* R code for measuring bond lengths and angles, March 2017.

#vectors

bond1 = vector()

bond2 = vector()

bond3 = vector()

angle1 = vector()

angle2 = vector()

angle3 = vector()

* Bond Length Code Below

#bond1chain2

sele1 = atom.select(pdb.prot, elety=c("CA","CB"), resno=as.numeric(string2[2]), chain=string2[1])

inds.xyz = c(sele1$xyz)

bond1 <- c(bond1,dist(matrix(pdb.prot$xyz[,inds.xyz], ncol=3, nrow=2, byrow=TRUE)))

* The last line finds the distance for bond length (in angstroms). It is a distance function, using a matrix and the pdb.prot file from the bio3d package to find the locations of the x,y, and z coordinates of the atoms, which are then measured to find bond length. This is for Bond1, Chain2, as displayed by above comments before code, and line chain=string2[1]).

#bond1chain1

sele1 = atom.select(pdb.prot, elety=c("CA","CB"), resno=as.numeric(string1[2]), chain=string1[1])

inds.xyz = c(sele1$xyz)

bond1 <- c(bond1,dist(matrix(pdb.prot$xyz[,inds.xyz], ncol=3, nrow=2, byrow=TRUE)))

#bond2chain2

* -->notifies us which bond and chain are being measured with this code.

sele1 = atom.select(pdb.prot, elety=c("CB","SG"), resno=as.numeric(string2[2]), chain=string2[1])

inds.xyz = c(sele1$xyz)

bond2 <- c(bond2,dist(matrix(pdb.prot$xyz[,inds.xyz], ncol=3, nrow=2, byrow=TRUE)))

* This code perfoms the same action as the above code for Bond1 Chain1, but for Bond2 Chain2, as displayed by the lines of code: chain=string2[1]) and bond2 <- code.

#bond2chain1

sele1 = atom.select(pdb.prot, elety=c("CB","SG"), resno=as.numeric(string1[2]), chain=string1[1])

inds.xyz = c(sele1$xyz)

bond2 <- c(bond2,dist(matrix(pdb.prot$xyz[,inds.xyz], ncol=3, nrow=2, byrow=TRUE)))

* Does same thing as other code for bond angles, but now using Bond2 Chain1.

#bond3

sele1 = atom.select(pdb.prot, elety=c("SG"), resno=as.numeric(string1[2]), chain=string1[1])

sele2 = atom.select(pdb.prot, elety=c("SG"), resno=as.numeric(string2[2]), chain=string2[1])

inds.xyz = c(sele1$xyz, sele2$xyz)

bond3 <- c(bond3,dist(matrix(pdb.prot$xyz[,inds.xyz], ncol=3, nrow=2, byrow=TRUE)))

* Same code for bond 3, but there are no chains, because it is in the middle. Chemically speaking, this is the Sulfur-Sulfur bond directly in the middle. So, the chains from Bond1 and Bond2 connect to Bond3, and Bond3 has no chains as a result.
* Bond Angle Code Below

#BA1chain2

sele1 = atom.select(pdb.prot, elety=c("N", "CA", "CB"), resno=as.numeric(string2[2]), chain=string2[1])

inds.xyz = c(sele1$xyz)

angle1 <- c(angle1,try(angle.xyz(pdb.prot$xyz[,inds.xyz])))

* This code measure the Bond Angle for Bond Angle1, Chain1. It defines and calls upon angle1, then uses the pdb.prot package from the bio3d website to find x, y, and z coordinates of the atoms again. Only this time, as you can see in last line of code, it measures angles instead of distance. Example: angle1 <- c(angle1,try(angle.xyz(pdb.prot$xyz[,inds.xyz])).

#BA1chain1

sele1 = atom.select(pdb.prot, elety=c("N", "CA", "CB"), resno=as.numeric(string1[2]), chain=string1[1])

inds.xyz = c(sele1$xyz)

angle1 <- c(angle1,try(angle.xyz(pdb.prot$xyz[,inds.xyz])))

* Does same thing as set of code above but for Bond Angle1, Chain1 (defined in chain = string statement and angle1 <- code).

#BA2chain2

sele1 = atom.select(pdb.prot, elety=c("CA", "CB", "SG"), resno=as.numeric(string2[2]), chain=string2[1])

inds.xyz = c(sele1$xyz)

angle2 <- c(angle2,try(angle.xyz(pdb.prot$xyz[,inds.xyz])))

* Same set of code, but for Bond Angle2, Chain2

#BA2chain1

sele1 = atom.select(pdb.prot, elety=c("CA", "CB", "SG"), resno=as.numeric(string1[2]), chain=string1[1])

inds.xyz = c(sele1$xyz)

angle2 <- c(angle2,try(angle.xyz(pdb.prot$xyz[,inds.xyz])))

* Same set of code for Bond Angle2, Chain 1

#BA3chain2

sele1 = atom.select(pdb.prot, elety=c("CB", "SG"), resno=as.numeric(string2[2]), chain=string2[1])

sele2 = atom.select(pdb.prot, elety=c("SG"), resno=as.numeric(string1[2]), chain=string1[1])

inds.xyz = c(sele1$xyz,sele2$xyz)

angle3 <- c(angle3,try(angle.xyz(pdb.prot$xyz[,inds.xyz])))

* Same set of code for Bond Angle3, Chain 2

#BA3chain1

sele1 = atom.select(pdb.prot, elety=c("CB", "SG"), resno=as.numeric(string1[2]), chain=string1[1])

sele2 = atom.select(pdb.prot, elety=c("SG"), resno=as.numeric(string2[2]), chain=string1[1])

inds.xyz = c(sele1$xyz,sele2$xyz)

angle3 <- c(angle3,try(angle.xyz(pdb.prot$xyz[,inds.xyz])))

* Same set of code for Bond Angle3, Chain 1

y <- as.character(bond1)

write.csv(y, file = 'bond1.csv', row.names = F)

y <- as.character(bond2)

write.csv(y, file = 'bond2.csv', row.names = F)

y <- as.character(bond3)

write.csv(y, file = 'bond3.csv', row.names = F)

y <- as.character(angle1)

write.csv(y, file = 'angle1.csv', row.names = F)

y <- as.character(angle2)

write.csv(y, file = 'angle2.csv', row.names = F)

y <- as.character(angle3)

write.csv(y, file = 'angle3.csv', row.names = F)

* These y = lines are what develop the excel files. The “write().csv” command creates the excel file while y is a variable that we put are data into and then implement into the excel file. There are multiple lines of code, as a file must be created for each bond length and bond angle.