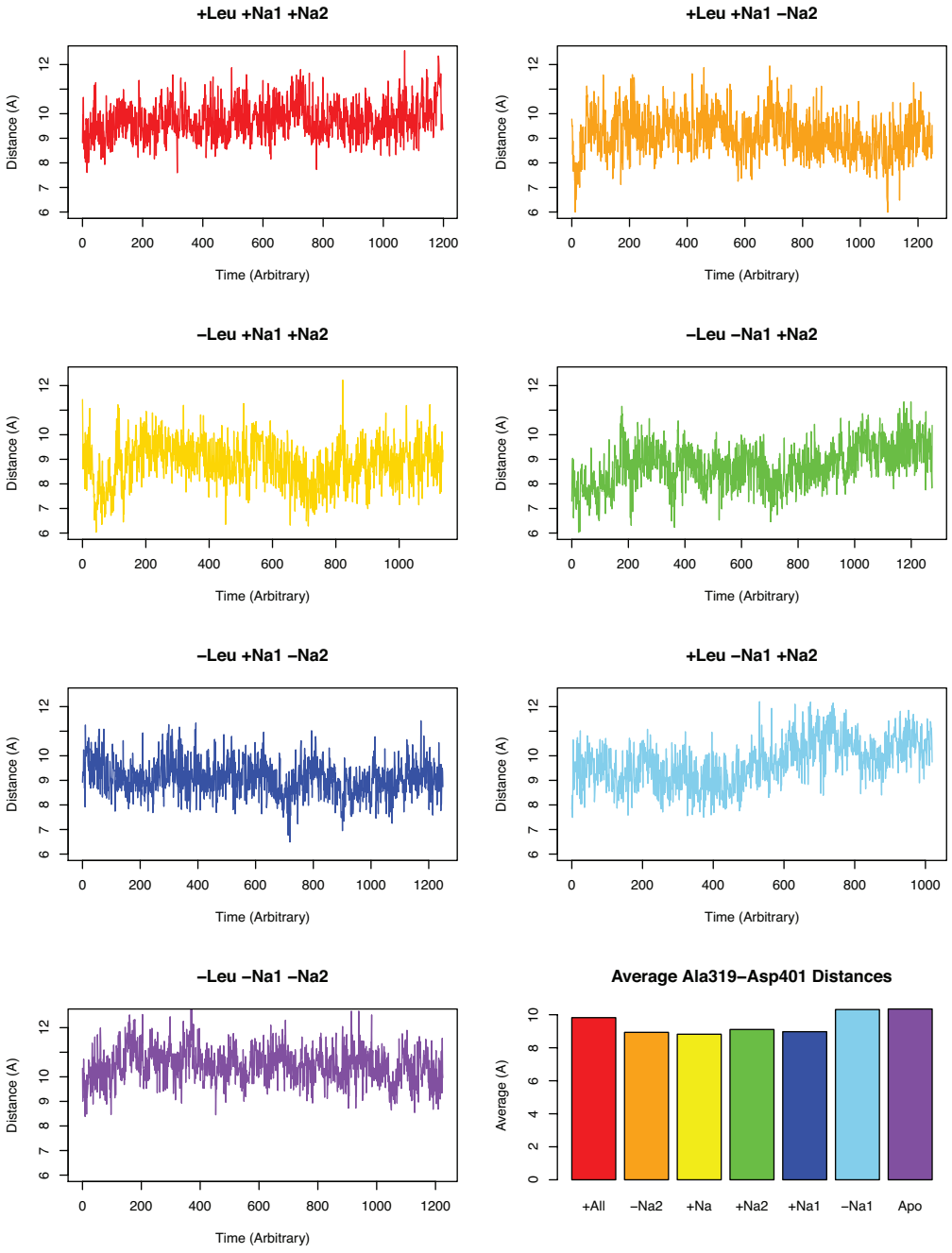

Figure S3: Distances observed between A319–D401 which have been proposed to form a coordinated interaction between TM10 and EL4 during the inward transition in order to occlude the extracellular vestibule[46]. Discussion of these results are presented in the manuscript. The averages plot is the average after 150 ns.

TM1–TM6 Coordination Interaction

It has been hypothesized that the coordinated movements that may be necessary between TM1 and its symmetrical helix TM6 could be facilitated by a hydrogen bond that forms and breaks between N21 and S256[50]. In our simulations, the distance between these residues remained mostly constant with the most fluctuations occurring in the simulation with only Na1 bound (see Figure S4).

As of the current data, not proper conclusions from this data can be supported except that the Na1 only structure holds a unique place in the transport cycle and could support the hypothesis that Na2 is the first to be transported intracellularly. It is difficult to compare this result to other data since the best comparison would be to compare the published crystallized inward structure to the occluded structure. However, the 2012 crystallized inward (3TT3)[46] file does not have the side chains of either N21 or S256. Using C_α coordinates, the distance is 11.25 Å. C_α, but the occluded crystal structure (2A65)[1] distance is 7.02 Å. At this time, it is unknown if this distance is truly an opening of the structures or an artifact of the resolutions and experimental conditions of the two different structures since the side chain orientation of the inward structure is unknown.

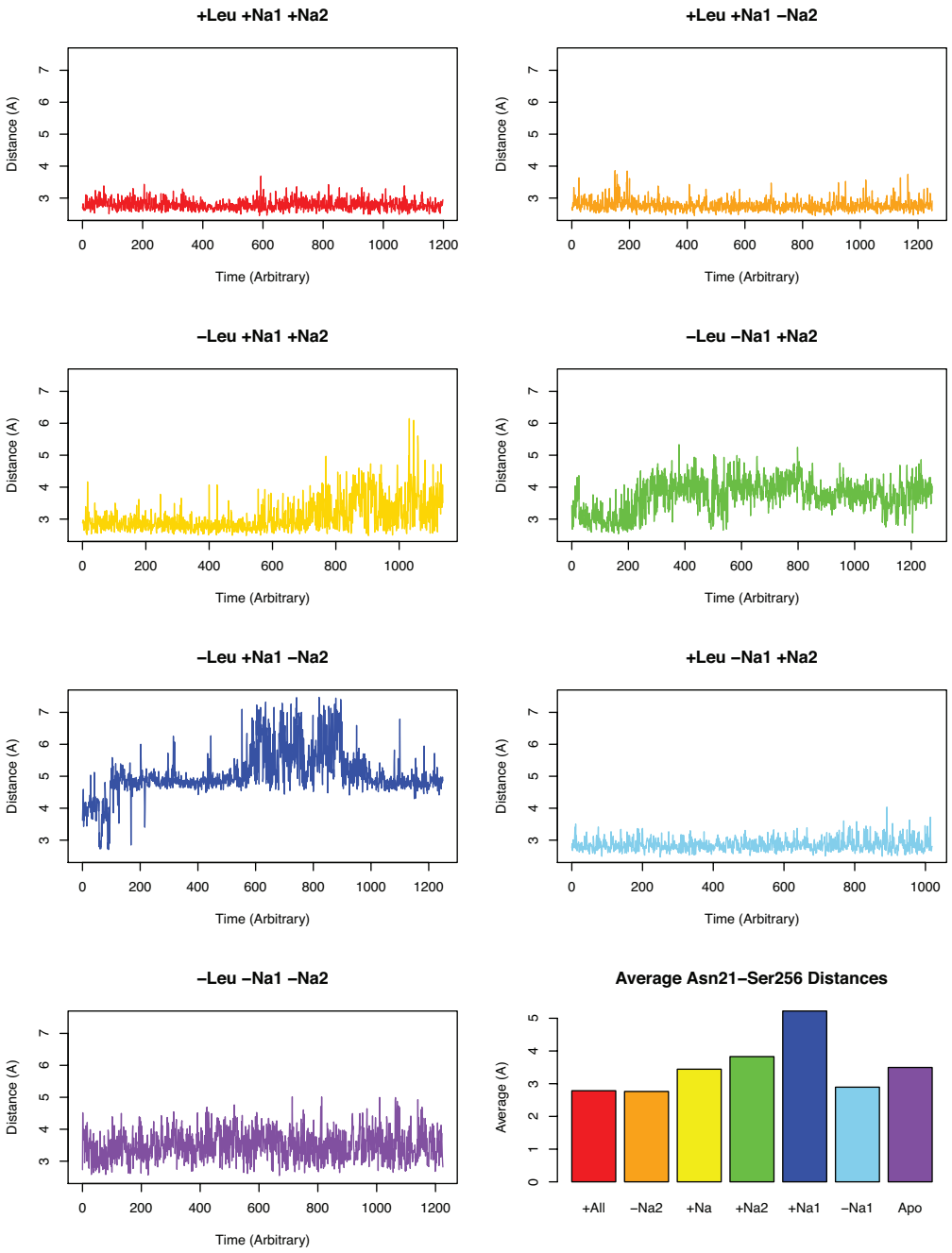

Figure S4: Distances observed in the intracellular N21–S256 Interaction. N21–S256 are proposed to form a hydrogen bond to stabilize the TM1–TM6 distances[50].

RMSD values of Trajectories to the 2012 Gouaux Inward Crystal

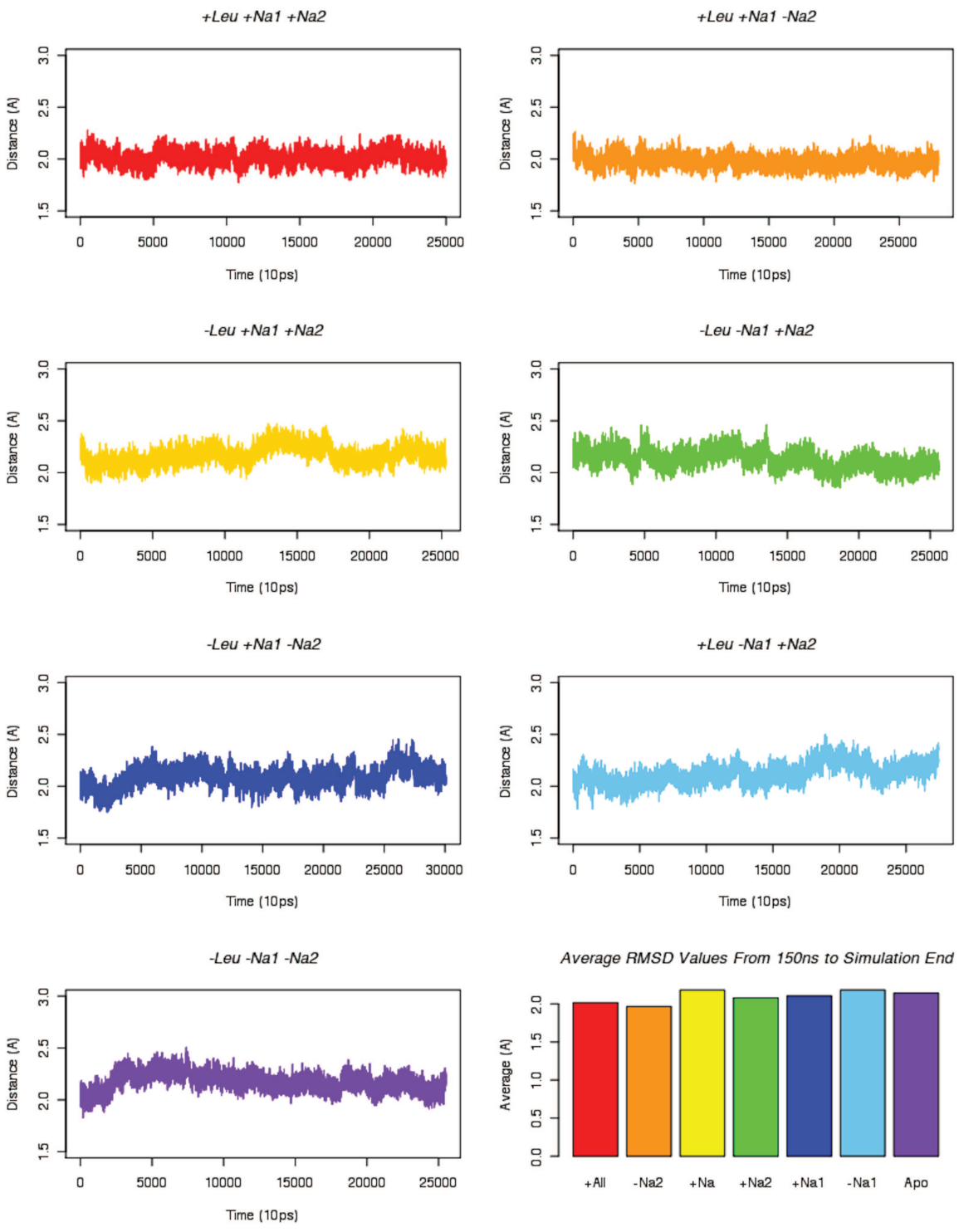

Figure S5: RMSD values between simulations and Gouaux inward structure[46] for all simulation frames taken at 10 ps. The averages plot is the average after 150 ns in order to prevent sampling error from early post minimization dynamics.

Analysis Codes

Bio3D Simulation Analysis

```
# Script file for Plotting a PCA sample space of LeuT trajectories seeded with
# Crystal Structures
# This script contains the code for the 2D PCA plots with histogram axes in the
# 2012 article.
# The 3D plot in the article utilizes the same analysis code and only modifies
# the plotting, but it was considered
# more beneficial to present the code with the histogram axes.
# You must be running R64 not R. R on a 64-bit machine will still only run in
# 32-bit.
# You must have installed the bio3D package for R by Barry Grant.
# You must have installed the bigmemory package for R.

#########################################################################
# DO NOT DO ANYTHING ELSE ON THE COMPUTER WHEN YOU RUN THIS SCRIPT. #
# IT EASILY DEVOURS RAM.                                                #
#########################################################################
#Load Libraries

library(bio3d)
library(rgl) # This is not actually used in this script. This is for the 3-D
# plot in the articles, but the 3D code is not supplied.
library(bigmemory)


#Read In Reference Structure, Trajectory, and Xrays
pdb <- read.pdb("LeuT_reference.pdb")
dcd <- read.dcd("all_runs_to_250ns_total.dcd",big=TRUE)

xrays <- pdbaln(c("LeuT_reference.pdb", "2A65.pdb","2Q6H.pdb", "3GJC.pdb",
                  "3F3A.pdb","LeuT_Inward_TMD1_Shaikh.pdb",
                  "LeuT_Inward_TMD2_Shaikh.pdb","Gouaux_Inward.pdb",
                  "Gouaux_Outward.pdb"))
# NOTE: pdbaln reads in the pdb files and aligns them but at the cost of having
# only c-alphas.


#Select CA for alignment
ca.ind <- atom.select(pdb, elety="CA")

#Superimpose Trajectory to reference
trj <- fit.xyz(pdb$xyz[ca.ind$xyz],dcd[,ca.ind$xyz])

#Superimpose Xrays to reference (none of them have missing residues in the
#areas of interest since they were manually added for previous analytical
#methods
# which were not utilized for publication, so gap inspection is not needed in
# this instance)
a <- fit.xyz(xrays$xyz[1,], xrays$xyz[2:nrow(xrays$xyz),])

#Indicies for residues of TMs 1b and 6a after CA's have been pulled
res.sel <- c(21:33,237:251)
crys.ind <- 0
for (i in 1:length(res.sel)){
        temp3 <- (res.sel[i]-4)*3
        temp2 <- temp3-1
        temp1 <- temp3-2
        crys.ind[3*i-2] <- temp1
        crys.ind[3*i-1] <- temp2
        crys.ind[3*i] <- temp3
}

#Combine XRAY and Trajectory superposes with the Crystals at the bottom
b <- rbind(trj[,crys.ind],a[,crys.ind])
```

```
#Peform a PCA on Transmembrane Domains TM1b and TM6a together
pc.trj <- pca.xyz(b)

# Create histograms as variables
pc1 <- hist(pc.trj$z[,1], breaks=100, freq=FALSE, xlab="",ylab="", main="")
pc2 <- hist(pc.trj$z[,2], breaks=100, freq=FALSE, xlab="",ylab="", main="")
pc3 <- hist(pc.trj$z[,3], breaks=100, freq=FALSE, xlab="",ylab="", main="")
dev.off()

#####################################################################
# Start PC1 to PC2 Plot
X11(type="cairo")
# Set up layout
nf <- layout(matrix(c(2,0,1,3), 2, 2, byrow=TRUE), c(3,0.5), c(0.5,3), TRUE)

pcpick1<-1
pcpick2<-2
# Select main section and create the PCA scree plot.
par(mar=c(3,3,1,1))
plot(pc.trj$z[which(seq(along=pc.trj$z)<=1199),pcpick1],
     pc.trj$z[which(seq(along=pc.trj$z)<=1199),pcpick2],
     col = "red", xlab="", ylab="", xlim=c(-7.5,12), ylim=c(-7.5,7.5),pch=16)
points(pc.trj$z[which((seq(along=pc.trj$z[,1])>1199)&(seq(along=pc.trj$z[,1])<=2449)),pcpick1],
       pc.trj$z[which((seq(along=pc.trj$z[,1])>1199)&(seq(along=pc.trj$z[,1])<=2449)),pcpick2],
       col = "orange",pch=16)
points(pc.trj$z[which((seq(along=pc.trj$z[,1])>2449)&(seq(along=pc.trj$z[,1])<=3714)),pcpick1],
       pc.trj$z[which((seq(along=pc.trj$z[,1])>2449)&(seq(along=pc.trj$z[,1])<=3714)),pcpick2],
       col = "yellow",pch=16)
points(pc.trj$z[which((seq(along=pc.trj$z[,1])>3714)&(seq(along=pc.trj$z[,1])<=4993)),pcpick1],
       pc.trj$z[which((seq(along=pc.trj$z[,1])>3714)&(seq(along=pc.trj$z[,1])<=4993)),pcpick2],
       col = "green",pch=16)
points(pc.trj$z[which((seq(along=pc.trj$z[,1])>4993)&(seq(along=pc.trj$z[,1])<=6248)),pcpick1],
       pc.trj$z[which((seq(along=pc.trj$z[,1])>4993)&(seq(along=pc.trj$z[,1])<=6248)),pcpick2],
       col = "blue",pch=16)
points(pc.trj$z[which((seq(along=pc.trj$z[,1])>6248)&(seq(along=pc.trj$z[,1])<=7523)),pcpick1],
       pc.trj$z[which((seq(along=pc.trj$z[,1])>6248)&(seq(along=pc.trj$z[,1])<=7523)),pcpick2],
       col = "purple",pch=16)
points(pc.trj$z[which((seq(along=pc.trj$z[,1])>7523)&(seq(along=pc.trj$z[,1])<=8793)),pcpick1],
       pc.trj$z[which((seq(along=pc.trj$z[,1])>7523)&(seq(along=pc.trj$z[,1])<=8793)),pcpick2],
       col = "sky␣blue",pch=16)
points(pc.trj$z[8794:8801,pcpick1],pc.trj$z[8794:8801,pcpick2], col = "black",
       pch=16, add=TRUE)
labs <- c("2A65", "2Q6H", "3GJC","3F3A", "Inward␣1", "Inward␣2","3TT3","3TT1")
text(pc.trj$z[8794:8801,pcpick1],pc.trj$z[8794:8801,pcpick2]+0.5, labs)

# Generate x-axis histogram
par(mar=c(0,3,1,1))
barplot(pc1$counts, axes=FALSE, horiz=FALSE, main="PC1",col="black")

# Y-axis histogram
par(mar=c(3,0,1,1))
barplot(pc2$counts, axes = FALSE, horiz = TRUE, main = "PC2",col="black")

savePlot(filename = "PC1-PC2.png")

#####################################################################
# Start PC1 to PC3 Plot
X11(type="cairo")
# Set up layout
nf <- layout(matrix(c(2,0,1,3), 2, 2, byrow=TRUE), c(3,0.5), c(0.5,3), TRUE)

pcpick1<-1
pcpick2<-3
# Select main section and create the PCA scree plot.
par(mar=c(3,3,1,1))
plot(pc.trj$z[which(seq(along=pc.trj$z)<=1199),pcpick1],
     pc.trj$z[which(seq(along=pc.trj$z)<=1199),pcpick2],
```

```
     col = "red", xlab="", ylab="", xlim=c(-7.5,12), ylim=c(-7.5,7.5),pch=16)
points(pc.trj$z[which((seq(along=pc.trj$z[,1])>1199)&(seq(along=pc.trj$z[,1])<=2449)),pcpick1],
       pc.trj$z[which((seq(along=pc.trj$z[,1])>1199)&(seq(along=pc.trj$z[,1])<=2449)),pcpick2],
       col = "orange",pch=16)
points(pc.trj$z[which((seq(along=pc.trj$z[,1])>2449)&(seq(along=pc.trj$z[,1])<=3714)),pcpick1],
       pc.trj$z[which((seq(along=pc.trj$z[,1])>2449)&(seq(along=pc.trj$z[,1])<=3714)),pcpick2],
       col = "yellow",pch=16)
points(pc.trj$z[which((seq(along=pc.trj$z[,1])>3714)&(seq(along=pc.trj$z[,1])<=4993)),pcpick1],
       pc.trj$z[which((seq(along=pc.trj$z[,1])>3714)&(seq(along=pc.trj$z[,1])<=4993)),pcpick2],
       col = "green",pch=16)
points(pc.trj$z[which((seq(along=pc.trj$z[,1])>4993)&(seq(along=pc.trj$z[,1])<=6248)),pcpick1],
       pc.trj$z[which((seq(along=pc.trj$z[,1])>4993)&(seq(along=pc.trj$z[,1])<=6248)),pcpick2],
       col = "blue",pch=16)
points(pc.trj$z[which((seq(along=pc.trj$z[,1])>6248)&(seq(along=pc.trj$z[,1])<=7523)),pcpick1],
       pc.trj$z[which((seq(along=pc.trj$z[,1])>6248)&(seq(along=pc.trj$z[,1])<=7523)),pcpick2],
       col = "purple",pch=16)
points(pc.trj$z[which((seq(along=pc.trj$z[,1])>7523)&(seq(along=pc.trj$z[,1])<=8793)),pcpick1],
       pc.trj$z[which((seq(along=pc.trj$z[,1])>7523)&(seq(along=pc.trj$z[,1])<=8793)),pcpick2],
       col = "sky blue",pch=16)
points(pc.trj$z[8794:8801,pcpick1],pc.trj$z[8794:8801,pcpick2], col = "black",
       pch=16, add=TRUE)
labs <- c("2A65", "2Q6H", "3GJC","3F3A", "Inward 1", "Inward 2","3TT3","3TT1")
text(pc.trj$z[8794:8801,pcpick1],pc.trj$z[8794:8801,pcpick2]+0.5, labs)

# Generate x-axis histogram
par(mar=c(0,3,1,1))
barplot(pc1$counts, axes=FALSE, horiz=FALSE, main="PC1",col="black")

# Y-axis histogram
par(mar=c(3,0,1,1))
barplot(pc3$counts, axes = FALSE, horiz = TRUE, main = "PC3",col="black")

savePlot(filename = "PC1-PC3.png")

#################################################################
# Start PC2 to PC3 Plot
X11(type="cairo")
# Set up layout
nf <- layout(matrix(c(2,0,1,3), 2, 2, byrow=TRUE), c(3,0.5), c(0.5,3), TRUE)

pcpick1<-2
pcpick2<-3
# Select main section and create the PCA scree plot.
par(mar=c(3,3,1,1))
plot(pc.trj$z[which(seq(along=pc.trj$z)<=1199),pcpick1],
     pc.trj$z[which(seq(along=pc.trj$z)<=1199),pcpick2],
     col = "red", xlab="", ylab="", xlim=c(-7.5,7.5), ylim=c(-7.5,7.5),pch=16)
points(pc.trj$z[which((seq(along=pc.trj$z[,1])>1199)&(seq(along=pc.trj$z[,1])<=2449)),pcpick1],
       pc.trj$z[which((seq(along=pc.trj$z[,1])>1199)&(seq(along=pc.trj$z[,1])<=2449)),pcpick2],
       col = "orange",pch=16)
points(pc.trj$z[which((seq(along=pc.trj$z[,1])>2449)&(seq(along=pc.trj$z[,1])<=3714)),pcpick1],
       pc.trj$z[which((seq(along=pc.trj$z[,1])>2449)&(seq(along=pc.trj$z[,1])<=3714)),pcpick2],
       col = "yellow",pch=16)
points(pc.trj$z[which((seq(along=pc.trj$z[,1])>3714)&(seq(along=pc.trj$z[,1])<=4993)),pcpick1],
       pc.trj$z[which((seq(along=pc.trj$z[,1])>3714)&(seq(along=pc.trj$z[,1])<=4993)),pcpick2],
       col = "green",pch=16)
points(pc.trj$z[which((seq(along=pc.trj$z[,1])>4993)&(seq(along=pc.trj$z[,1])<=6248)),pcpick1],
       pc.trj$z[which((seq(along=pc.trj$z[,1])>4993)&(seq(along=pc.trj$z[,1])<=6248)),pcpick2],
       col = "blue",pch=16)
points(pc.trj$z[which((seq(along=pc.trj$z[,1])>6248)&(seq(along=pc.trj$z[,1])<=7523)),pcpick1],
       pc.trj$z[which((seq(along=pc.trj$z[,1])>6248)&(seq(along=pc.trj$z[,1])<=7523)),pcpick2],
       col = "purple",pch=16)
points(pc.trj$z[which((seq(along=pc.trj$z[,1])>7523)&(seq(along=pc.trj$z[,1])<=8793)),pcpick1],
       pc.trj$z[which((seq(along=pc.trj$z[,1])>7523)&(seq(along=pc.trj$z[,1])<=8793)),pcpick2],
       col = "sky blue",pch=16)
points(pc.trj$z[8794:8801,pcpick1],pc.trj$z[8794:8801,pcpick2], col = "black",
       pch=16, add=TRUE)
labs <- c("2A65", "2Q6H", "3GJC","3F3A", "Inward 1", "Inward 2","3TT3","3TT1")
```

```
text(pc.trj$z[8794:8801,pcpick1],pc.trj$z[8794:8801,pcpick2]+0.5, labs)

# Generate x-axis histogram
par(mar=c(0,3,1,1))
barplot(pc3$counts, axes=FALSE, horiz=FALSE, main="PC3",col="black")

# Y-axis histogram
par(mar=c(3,0,1,1))
barplot(pc2$counts, axes = FALSE, horiz = TRUE, main = "PC2",col="black")

savePlot(filename = "PC2-PC3.png")
```

HOLE Scripts

Running HOLE

```
# Function script to run HOLE while in VMD
# Load the topology file and the trajectory
# into VMD. Then run the hole_over_time function
proc hole_over_time {} {
     # Customize the following lines to set the paths to the hole executable
     # and the radius file.
  set holebin /usr/local/hole2/exe/hole
  set holerad /usr/local/hole2/rad/simple.rad
  set sph_process /usr/local/hole2/exe/sph_process
  set sos_triangle /usr/local/hole2/exe/sos_triangle
  set tmp /tmp
  set qpt_conv /usr/local/hole2/exe/qpt_conv
  # Customize the following to set default values. See the runhole
  # comments for what these parameters do.
  #cvect - vector along which the pore lies, cpoint - point within the pore
    set cvect [list 0 0 1]
  set cpoint [list 39.41150 38.46850 49.77550]
  set sample 0.5
  set endrad 15.
  set mol top
    # use frame 0 for the reference
      set num_steps [molinfo $mol get numframes]

    #loop over all frames in the trajectory

for {set frame 0} {$frame < $num_steps} {incr frame} {
    puts "Calculating hole $frame "
    set sphpdb outputhole_$frame.sph

# write coordinates to files
puts $mol
set sel [atomselect $mol "protein" frame $frame]
set pdb tmpholeinputfiles_$frame.pdb
$sel writepdb $pdb

  # construct HOLE input string
  set str "\ncoord $pdb\n"
  append str "radius $holerad\n"
  append str "cvect $cvect\n"
  append str "cpoint $cpoint\n"
  append str "sample $sample\n"
  append str "endrad $endrad\n"
  append str "sphpdb $sphpdb\n"

# Call HOLE and collect output
puts "Calling HOLE..."

puts "$holebin $str "
flush stdout
  set result [exec "$holebin" "<< $str >> hole.log"]

file delete $pdb
```

}}

Converting HOLE to VMD channel

```
#!/bin/bash
#Converts VMD hole output qpt files to VMD readable
#Change the range for your data set before running

echo "Change the range for your data set before running"

for i in {0..8083}; do
/usr/local/hole2/exe/sph_process -dotden 15 -color outputhole_$i.sph outputhole_$i.qpt;
 ( echo "D";
 echo "outputhole_$i.qpt";
echo "outputhole_$i.vmd_plot";
 echo "1";
 ) | /usr/local/hole2/exe/qpt_conv;
rm outputhole_$i.qpt;
echo "Finished with outputhole_$i.vmd_plot";
 done
```

Visualizing HOLE in VMD with the Trajectory

```
# Draws a channel previously calculated by
# HOLE within VMD and animates the change by frame.
#
# You must have already run hole_trj.tcl and sph_to_vmd.bsh first.
#
# Start a new session of vmd and load your molecule with
# its trajectory. Then run with "source draw_hole_trj.tcl".

proc draw_hole_trj {args} {
  set molid 0
  set frame [molinfo $molid get frame]
  source outputhole_$frame.vmd_plot
}

trace variable vmd_frame(0) w draw_hole_trj
animate goto 0
```