**Appendix**

**Reading the data in parallel- sample checking code**

import numpy as np

from numpy import arange,array,ones,linalg

from mpi4py import MPI

wt1 = MPI.Wtime()

comm = MPI.COMM\_WORLD

rank = comm.Get\_rank()

mpisize = comm.Get\_size()

def is\_number(s):

try:

float(s)

return s

except ValueError:

return 0

infile=open('dft\_vs\_dft\_elumo\_dump.dat')

num\_lines = sum(1 for line in infile)

print num\_lines

start=rank\*(num\_lines-2)/mpisize+1

end=(rank+1)\*(num\_lines-2)/mpisize

lines=end-start+1

print rank, start,end,lines

x1=np.zeros(lines)

x2=np.zeros(lines)

x3=np.zeros(lines)

x4=np.zeros(lines)

x5=np.zeros(lines)

x6=np.zeros(lines)

x7=np.zeros(lines)

x8=np.zeros(lines)

x9=np.zeros(lines)

infile=open('dft\_vs\_dft\_elumo\_dump.dat')

for \_ in xrange(start+1):

next(infile)

for i,line in enumerate(infile):

if i==(lines):

break

values = line.strip().split(',')

x1[i] = is\_number((values[2]))

x2[i] = is\_number((values[3]))

x3[i] = is\_number((values[4]))

x4[i] = is\_number((values[5]))

x5[i] = is\_number((values[6]))

x6[i] = is\_number((values[7]))

x7[i] = is\_number((values[8]))

x8[i] = is\_number((values[9]))

x9[i] = is\_number((values[10]))

wt2 = MPI.Wtime()

if rank==0:

print mpisize,wt2-wt1

Code written for HPC project

Note- Please note that this code only contain the parallel implementation for a single property. The serial code attached in the email is a more generalized code which would compute correlations between different properties and also plot the data and write the output into text file.