Supporting information

1 PRISM theory calculation program

Python 3.6 source code for PRISM theory calculation of a polymer ring with different N at $T = 450$ K.

```
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
import warnings
warnings.filterwarnings("ignore")
plt.style.use("classic")
from scipy.fftpack import dst
def lj_f(e,sigma,r):
  f=4*e*(-12*(sigma**12/r**10)+6*(sigma**6/r**4))
  return f

def lj_p(r):
  f=4*(1)*((1/(r)**12)-(1/(r)**6))
  return f

def omega(k,N):
  d=1
  sum1=0
  for i in range(N):
    if i==0:
      pass
    else:
      sum1=sum1+(N-i)*np.exp(-k**2*d**2*i*(N-i)/(6*N))
  return 1+(2/N)*sum1

sigma=1
beta=0.101

dr=0.001
r=np.arange(dr,100,dr)
dk=np.pi/(dr*(len(r)+1))
```

```
k=np.linspace(dk,dk*len(r),len(r))
g=np.zeros(len(r))
h=np.zeros(len(r))
y=np.zeros(len(r))
y_o=np.zeros(len(r))
err=10
count=0
alpha=0.2
d=0
c=np.zeros(len(r))
for i in range(len(r)):
  if r[i]<=d:
    pass
  else:
    g[i]=np.exp(-beta*lj_p(r[i]))
    h[i]=g[i]-1
    c[i]=h[i]
coeff_to_fourier=2.0*np.pi*r*dr
coeff_to_real=k*dk/(4*np.pi**2)

hk=np.zeros(len(r))
ck=np.zeros(len(r))
ci=np.zeros(len(r))
wk=np.zeros(len(r))
err=1
count=0
#gk=dst(coeff_to_fourier*g,type=2)/k
#gr=dst(coeff_to_real*gk,type=3)/r
N=input("number of beads: ")
N=int(N)
rho=32.73318525-23.78266866/float(N)
rho=float(rho)*0.395**3
frac=1
while(err>10**(-6)):
  #get direct correlation function to approximate c according to
     closure
  for i in range(len(r)):
    c[i]=(np.exp(-beta*lj_p(r[i]))-1)*(1+h[i]-c[i])
  #Fourier Transform 1D
  ck=dst(coeff_to_fourier*c,type=2)/k
  for i in range(len(h)):
    hk[i]=omega(k[i],N)**2*ck[i]/(1-rho*omega(k[i],N)*ck[i])#omega12[i
       ]**2*ck[i]/(1-rho*omega12[i]*ck[i])
  #Inverse Fourier Transform 1D
  h2=dst(coeff_to_real*hk,type=3)/r
  err=0
  for i in range(len(h)):
    err=err+(h[i]-h2[i])**2/len(h)
    h[i]=frac*(h2[i])+(1-frac)*h[i]
  err=np.sqrt(err)
  if count>=2000:
    break
  count=count+1
  if count%1==0:
    print(count,err)

plt.figure()
plt.rc('font', **{'family': 'serif', 'serif': ['Computer Modern']})
plt.rc('text', usetex=True)
plt.gcf().set_size_inches(4,3,forward=True)
plt.subplots_adjust(left=0.18,bottom=0.15)
g=[ii+1 for ii in h]
plt.plot(r,g,'-r')
plt.plot(r,g,'o',markerfacecolor='w',markeredgecolor='r',markersize=4)
plt.xlim(0,4)
plt.xlabel("$r$")
plt.ylabel("$g(r)$")
plt.savefig("g-"+str(N)+".png",dpi=300)

filename2 = "g-"+str(N)+"-py.txt"
writefile = open(filename2,'w')
for i in range(len(r)):
  writefile.write(str(r[i]))
  writefile.write("\t")
  writefile.write(str(g[i]))
  writefile.write("\n")
```