Roll number: 1068

Molecular Dynamic Simulation of Interaction between C-terminal SH2 Domain of SHP-2 and CagA

Word count: 1893

Abstract

In this study, we investigated the interaction of the C-terminal SH2 structural domain of SHP-2 protein with the EPIYA peptide of *Helicobacter pylori* CagA protein by molecular dynamic simulations. The simulation results showed that phosphorylated EPIYA and EPIYpA-D exhibited higher stability in interacting with the SH2 domain compared with dephosphorylated EPIYA and EPIpYA-C respectively. In addition, by comparing the stability of the N and C-terminal SH2 domains in binding to EPIpYA, it was found that the N-terminal SH2 structural domain may have a relatively higher affinity for CagA proteins.

Although the results of Root Mean Square Deviation (RMSD) analysis were consistent with literature data, proving the reliability of the molecular dynamic simulation with GROMACS, the results of Radius of Gyration (RG) analysis failed to reflect the same trend. In order to save time computational resources, the SHP-2 peptide was artificially truncated in this study, which may have affected the RMSD and RG values and their fluctuations. Therefore, it is recommended that future studies use the intact SHP-2 protein for simulations and increase the number and duration of simulations to minimize systematic errors. In addition, only GROMACS software was used for simulations in this study, and future studies could try using other simulation software such as LAMMPS and VASP and compare the results.

The results of this study provide a basis for further verification of whether the N-terminal SH2 structural domain of SHP-2 protein has a higher affinity for CagA protein than the C-terminal SH2 structural domain by surface plasmon resonance analysis, which will help to gain a deeper understanding of the mechanism of the effect of the H. pylori CagA protein on SHP-2 protein and its downstream pathway.

Introduction

Helicobacter pylori is a worldwide trigger of gastrointestinal diseases such as atrophic gastritis and peptic ulcer disease (Higashi, Tsutsumi, Fujita, *et al.*, 2002). Although the global prevalence of *H. pylori* infection has significantly decreased from 58.2% in the 1980s, there is still 43.1% population around the world plagued by the bacteria (Li *et al.*, 2023). It has already been proven that chronic infection with cagA$^+$ *H. pylori* strongly correlates with the development of gastric cancer. When injected into an epithelial cell, CagA protein binds with Src homology 2-Containing Protein Tyrosine Phosphatase 2 (SHP-2, encoded by PTPN*11*) and causes its degeneration, leading to morphological changes in the cell (Higashi, Tsutsumi, Muto, *et al.*, 2002). According to previous research, at both N and C- terminal of SHP-2, it has two Src homology 2 (SH2) domains which are able to bind specific peptide motifs containing phosphorylated tyrosine (pY). If the phosphate group was removed, the binding strength would drop significantly (Asmamaw *et al.*, 2022). Glu-Pro-Ile-Tyr-Ala (EPIYA) sequences within CagA are binding sites of the SH2 domain, which are classified as EPIYA-C and EPIYA-D by their neighboring residues. Respectively found in *H. pylori* isolates from Western and East Asian countries, EPIYA-C has a much lower affinity than EPIYA-D with SHP-2, preliminarily explaining the relatively high prevalence of gastric carcinoma in East Asia (Higashi, Tsutsumi, Fujita, *et al.*, 2002). However, the previous experiments only used the N-terminal SH2 domain, and the properties of the C-terminal SH2 domain are not much studied.

As computing power elevating and stimulatory techniques advance, in silico pre-experiment has become more popular these days, for which is less time-consuming and less costly. Molecular dynamics simulations are techniques to construct motions of biological molecules at atomic resolution (Filipe and Loura, 2022). Interactions between two proteins can be simulated and analyzed by molecular dynamics, yet it is a simplified model after all with uncertain accuracy.

In this case, I wanted to validate the usability of molecular dynamics for protein interaction analysis and explore the binding of the C-terminal SH2 domain with EPIYA. By first simulating the interactions between the N-terminal SH2 domain and phosphorylated/unphosphorylated EPIYA, N-terminal SH2 domain and EPIYA-C/D, the reliability of the approach was tested. Then, the properties of the C-terminal SH2 domain were explored through molecular dynamics.

Materials & Methods

Two protein structures from Protein Data Bank (PDB) were employed for the simulations: ID 5X94, crystal structure of SHP2_SH2-CagA EPIYA_D peptide complex; ID 5X7B, crystal structure of SHP2_SH2-CagA EPIYA_C peptide complex. Two downloaded .pdb files were firstly modified with PDB Reader & Manipulator of CHARMM-GUI (https://www.charmm-gui.org/?doc=input/pdbreader) to remove all engineered amino acid residues and add phosphate groups to EPIYA sequences. To save time and computing resources, two SHP-2 proteins were both manually cleaved into halves between 105 ALA and 106 ASP with a text editor, each had a SH2 domain with an EPIYA 13-mer peptide. For molecular dynamic simulations, I used GROMACS, which is a well-recognized package designed for biochemical molecules like proteins, lipids, and nucleic acids. Following a tutorial from Justin A. Lemkul at http://www.mdtutorials.com/gmx/index.html, energy minimization, equilibration, and molecular dynamics were conducted with 5 combination groups: N-SH2 + EPIpYA-D, N-SH2 + EPIYA-D, C-SH2 + EPIpYA-D, N-SH2 + EPIpYA-C, C-SH2 + EPIpYA-C. CHARMM36 force field (version July 2022) was applied in the simulation step. Root Mean Square Deviation (RMSD) and Radius of Gyration (RG) over time of each simulation were collected. Three groups with EPIpYA-D and EPIYA-D had 4 repeated simulations of 4 ns, two groups with EPIpYA-C had 3 repeated simulations of 10 ns, since the latter ones' RMSD reached stable more slowly. Data analysis and visualization were conducted by R version 4.3.2 and R package ggplot2 version 3.4.4. One-tailed

Welch Two Sample t-test was applied for comparisons between groups.

Results

RMSD is the most commonly used measure for describing protein structure dynamics. In this project, RMSD of each protein represents the overall displacement of corresponding atoms between the protein structures before and after the molecular dynamic simulation. A high RMSD value means big movements during the simulation, reflecting instability. RG shows the compactness of a protein structure, its low value and minor fluctuation are evidence of stability of the structure. RMSD and RG can only be used for general comparisons rather than precise calculations. The mean values of RMSD and RG from all simulations within a group at each timepoint were calculated for comparison, coefficient of dispersion of RG was also determined to reflect fluctuation.

As we know before, the binding affinity between the SH2 domain and unphosphorylated EPIYA is much lower than the affinity between the SH2 domain and EPIYA with phosphorylated tyrosine (Asmamaw *et al.*, 2022), thus, the latter combination is supposed to be relatively unstable. According to my simulations of the N-SH2 domain, the RMSD values of both groups had some overlapping regions in the first 1 ns; as the two lines become steady, RMSDs of the group with phosphate was significantly lower than the group without phosphate (Fig. 1A). Besides, it has been proven that EPIpYA-D binds to SH2 domain more tightly than EPIpYA-C (Higashi, Tsutsumi, Fujita, *et al.*, 2002). My simulations reflected the same trend: RMSD of the EPIpYA-C groups were significantly higher than the EPIpYA-D groups interacting with both N- and C-SH2 domains (Fig. 1B and 1C). The RMSD difference between EPIpYA-D and EPIpYA-C increases over time with both SH2 domains. The results matched the literature perfectly, which showed a good usability of the molecular dynamic simulation of GROMACS.

Since the binding affinity diversity between N-SH2 and C-SH2 domains of SHP-2 is not well understood, I compared RMSD of them interacting with either EPIpYA-D and EPIpYA-C. A big gap between two RMSD lines was shown on the plots (Fig. 1D and 1E). As expected, the statistical tests gave the same results that the N-SH2 domain had lower RMSD, reflecting higher stability binding with either EPIpYA-D or EPIpYA-C.

Opposed to RMSD, RG didn’t provide good matches to literature. As for the coefficient of dispersion, although the differences between groups with and without phosphate and between groups of two SH2 domains were detected, there was no significant difference between EPIpYA-D and EPIpYA-C (Fig. 2). The situation was the same for RG mean, which is not presented in the figure.

Conclusion and Discussion

Based on my simulations, the analysis of the collected RMSD data revealed that phosphorylated EPIYA and EPIYA-D exhibit greater stability in their interactions with the SH2 domains compared to dephosphorylated EPIYA and EPIYA-C respectively. The successful reproduction of previous experimental results affirmed the reliability of molecular dynamics. Furthermore, by comparing the stability of the interactions between the N and C-terminal SH2 domains with EPIpYA, it was observed that the N-terminal SH2 domain potentially possesses a higher relative affinity for the CagA protein than the C-terminal SH2 domain.

However, the analysis of RG did not reflect the same results. While RG is not as widely recognized as RMSD, its values and the extent of their fluctuations over time can still serve as indicators of the binding stability between the peptide chains. To save time and computational resources, I manually truncated the SHP-2 polypeptide between residues 105 ALA and 106 ASP, separating the N-terminal and C-terminal SH2 domains for subsequent molecular dynamics simulations, rather than using the full-length protein. Consequently, the free movement of the truncated polypeptide ends likely affected the RMSD and RG values and their fluctuations. Additionally, the truncation site in the SHP-2 was chosen based on its 3D structure without testing, which might have impacted the accuracy of the simulations. Future studies could consider using the full-length SHP-2 protein for molecular dynamics simulations with the EPIYA peptide binding to either the N or C-terminal SH2 domains.

In this experiment, the groups involving interactions with EPIYA-D underwent 4 repeat simulations and the groups involving EPIYA-C underwent 3, which are relatively few. Future research should consider increasing the number and duration of simulations for each group, combining data from multiple runs to minimize systematic errors. Additionally, this experiment utilized only the GROMACS software for molecular dynamics simulations. Future explorations could employ other molecular dynamics software such as LAMMPS and VASP for comparative analysis.

The RMSD results obtained from the molecular dynamics simulations only represent the stability of the SH2 domain binding to EPIYA and do not directly reflect the binding affinity between them. In subsequent research, based on the results of these simulations, surface plasmon resonance (SPR) analysis could be conducted to determine whether the N-terminal SH2 domain of the SHP-2 protein possesses a higher affinity for the CagA protein than the C-terminal SH2 domain. This would assist in studying the impact mechanism of *H. pylori's* CagA protein on the SHP-2 protein and its downstream pathways.

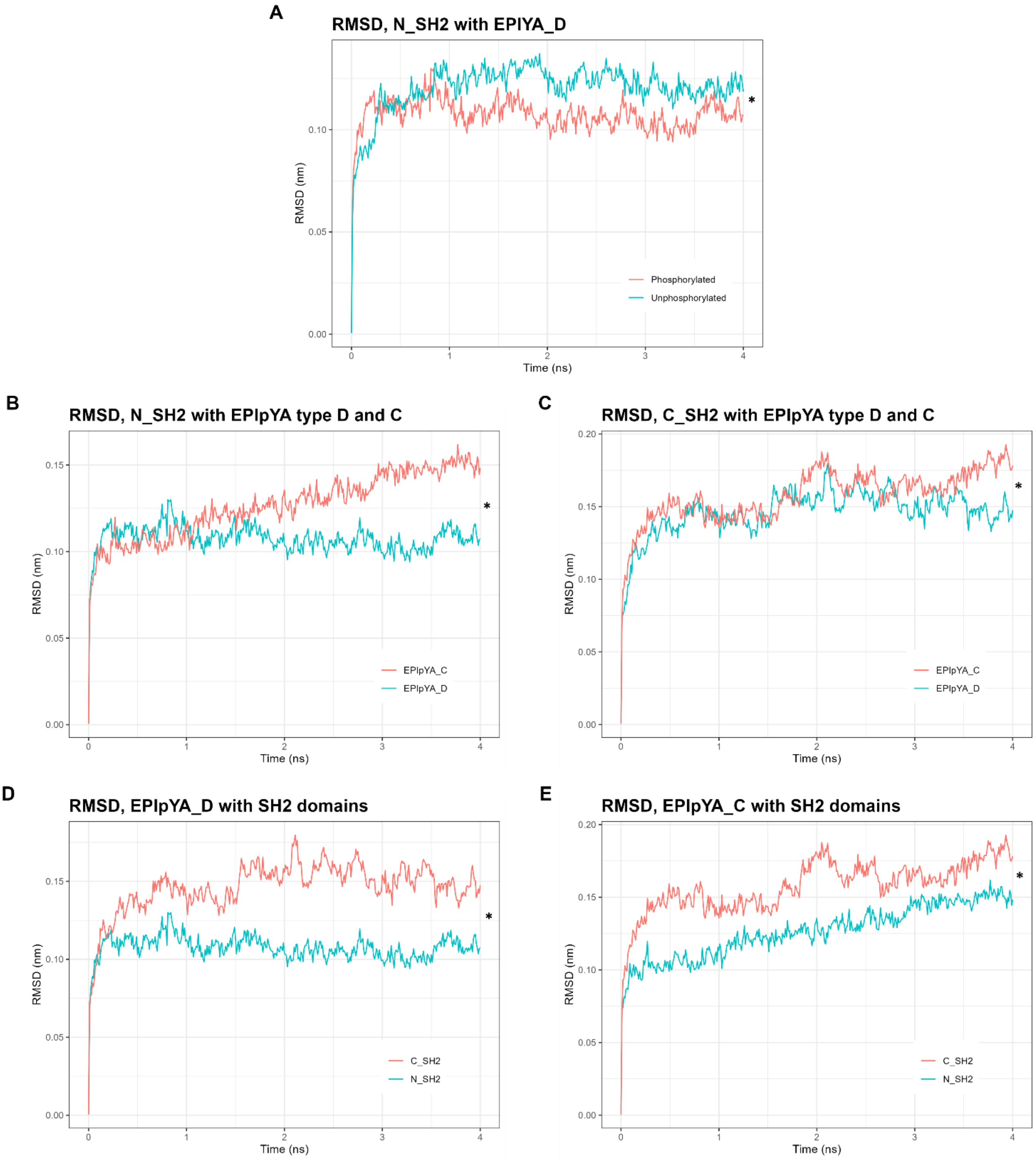


Fig. 1 Comparisons of Root Mean Square Deviation Means: (A) N-SH2 binding with phosphorylated/unphosphorylated EPIYA-D; **(B, C)** Comparing the influence of EPIpYA type change on the binding stability of N-SH2 and C-SH2; **(D, E)** Comparing the influence of SH2 position change on the binding stability of EPIpYA-D and EPIpYA-C. Welch Two Sample t-test; *: p-value < 2.2e-16.

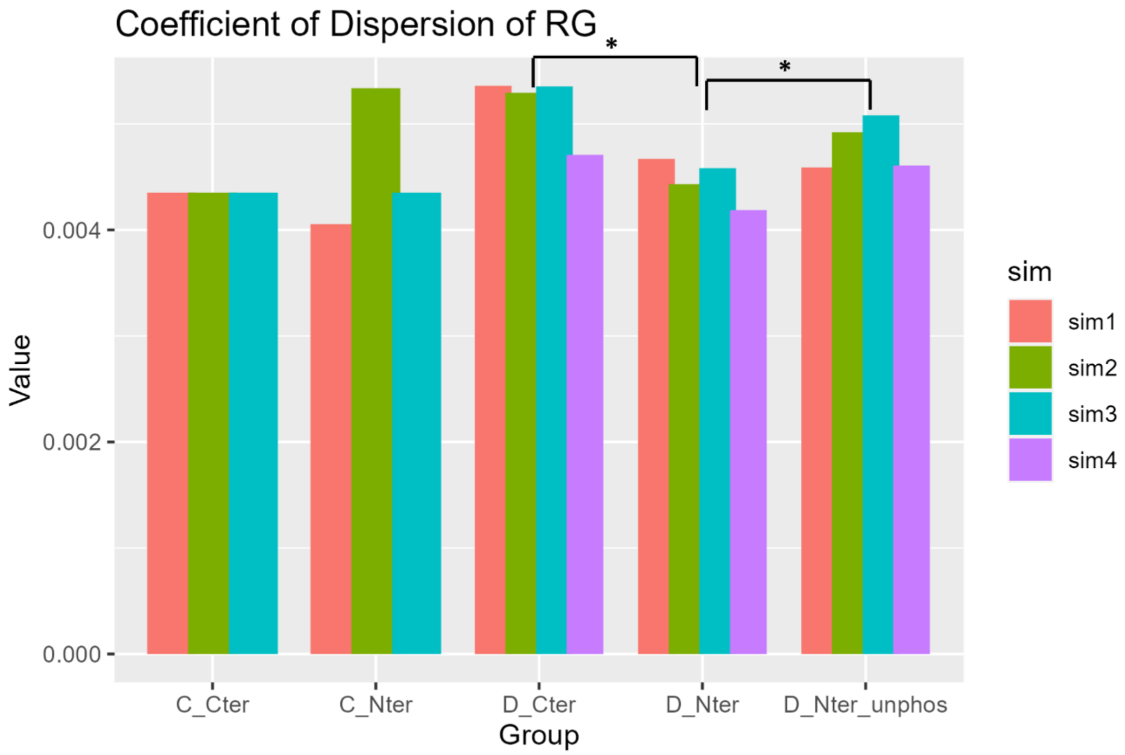


Fig. 2 Coefficient of Dispersion - Radius of Gyration: The coefficient of dispersion of RG presents significant differences in the phosphate test and the SH2-position test using EPIYA-D. The positional influence of SH2 binding with EPIYA-C and the affinity difference between EPIYA-D and EPIYA-C are not detected by RG. C: EPIYA-C; D: EPIYA-D; Nter: N-terminal-SH2; Cter: C-terminal-SH2; Welch Two Sample t-test; *: p-value < 0.05.

Acknowledgement

I appreciate Dr. Hugo Sámano-Sánchez for providing research advice, high-performance computing environment, and code scripts. I'm also grateful to Shengzhi Chen, Zhanyi Lin, and Chen Cui for their discussions with me about troubleshooting.

Reference

Asmamaw, M.D. *et al.* (2022) 'A comprehensive review of SHP2 and its role in cancer', *Cellular Oncology*, 45(5), pp. 729–753. Available at: https://doi.org/10.1007/s13402-022-00698-1.

Filipe, H.A.L. and Loura, L.M.S. (2022) 'Molecular Dynamics Simulations: Advances and Applications', *Molecules*, 27(7), p. 2105. Available at: https://doi.org/10.3390/molecules27072105.

Higashi, H., Tsutsumi, R., Fujita, A., *et al.* (2002) ‘Biological activity of the *Helicobacter pylori* virulence factor CagA is determined by variation in the tyrosine phosphorylation sites’, *Proceedings of the National Academy of Sciences*, 99(22), pp. 14428–14433. Available at: https://doi.org/10.1073/pnas.222375399.

Higashi, H., Tsutsumi, R., Muto, S., *et al.* (2002) ‘SHP-2 Tyrosine Phosphatase as an Intracellular Target of *Helicobacter pylori* CagA Protein’, *Science*, 295(5555), pp. 683–686. Available at: https://doi.org/10.1126/science.1067147.

Li, Y. *et al.* (2023) ‘Global prevalence of *Helicobacter pylori* infection between 1980 and 2022: a systematic review and meta-analysis’, *The Lancet Gastroenterology & Hepatology*, 8(6), pp. 553–564. Available at: https://doi.org/10.1016/S2468-1253(23)00070-5.

Codes

GROMACS

```
# Generate topology file, position restraint file and the post-processed structure file

gmx_mpi pdb2gmx -f SHP2_SH2_CagA_EPIYA_C_Nter.pdb -o SHP2_SH2_CagA_EPIYA_C_Nter_processed.gro -ff charmm36-jul2022 -ignh -water spce


# Define the box

gmx_mpi editconf -f SHP2_SH2_CagA_EPIYA_C_Nter_processed.gro -o SHP2_SH2_CagA_EPIYA_C_Nter_newbox.gro -c -d 1.0 -bt cubic


# Fill it with solvent (water)

gmx_mpi solvate -cp SHP2_SH2_CagA_EPIYA_C_Nter_newbox.gro -cs spc216.gro -o SHP2_SH2_CagA_EPIYA_C_Nter_solv.gro -p topol.top


# Adding ions (Cl-1)

gmx_mpi grompp -f ions.mdp -c SHP2_SH2_CagA_EPIYA_C_Nter_solv.gro -p topol.top -o ions.tpr


# genion

gmx_mpi genion -s ions.tpr -o SHP2_SH2_CagA_EPIYA_C_Nter_solv_ions.gro -p topol.top -pname NA -nname CL -neutral    # SOL


# Energy minimization

gmx_mpi grompp -f minim.mdp -c SHP2_SH2_CagA_EPIYA_C_Nter_solv_ions.gro -p topol.top -o em.tpr -maxwarn 1

gmx_mpi mdrun -v -deffnm em -ntomp 16 -nb gpu

# Energy potential should be in the range    -10e5:-10e6


# See potential convergence

gmx_mpi energy -f em.edr -o potential.xvg
```

```
#   NVT Equilibration
gmx_mpi grompp -f nvt.mdp -c em.gro -r em.gro -p topol.top -o nvt.tpr
gmx_mpi mdrun -deffnm nvt -ntomp 16 -nb gpu
gmx_mpi energy -f nvt.edr -o temperature.xvg
```

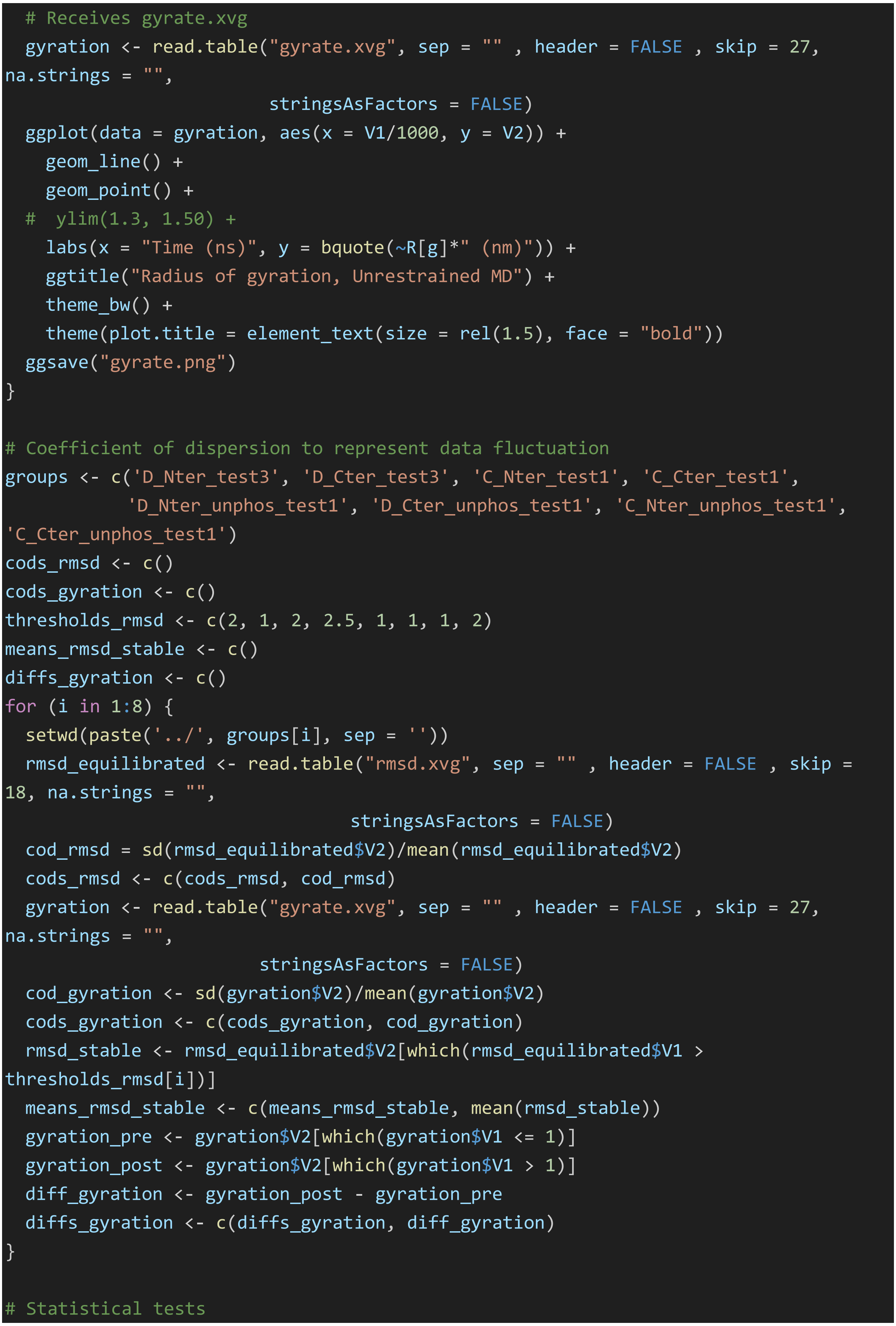
```
# NPT Equilibration
gmx_mpi grompp -f npt.mdp -c nvt.gro -r nvt.gro -t nvt.cpt -p topol.top -o npt.tpr
gmx_mpi mdrun -deffnm npt -ntomp 16 -nb gpu
gmx_mpi energy -f npt.edr -o pressure.xvg
gmx_mpi energy -f npt.edr -o density.xvg
```

```
# Molecular Dynamics
gmx_mpi grompp -f md.mdp -c npt.gro -t npt.cpt -p topol.top -o md_0_4.tpr
gmx_mpi mdrun -deffnm md_0_4 -ntomp 16 -nb gpu
```

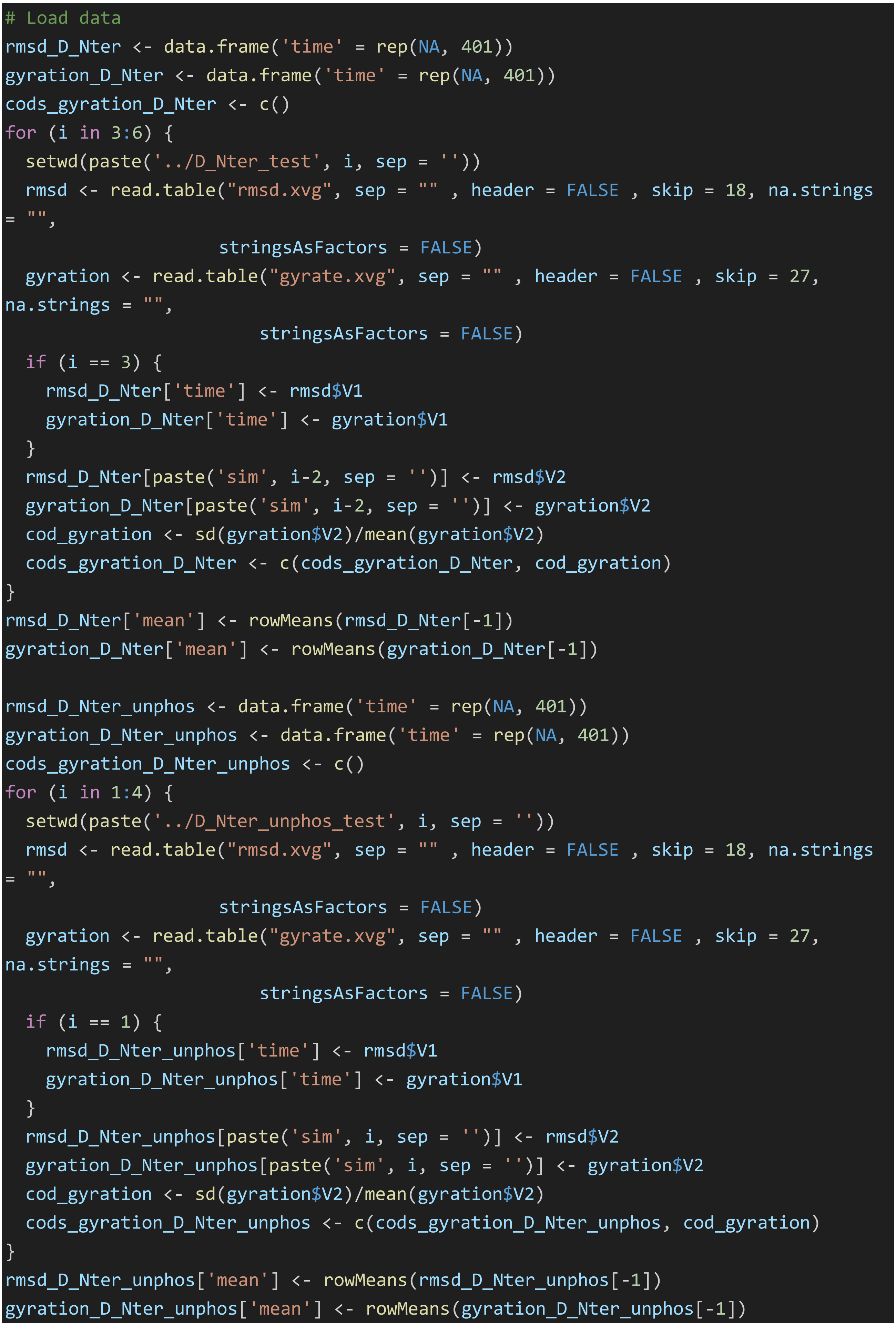
```
# Analysis
gmx_mpi trjconv -s md_0_4.tpr -f md_0_4.xtc -o md_0_4_noPBC.xtc -pbc mol -center    # 'Protein' to be centered, 'System' for output
gmx_mpi rms -s md_0_4.tpr -f md_0_4_noPBC.xtc -o rmsd.xvg -tu ns    # 'Backbone' for both the least-squares fit and the group for RMSD calculation
gmx_mpi rms -s em.tpr -f md_0_4_noPBC.xtc -o rmsd_xtal.xvg -tu ns    # 'Backbone' for both the least-squares fit and the group for RMSD calculation
gmx_mpi gyrate -s md_0_4.tpr -f md_0_4_noPBC.xtc -o gyrate.xvg    # 'Protein' for gyrate radius calculation
```

R

```r
# Plots to evaluate the GROMACS pipeline
library(ggplot2)
setwd("D:/DeskTop/Courses/CBSB3/MiniProject")
setwd("./D_Nter_test3")

# Energy minimization
# Receives potential.xvg
potential <- read.table("potential.xvg", sep = "" , header = FALSE , skip = 24, na.strings = "",
                       stringsAsFactors = FALSE)
ggplot(data = potential, aes(x = V1, y = V2)) +
  geom_line() +
  geom_point() +
  ylim(min(potential$V2), 0) +
  labs(x = "Energy Minimization Step", y = bquote("Potential Energy (kJ "*~mol^-1*')')) +
  ggtitle("Energy Minimization, Steepest Descent") +
  theme_bw() +
  theme(plot.title = element_text(size = rel(1.5), face = "bold"))
ggsave("potential.png")

# Temperature equilibration
# Receives temperature.xvg
temperature <- read.table("temperature.xvg", sep = "" , header = FALSE , skip = 24,
na.strings = "",
                         stringsAsFactors = FALSE)
temperature$average10ps <- NA
temperature$average10ps[10:nrow(temperature)] <- sapply(10:nrow(temperature),
function(x){mean(temperature$V2[(x-9):x])})
ggplot(data = temperature, aes(x = V1, y = V2)) +
  geom_line() +
  geom_point() +
  geom_line(aes(y = average10ps, col = "Running average 10 ps")) +
  labs(x = "Time (ps)", y = "Temperature (K)") +
  ggtitle("Temperature, NVT equilibration") +
  theme_bw() +
  theme(legend.position = c(0.80, 0.9),
        legend.title = element_blank(),
        plot.title = element_text(size = rel(1.5), face = "bold"))
ggsave("temperature.png")

# Pressure equilibration
# Receives pressure.xvg
pressure <- read.table("pressure.xvg", sep = "" , header = FALSE , skip = 24,
na.strings = "",
                      stringsAsFactors = FALSE)
pressure$average10ps <- NA
pressure$average10ps[10:nrow(pressure)] <- sapply(10:nrow(pressure),
function(x){mean(pressure$V2[(x-9):x])})
ggplot(data = pressure, aes(x = V1, y = V2)) +
  geom_line() +
  geom_point() +
  geom_line(aes(y = average10ps, col = "Running average 10 ps")) +
  labs(x = "Time (ps)", y = "Pressure (bar)") +
  ggtitle("Pressure, NPT equilibration") +
  theme_bw() +
  theme(legend.position = c(0.80, 0.9),
        legend.title = element_blank(),
        plot.title = element_text(size = rel(1.5), face = "bold"))
ggsave("pressure.png")

# Density equilibration
# Receives density.xvg
density <- read.table("density.xvg", sep = "" , header = FALSE , skip = 24,
na.strings = "",
                     stringsAsFactors = FALSE)
density$average10ps <- NA
density$average10ps[10:nrow(density)] <- sapply(10:nrow(density),
function(x){mean(density$V2[(x-9):x])})
ggplot(data = density, aes(x = V1, y = V2)) +
  geom_line() +
  geom_point() +
  geom_line(aes(y = average10ps, col = "Running average 10 ps")) +
  labs(x = "Time (ps)", y = bquote("Density (kg "*~m^-3*')')) +
  ggtitle("Density, NPT equilibration") +
  theme_bw() +
  theme(legend.position = c(0.80, 0.2),
        legend.title = element_blank(),
        plot.title = element_text(size = rel(1.5), face = "bold"))
ggsave("density.png")

groups <- c('D_Nter_test6', 'D_Nter_unphos_test4', 'D_Cter_test6', 'C_Nter_test4',
'C_Cter_test4')
for (group in groups) {
  setwd(paste('../', group, sep = ''))
  # RMSD, backbone
  # Receives rmsd.xvg and rmsd_xtal.xvg
  rmsd_equilibrated <- read.table("rmsd.xvg", sep = "" , header = FALSE , skip =
18, na.strings = "",
                       stringsAsFactors = FALSE)
  rmsd_xtal <- read.table("rmsd_xtal.xvg", sep = "" , header = FALSE , skip = 18,
na.strings = "",
                                 stringsAsFactors = FALSE)
  rmsd <- rmsd_equilibrated
  names(rmsd) <- c("time", "equilibrated")
  rmsd$xtal <- rmsd_xtal$V2
  ggplot(data = rmsd, aes(x = time)) +
    geom_line(aes(y = equilibrated, col = "Ref: Equilibrated")) +
    geom_line(aes(y = xtal, col = "Ref: Original") ) +
    labs(x = "Time (ns)", y = "RMSD (nm)") +
    ggtitle("RMSD, backbone") +
    theme_bw() +
    theme(legend.position = c(0.80, 0.2),
          legend.title = element_blank(),
          plot.title = element_text(size = rel(1.5), face = "bold"))
  ggsave("RMSD.png")

  # Radius of gyration
```

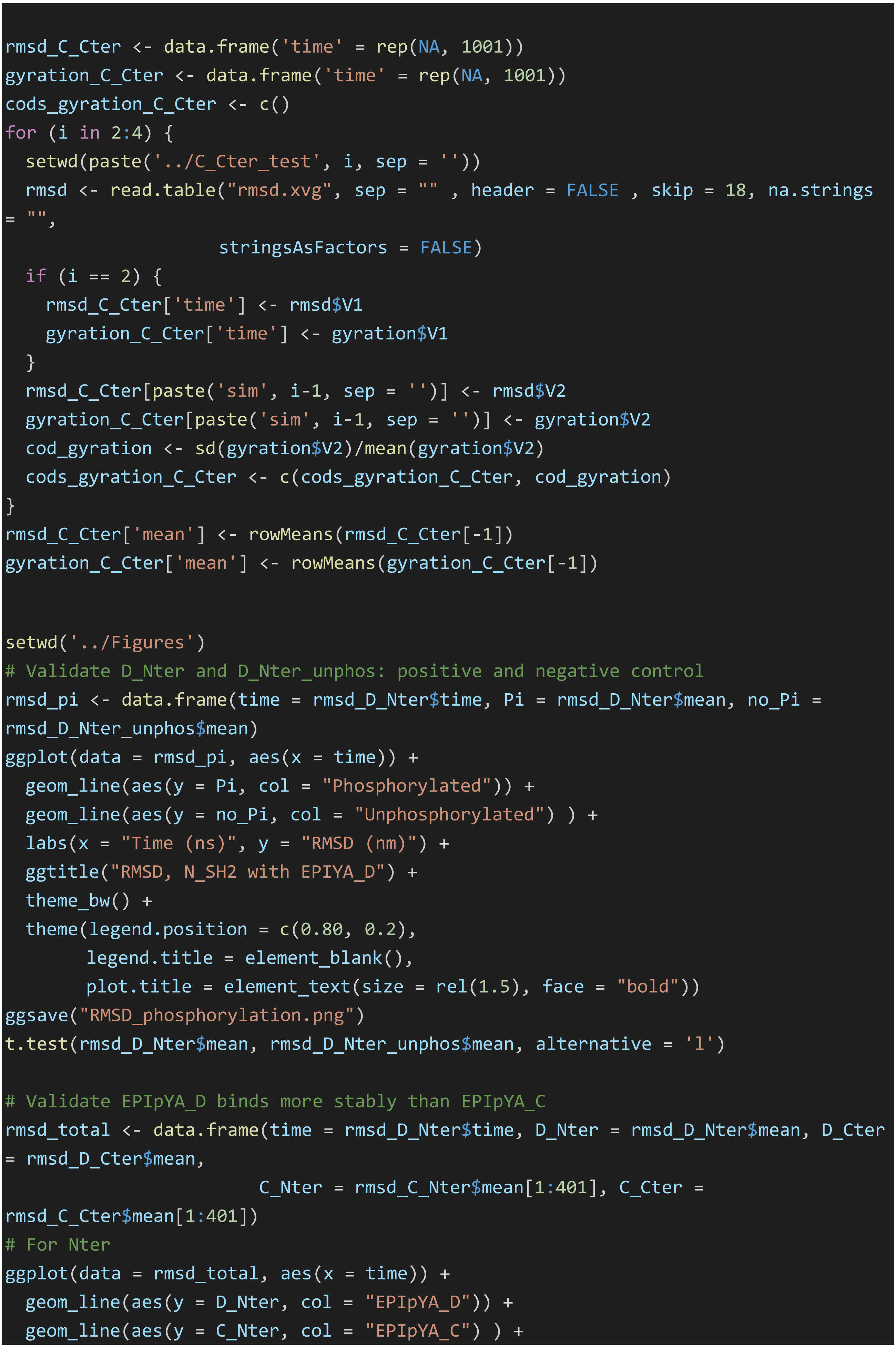


```
  # Receives gyrate.xvg
  gyration <- read.table("gyrate.xvg", sep = "" , header = FALSE , skip = 27,
na.strings = "",
                         stringsAsFactors = FALSE)
  ggplot(data = gyration, aes(x = V1/1000, y = V2)) +
    geom_line() +
    geom_point() +
  #  ylim(1.3, 1.50) +
    labs(x = "Time (ns)", y = bquote(~R[g]*" (nm)")) +
    ggtitle("Radius of gyration, Unrestrained MD") +
    theme_bw() +
    theme(plot.title = element_text(size = rel(1.5), face = "bold"))
  ggsave("gyrate.png")
}

# Coefficient of dispersion to represent data fluctuation
groups <- c('D_Nter_test3', 'D_Cter_test3', 'C_Nter_test1', 'C_Cter_test1',
            'D_Nter_unphos_test1', 'D_Cter_unphos_test1', 'C_Nter_unphos_test1',
'C_Cter_unphos_test1')
cods_rmsd <- c()
cods_gyration <- c()
thresholds_rmsd <- c(2, 1, 2, 2.5, 1, 1, 1, 2)
means_rmsd_stable <- c()
diffs_gyration <- c()
for (i in 1:8) {
  setwd(paste('../', groups[i], sep = ''))
  rmsd_equilibrated <- read.table("rmsd.xvg", sep = "" , header = FALSE , skip =
18, na.strings = "",
                                 stringsAsFactors = FALSE)
  cod_rmsd = sd(rmsd_equilibrated$V2)/mean(rmsd_equilibrated$V2)
  cods_rmsd <- c(cods_rmsd, cod_rmsd)
  gyration <- read.table("gyrate.xvg", sep = "" , header = FALSE , skip = 27,
na.strings = "",
                        stringsAsFactors = FALSE)
  cod_gyration <- sd(gyration$V2)/mean(gyration$V2)
  cods_gyration <- c(cods_gyration, cod_gyration)
  rmsd_stable <- rmsd_equilibrated$V2[which(rmsd_equilibrated$V1 >
thresholds_rmsd[i])]
  means_rmsd_stable <- c(means_rmsd_stable, mean(rmsd_stable))
  gyration_pre <- gyration$V2[which(gyration$V1 <= 1)]
  gyration_post <- gyration$V2[which(gyration$V1 > 1)]
  diff_gyration <- gyration_post - gyration_pre
  diffs_gyration <- c(diffs_gyration, diff_gyration)
}

# Statistical tests
```



```r
# Load data
rmsd_D_Nter <- data.frame('time' = rep(NA, 401))
gyration_D_Nter <- data.frame('time' = rep(NA, 401))
cods_gyration_D_Nter <- c()
for (i in 3:6) {
  setwd(paste('../D_Nter_test', i, sep = ''))
  rmsd <- read.table("rmsd.xvg", sep = "" , header = FALSE , skip = 18, na.strings
= "",
                     stringsAsFactors = FALSE)
  gyration <- read.table("gyrate.xvg", sep = "" , header = FALSE , skip = 27,
na.strings = "",
                         stringsAsFactors = FALSE)
  if (i == 3) {
    rmsd_D_Nter['time'] <- rmsd$V1
    gyration_D_Nter['time'] <- gyration$V1
  }
  rmsd_D_Nter[paste('sim', i-2, sep = '')] <- rmsd$V2
  gyration_D_Nter[paste('sim', i-2, sep = '')] <- gyration$V2
  cod_gyration <- sd(gyration$V2)/mean(gyration$V2)
  cods_gyration_D_Nter <- c(cods_gyration_D_Nter, cod_gyration)
}
rmsd_D_Nter['mean'] <- rowMeans(rmsd_D_Nter[-1])
gyration_D_Nter['mean'] <- rowMeans(gyration_D_Nter[-1])

rmsd_D_Nter_unphos <- data.frame('time' = rep(NA, 401))
gyration_D_Nter_unphos <- data.frame('time' = rep(NA, 401))
cods_gyration_D_Nter_unphos <- c()
for (i in 1:4) {
  setwd(paste('../D_Nter_unphos_test', i, sep = ''))
  rmsd <- read.table("rmsd.xvg", sep = "" , header = FALSE , skip = 18, na.strings
= "",
                     stringsAsFactors = FALSE)
  gyration <- read.table("gyrate.xvg", sep = "" , header = FALSE , skip = 27,
na.strings = "",
                         stringsAsFactors = FALSE)
  if (i == 1) {
    rmsd_D_Nter_unphos['time'] <- rmsd$V1
    gyration_D_Nter_unphos['time'] <- gyration$V1
  }
  rmsd_D_Nter_unphos[paste('sim', i, sep = '')] <- rmsd$V2
  gyration_D_Nter_unphos[paste('sim', i, sep = '')] <- gyration$V2
  cod_gyration <- sd(gyration$V2)/mean(gyration$V2)
  cods_gyration_D_Nter_unphos <- c(cods_gyration_D_Nter_unphos, cod_gyration)
}
rmsd_D_Nter_unphos['mean'] <- rowMeans(rmsd_D_Nter_unphos[-1])
gyration_D_Nter_unphos['mean'] <- rowMeans(gyration_D_Nter_unphos[-1])
```

```
rmsd_D_Cter <- data.frame('time' = rep(NA, 401))
gyration_D_Cter <- data.frame('time' = rep(NA, 401))
cods_gyration_D_Cter <- c()
for (i in 3:6) {
  setwd(paste('../D_Cter_test', i, sep = ''))
  rmsd <- read.table("rmsd.xvg", sep = "" , header = FALSE , skip = 18, na.strings
= "",
                     stringsAsFactors = FALSE)
  gyration <- read.table("gyrate.xvg", sep = "" , header = FALSE , skip = 27,
na.strings = "",
                         stringsAsFactors = FALSE)
  if (i == 3) {
    rmsd_D_Cter['time'] <- rmsd$V1
    gyration_D_Cter['time'] <- gyration$V1
  }
  rmsd_D_Cter[paste('sim', i-2, sep = '')] <- rmsd$V2
  gyration_D_Cter[paste('sim', i-2, sep = '')] <- gyration$V2
  cod_gyration <- sd(gyration$V2)/mean(gyration$V2)
  cods_gyration_D_Cter <- c(cods_gyration_D_Cter, cod_gyration)
}
rmsd_D_Cter['mean'] <- rowMeans(rmsd_D_Cter[-1])
gyration_D_Cter['mean'] <- rowMeans(gyration_D_Cter[-1])

rmsd_C_Nter <- data.frame('time' = rep(NA, 1001))
gyration_C_Nter <- data.frame('time' = rep(NA, 1001))
cods_gyration_C_Nter <- c()
for (i in 2:4) {
  setwd(paste('../C_Nter_test', i, sep = ''))
  rmsd <- read.table("rmsd.xvg", sep = "" , header = FALSE , skip = 18, na.strings
= "",
                     stringsAsFactors = FALSE)
  gyration <- read.table("gyrate.xvg", sep = "" , header = FALSE , skip = 27,
na.strings = "",
                         stringsAsFactors = FALSE)
  if (i == 2) {
    rmsd_C_Nter['time'] <- rmsd$V1
    gyration_C_Nter['time'] <- gyration$V1
  }
  rmsd_C_Nter[paste('sim', i-1, sep = '')] <- rmsd$V2
  gyration_C_Nter[paste('sim', i-1, sep = '')] <- gyration$V2
  cod_gyration <- sd(gyration$V2)/mean(gyration$V2)
  cods_gyration_C_Nter <- c(cods_gyration_C_Nter, cod_gyration)
}
rmsd_C_Nter['mean'] <- rowMeans(rmsd_C_Nter[-1])
gyration_C_Nter['mean'] <- rowMeans(gyration_C_Nter[-1])
```



```
rmsd_C_Cter <- data.frame('time' = rep(NA, 1001))
gyration_C_Cter <- data.frame('time' = rep(NA, 1001))
cods_gyration_C_Cter <- c()
for (i in 2:4) {
  setwd(paste('../C_Cter_test', i, sep = ''))
  rmsd <- read.table("rmsd.xvg", sep = "" , header = FALSE , skip = 18, na.strings
= "",
                     stringsAsFactors = FALSE)
  if (i == 2) {
    rmsd_C_Cter['time'] <- rmsd$V1
    gyration_C_Cter['time'] <- gyration$V1
  }
  rmsd_C_Cter[paste('sim', i-1, sep = '')] <- rmsd$V2
  gyration_C_Cter[paste('sim', i-1, sep = '')] <- gyration$V2
  cod_gyration <- sd(gyration$V2)/mean(gyration$V2)
  cods_gyration_C_Cter <- c(cods_gyration_C_Cter, cod_gyration)
}
rmsd_C_Cter['mean'] <- rowMeans(rmsd_C_Cter[-1])
gyration_C_Cter['mean'] <- rowMeans(gyration_C_Cter[-1])


setwd('../Figures')
# Validate D_Nter and D_Nter_unphos: positive and negative control
rmsd_pi <- data.frame(time = rmsd_D_Nter$time, Pi = rmsd_D_Nter$mean, no_Pi =
rmsd_D_Nter_unphos$mean)
ggplot(data = rmsd_pi, aes(x = time)) +
  geom_line(aes(y = Pi, col = "Phosphorylated")) +
  geom_line(aes(y = no_Pi, col = "Unphosphorylated") ) +
  labs(x = "Time (ns)", y = "RMSD (nm)") +
  ggtitle("RMSD, N_SH2 with EPIYA_D") +
  theme_bw() +
  theme(legend.position = c(0.80, 0.2),
        legend.title = element_blank(),
        plot.title = element_text(size = rel(1.5), face = "bold"))
ggsave("RMSD_phosphorylation.png")
t.test(rmsd_D_Nter$mean, rmsd_D_Nter_unphos$mean, alternative = 'l')

# Validate EPIpYA_D binds more stably than EPIpYA_C
rmsd_total <- data.frame(time = rmsd_D_Nter$time, D_Nter = rmsd_D_Nter$mean, D_Cter
= rmsd_D_Cter$mean,
                         C_Nter = rmsd_C_Nter$mean[1:401], C_Cter =
rmsd_C_Cter$mean[1:401])
# For Nter
ggplot(data = rmsd_total, aes(x = time)) +
  geom_line(aes(y = D_Nter, col = "EPIpYA_D")) +
  geom_line(aes(y = C_Nter, col = "EPIpYA_C") ) +
```

```r
  labs(x = "Time (ns)", y = "RMSD (nm)") +
  ggtitle("RMSD, N_SH2 with EPIpYA type D and C") +
  theme_bw() +
  theme(legend.position = c(0.80, 0.2),
        legend.title = element_blank(),
        plot.title = element_text(size = rel(1.5), face = "bold"))
ggsave("RMSD_CagA_Nter.png")
t.test(rmsd_D_Nter$mean, rmsd_C_Nter$mean, alternative = 'l')
# For Cter
ggplot(data = rmsd_total, aes(x = time)) +
  geom_line(aes(y = D_Cter, col = "EPIpYA_D")) +
  geom_line(aes(y = C_Cter, col = "EPIpYA_C") ) +
  labs(x = "Time (ns)", y = "RMSD (nm)") +
  ggtitle("RMSD, C_SH2 with EPIpYA type D and C") +
  theme_bw() +
  theme(legend.position = c(0.80, 0.2),
        legend.title = element_blank(),
        plot.title = element_text(size = rel(1.5), face = "bold"))
ggsave("RMSD_CagA_Cter.png")
t.test(rmsd_D_Cter$mean, rmsd_C_Cter$mean, alternative = 'l')

# Explore Nter is more stable than Cter with EPIpYA
# For EPIpYA_D
ggplot(data = rmsd_total, aes(x = time)) +
  geom_line(aes(y = D_Nter, col = "N_SH2")) +
  geom_line(aes(y = D_Cter, col = "C_SH2") ) +
  labs(x = "Time (ns)", y = "RMSD (nm)") +
  ggtitle("RMSD, EPIpYA_D with SH2 domains") +
  theme_bw() +
  theme(legend.position = c(0.80, 0.2),
        legend.title = element_blank(),
        plot.title = element_text(size = rel(1.5), face = "bold"))
ggsave("RMSD_SH2_D.png")
t.test(rmsd_D_Nter$mean, rmsd_D_Cter$mean, alternative = 'l')
# For EPIpYA_C
ggplot(data = rmsd_total, aes(x = time)) +
  geom_line(aes(y = C_Nter, col = "N_SH2")) +
  geom_line(aes(y = C_Cter, col = "C_SH2") ) +
  labs(x = "Time (ns)", y = "RMSD (nm)") +
  ggtitle("RMSD, EPIpYA_C with SH2 domains") +
  theme_bw() +
  theme(legend.position = c(0.80, 0.2),
        legend.title = element_blank(),
        plot.title = element_text(size = rel(1.5), face = "bold"))
ggsave("RMSD_SH2_C.png")
t.test(rmsd_C_Nter$mean, rmsd_C_Cter$mean, alternative = 'l')
```

```
# Gyration
# means
t.test(gyration_D_Nter$mean, gyration_D_Nter_unphos$mean, alternative = 'l')  # yes
t.test(gyration_D_Nter$mean, gyration_C_Nter$mean, alternative = 'l')  # no
t.test(gyration_D_Cter$mean, gyration_C_Cter$mean, alternative = 'l')  # no
t.test(gyration_D_Nter$mean, gyration_D_Cter$mean, alternative = 'l')  # yes
t.test(gyration_C_Nter$mean, gyration_C_Cter$mean, alternative = 'l')  # no
# cods
cods_gyration <- data.frame('group' = c(rep('D_Nter', 4), rep('D_Nter_unphos', 4),
rep('D_Cter', 4), rep('C_Nter', 3), rep('C_Cter', 3)),
                            'sim' = c(rep(c('sim1', 'sim2', 'sim3', 'sim4'), 3),
rep(c('sim1', 'sim2', 'sim3'), 2)),
                            'value' = c(cods_gyration_D_Nter,
cods_gyration_D_Nter_unphos, cods_gyration_D_Cter, cods_gyration_C_Nter,
cods_gyration_C_Cter))
ggplot(data = cods_gyration, aes(x = group, y = value,fill=sim)) +
  geom_bar(stat="identity",position=position_dodge(0.75)) +
  labs(x = 'Group', y = 'Value') +
  ggtitle('Coefficient of Dispersion of RG')
ggsave('COD_RG.png')
t.test(cods_gyration_D_Nter, cods_gyration_D_Nter_unphos, alternative = 'l')  # yes
t.test(cods_gyration_D_Nter, cods_gyration_C_Nter, alternative = 'l')  # no
t.test(cods_gyration_D_Cter, cods_gyration_C_Cter, alternative = 'l')  # no
t.test(cods_gyration_D_Nter, cods_gyration_D_Cter, alternative = 'l')  # yes
t.test(cods_gyration_C_Nter, cods_gyration_C_Cter, alternative = 'l')  # no
```