Supporting Material for:

Simulated Self-Assembly of the HIV-1 Capsid: Protein Shape and Native Contacts are Sufficient for Two-Dimensional Lattice Formation

Bo Chen and Robert Tycko[*]
Laboratory of Chemical Physics
National Institute of Diabetes and Digestive and Kidney Diseases
National Institutes of Health
Bethesda, MD 20892

[*]corresponding author: Dr. Robert Tycko, National Institutes of Health, Building 5, Room 112, Bethesda, MD 20892-0520. phone: 301-402-8272. fax: 301-496-0825. e-mail: robertty@mail.nih.gov.

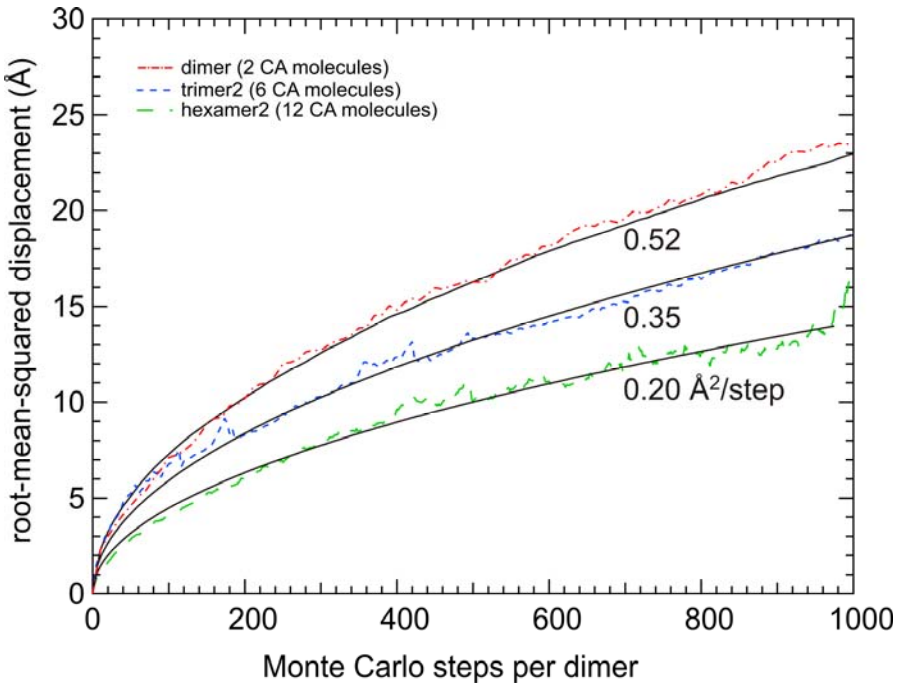

Figure S1. Dependences of root-mean-squared displacement on MC steps per dimer from simulations on isolated CA dimer, trimer2, and hexamer2 units at ε' = 13.4. Solid lines are least-squares fits with the expression $d_{rms} = (Dn_{step})^{1/2}$, yielding the indicated values of the diffusion constant D. Values of D have the ratios 2.60:1.74:1.00. If the diffusion constants were inversely proportional to the square root of the oligomer number, the ratios would be 2.45:1.73:1.00. (If the exponent of n_{step} is treated as an adjustable parameter, the best-fit values from dimer, trimer2, and hexamer2 simulations are 0.522, 0.461, and 0.499, respectively.)

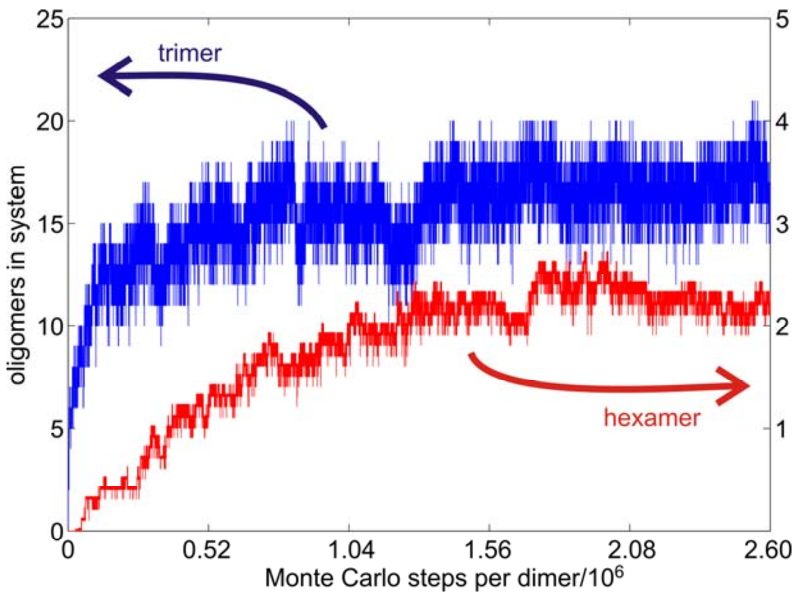

Figure S2. Average numbers of trimer2 and hexamer2 units in the early stages of 10 simulations of CA self-assembly at ε' = 13.4 and $[CA]_{2D}$ = 0.016 nm^{-2}. Consistently, trimer2 populations develop before the appearance of a hexagonal lattice.

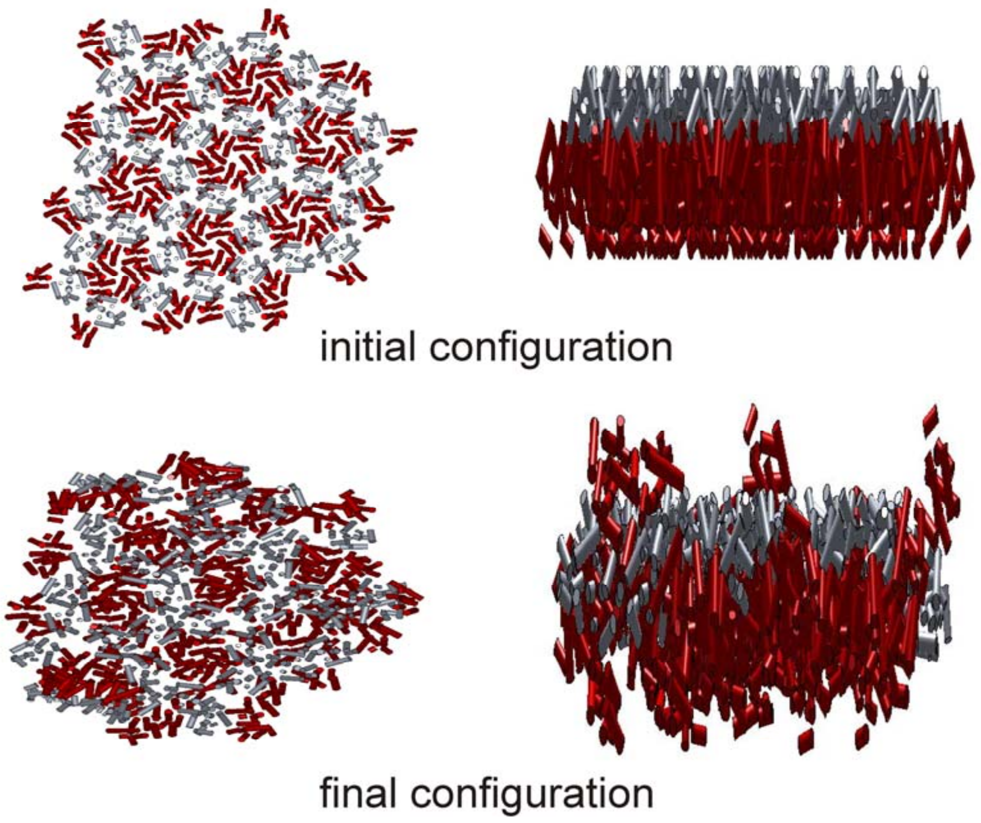

Figure S3. Initial (top) and final (bottom) configurations from MC simulations of CA lattice stability with translational and rotational motions in three dimensions, showing top (left) and side (right) views. 5.26 X 10^6 MC steps per dimer were performed with ε' = 13.4 and with the centers of mass of the CA dimers contained within a 2497 nm X 2497 nm X 4.0 nm box with periodic boundary conditions. The 4.0 nm box height is roughly equal to the diameter of a hexamer2 unit and roughly twice the thickness of an ideal hexagonal lattice, so that complete loss of positional and orientational order could occur in principle. In fact, the hexagonal lattice structure of the initial configuration is largely preserved, with disorder developing primarily at the edges of the cluster.

Supporting Methods

Hard-wall repulsion algorithm

MC moves were rejected whenever any two cylinders (representing α-helices) from different CA dimers overlapped, using the algorithm in the Fortran95 subroutine shown below. Inputs to this subroutine are points at the top and bottom of the two cylinders (bc1 and bc2) and the radii of the cylinders (r1 and f2). The subroutine returns ID = 1 when the cylinders overlap:

```
        subroutine cylinderdist_full(bc1,bc2,r1,r2,mu_a,mu_b,dist,ID)
        real,intent(in)::bc1(2,3),bc2(2,3),r1,r2
        real,intent(out)::mu_a(1,3),mu_b(1,3),dist
        integer,intent(out)::ID
        real::A(1,3),B(1,3),C(1,3),D(1,3),M1(3,2),M2(3,2)
        real::p(1,2),q(1,2),r(1,2),s(1,2),n1(1,2),n2(1,2)
        real::d1,d2,u,v
        real::left,right
        logical judge1, judge2
                interface mean
                function mean1(v,d)
                real,intent(in)::v(:)
                integer,intent(in)::d
                real::mean1
                endfunction mean1

                function mean2(v,d)
                real,intent(in)::v(:,:)
                integer,intent(in)::d
                real,target::mean21(1,size(v,2)),mean22(size(v,1),1)
                real,pointer::mean2(:,:)
                endfunction mean2

                function mean3(v,d)
                real,intent(in)::v(:,:,:)
                integer,intent(in)::d
                real,target::mean31(1,size(v,2),size(v,3)),mean32(size(v,1),1,size(v,3))&
                     &,mean33(size(v,1),size(v,2),1)
                real,pointer::mean3(:,:,:)
                endfunction mean3

                function mean4(v,d)
                real,intent(in)::v(:,:,:,:)
                integer,intent(in)::d
real,target::mean41(1,size(v,2),size(v,3),size(v,4)),mean42(size(v,1),1,size(v,3),size(v,4))&
          &,mean43(size(v,1),size(v,2),1,size(v,4)),mean44(size(v,1),size(v,2),size(v,3),1)
                real,pointer::mean4(:,:,:,:)
                endfunction mean4
                endinterface mean
         A=reshape((/bc1(1,:)/),(/1,3/))
         B=reshape((/bc1(2,:)/),(/1,3/))
         C=reshape((/bc2(1,:)/),(/1,3/))
         D=reshape((/bc2(2,:)/),(/1,3/))
        M1=null_3b3(A-B);
        r=matmul((C-A),M1);
        s=matmul((D-A),M1);
        n1=matmul((s-r),reshape((/0,1,-1,0/),(/2,2/)))
        n1=n1/norm_bo(n1)
        v=dot(n1,s)
        d1=norm_bo(v*n1-r)/norm_bo(s-r)*cos_bo((v*n1-r),(s-r))
        j=0
        if(d1>1.0 .or. d1==1.0) then
            d1=1.0
            j=j+1
        elseif (d1<0.0 .or. d1==0.0) then
            d1=0.0
            j=j+1
```

```
endif
mu_b=C+d1*(D-C)
M2=null_3b3(D-C)
p=matmul((A-D),M2)
q=matmul((B-D),M2)
n2=matmul((q-p),reshape((/0,1,-1,0/),(/2,2/)))
n2=n2/norm_bo(n2)
u=dot(n2,q);
d2=norm_bo(u*n2-p)/norm_bo(q-p)*cos_bo((u*n2-p),(q-p))
if (d2>1.0.or.d2==1.0)then
    d2=1.0
    j=j+2
elseif (d2<0.0 .or. d2==0.0)then
    d2=0.0
    j=j+2
endif
mu_a=A+d2*(B-A)
select case(j)
    case(1)
        d2=dot((mu_b-A),(B-A))/(norm_bo(B-A))**2
        if (d2>1.0 .or.d2==1.0)then
            d2=1.0
            j=j+2
        elseif (d2<0.0 .or. d2==0)then
            d2=0.0
            j=j+2
        endif
        mu_a=A+d2*(B-A)
    case(2)
        d1=dot((mu_a-C),(D-C))/(norm_bo(D-C))**2;
        if (d1>1.0.or.d1==1)then
            d1=1.0
            j=j+1
        elseif (d1<0.0.or.d1==0)then
            d1=0.0
            j=j+1
        endif
        mu_b=C+d1*(D-C)
    case(3)
        d2=dot((mu_b-A),(B-A))/(norm_bo(B-A))**2
        d1=dot((mu_a-C),(D-C))/(norm_bo(D-C))**2
        if (d2<1)then
        if(d2>0)then
            j=j-2
            mu_a=A+d2*(B-A)
        endif
        endif

        if (d1<1)then
        if(d1>0)then
            j=j-1
             mu_b=C+d1*(D-C)
        endif
        endif

endselect
ID=0
if (norm_bo(mu_a-mu_b)<(r1+r2))then
    select case(j)
        case(0)
        ID=1
        case(1)
            if ((norm_bo(mu_b-mu_a)-r1)<r2*sin_bo((mu_b-mu_a),(C-D)))then
                ID=1

            endif
        case(2)

             if ((norm_bo(mu_b-mu_a)-r2)<r1*sin_bo((mu_b-mu_a),(A-B)))then
                ID=1
            endif
```

```
               case(3)
       if(abs(dot((mu_b-mu_a),(C-D)))/norm_bo(C-D)<r1)then
       if(abs(dot((mu_b-mu_a),(A-B)))/norm_bo(A-B)<r2)then
if(norm_bo(mu_a-mu_b)<sqrt(r1**2+r2**2+2*r1*r2*cos_bo((mu_a-mean(bc1,1)),(mu_b-
mean(bc2,1)))))then
       ID=1

                   endif
                   endif
                   endif
           endselect

       endif
         dist=norm_bo(mu_a-mu_b)
       endsubroutine cylinderdist_full
```

Hexamer counting algorithm

In order to generate graphs in Figs. 4 and 5, we needed to count the number of hexamer2 units in each configuration from an MC run. To do this, we identified the two nearest neighbors A and C of CA dimer B (considering only their centers of mass) and tested whether the nearest neighbor distances AB and BC were both equal to 3.05 ± 0.5 nm (the expected distance within a hexamer2 unit) and whether the angle ABC formed by the three centers of mass was 120° ± 43°. If the result was negative, dimer B was not in a hexamer2 unit. If the result was positive, the same test was applied to CA dimer C. If the result was again positive, the test was repeated sequentially, up to five total times, thereby identifying a set of seven CA dimers, A-G. If dimers A and G were the same, then dimer B was considered to belong to a hexamer2 unit, as shown below. After applying this process to each CA dimer, the total number of dimers belonging to hexamer2 units was divided by six to generate the number of hexamer2 units.