A Novel Hybrid Single Molecule Approach Reveals Spontaneous DNA Motion in the Nucleosome

Sijie Wei[1], Samantha J. Falk[2], Ben E. Black[2], and Tae-Hee Lee[1,]*

[1]Department of Chemistry, The Pennsylvania State University, University Park, PA 16802. [2]Department of Biochemistry and Biophysics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA 19104

*Correspondence to: Tae-Hee Lee, txl18@psu.edu

Supplementary Data

Figure S1. Nucleosomal DNA constructs used for the measurements. A single strand 20 base linker DNA with a biotin (gray filled circle) at its 5' end was added to the 5' end of a 147bp human α-satellite sequence (SAT)(1) or the Widom 601 sequence (601)(2). Cy3 was labeled along the phosphate backbone of the Watson (forward) strand replacing the +2nd nucleotide from the dyad of the nucleosome, and ATTO647N was labeled terminally after the +73rd nucleotide of the Crick (reverse) strand. The distance between the two fluorophores is about 4nm in the crystal structure, corresponding to a FRET efficiency value of ~0.7. As a control, 40bp ds-DNA with an identical 20 base DNA linker was labeled with Cy3 (green) (DNA-D). The same DNA control was labeled with Cy3 and ATTO647N (red) (DNA-DA) resulting in 0.6~0.7 FRET. The sequence and the labeling positions are listed in Table S6.

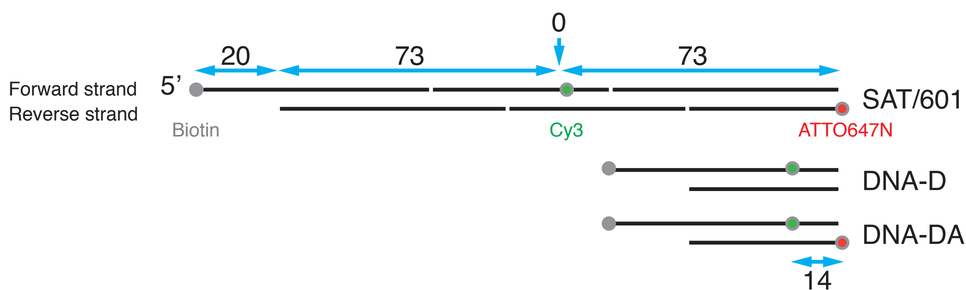

Figure S2. Gel shift assay results for nucleosomal DNA constructs and nucleosomes. a, A urea denaturing gel image (5% PAGE at 7M urea and 0.4x TBE) of a nucleosomal DNA construct (H3-SAT) shows that the ligation reaction is complete and no nicked product is visible (i.e. no complete set of nicked partners are present). Ligation products were denatured at 7M urea for 10min at 80°C before loading. We used 30min ligation time for sample preparation because it was the shortest time resulting in almost complete ligation. Further incubation did not result in significant change in the ligation yield. The major products are the 147bp and 167bp nucleosomal DNA strands as designed. The dim longer (>200bp) and shorter strand bands (107bp based on the lengths of the oligonucleotides) are likely due to unwanted ligation and errors in the measured amounts of the oligonucleotides for the ligation reaction. These products will not form canonical nucleosomes and are filtered out during data collection or analysis (see the Single molecule measurements section in Materials and Methods). **b**, A native gel (5% PAGE at 0.2x TBE) image of a nucleosome sample (H3-SAT) shows that annealing shifts and homogenizes the positioning and the final centrifuged sample (supernatant, white rectangle) used for the measurements contains significantly reduced aggregates.

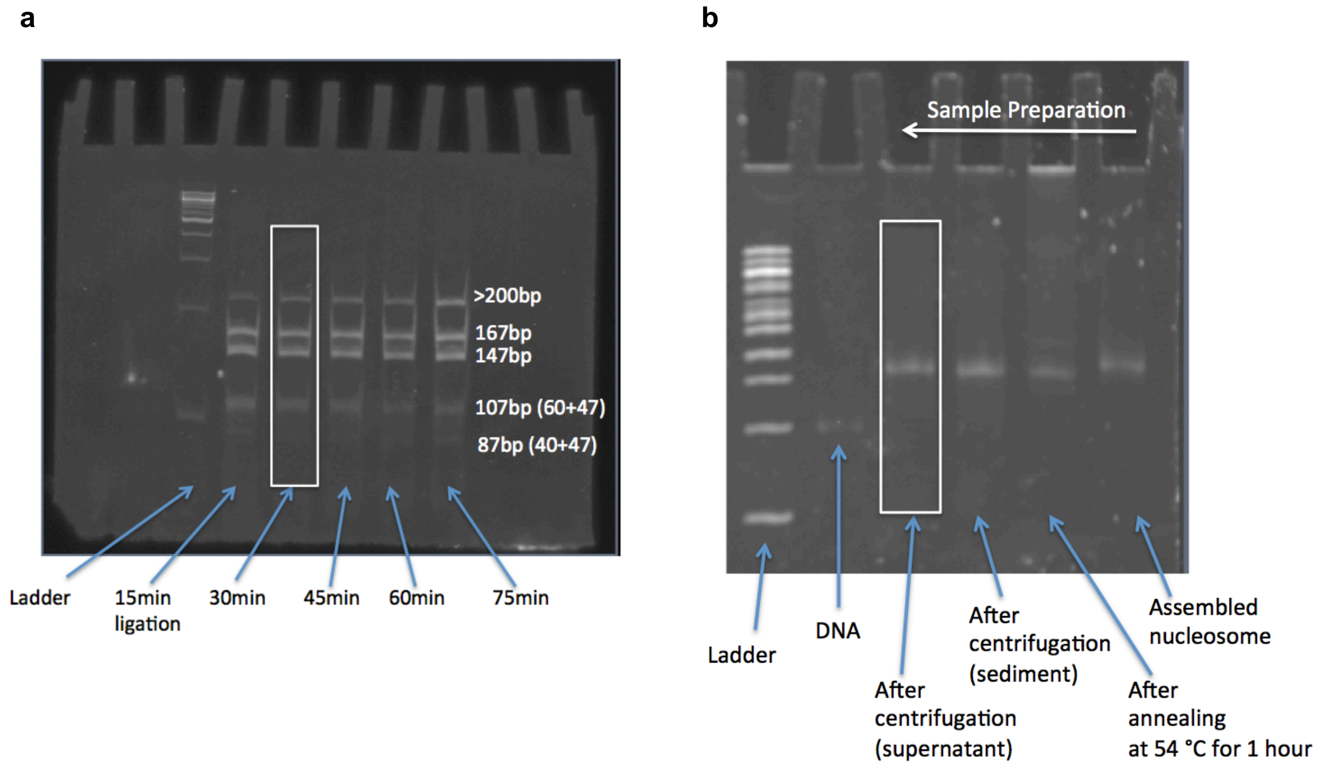

Figure S3. FCS spectra of H3-SAT nucleosomes and DNA controls at 100mM NaCl reveal that ATTO647N correlation has a much lower noise level than Cy3 correlation. a, FCS spectra (0.4μs to 20ms) calculated from the Cy3 photon streams out of the H3-SAT nucleosome (filled triangle) and DNA-D control (empty triangle) in the 100mM NaCl TE buffer. **b**, FCS spectra (0.4μs to 20ms) calculated from the ATTO647N photon streams out of the H3-SAT nucleosome (filled triangle) and DNA-DA control (empty triangle) in the 100mM NaCl TE buffer.

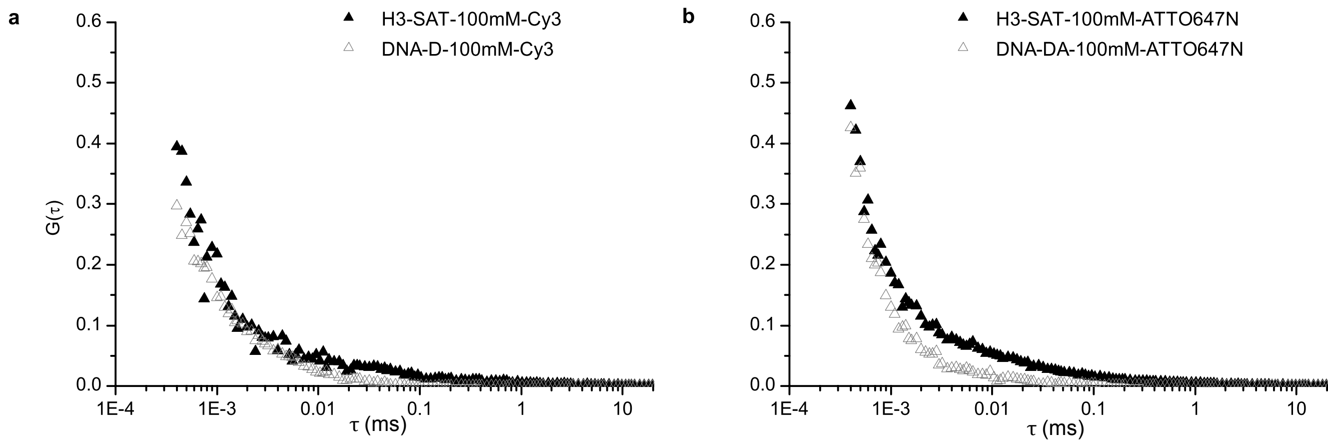

Figure S4. FCS spectra of H3-601 and CA-SAT nucleosomes over the entire spectral range. a, FCS spectra (0.4μs to 20ms) calculated from the ATTO647N photon streams out of H3-601 nucleosomes in 2mM (filled square), 10mM (empty square), 50mM (filled circle), and 100mM(empty circle) NaCl TE buffers. DNA-DA control is also shown for comparison (empty triangle). **b**, FCS spectra (0.4μs to 20ms) calculated from the ATTO647N photon streams out of CA-SAT nucleosomes.

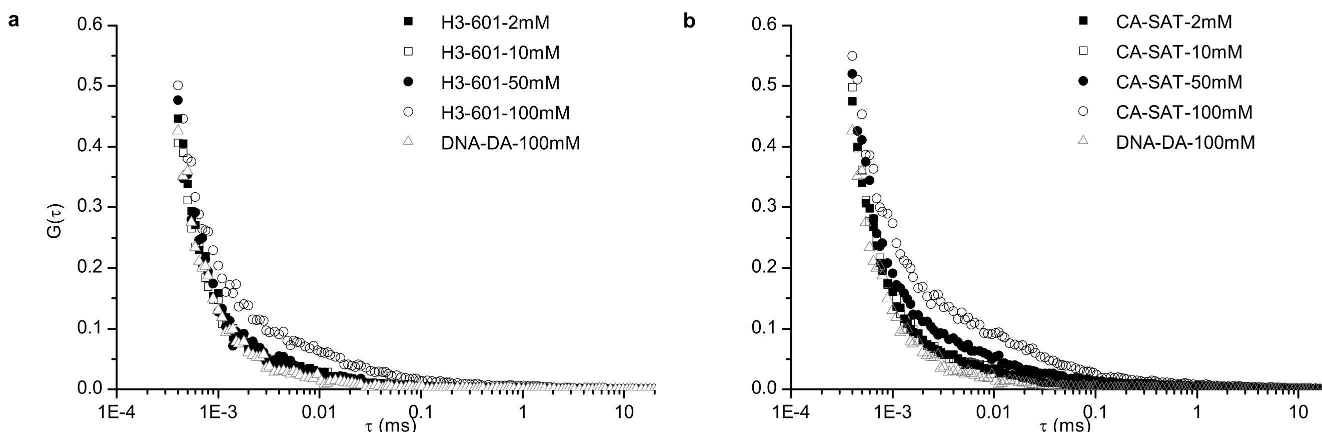

Figure S5. Photophysical dynamics of fluorophores revealed in FCS spectra. FCS spectra (0.4μs to 10μs) from all the samples used in the measurements in 2mM (filled square), 10mM (empty square), 50mM (filled circle), and 100mM (empty circle) NaCl TE buffers. These curves have identical shapes to the DNA controls (empty triangle), indicating that the dynamics at <0.01ms is likely due to the fluorophore photophysical dynamics.

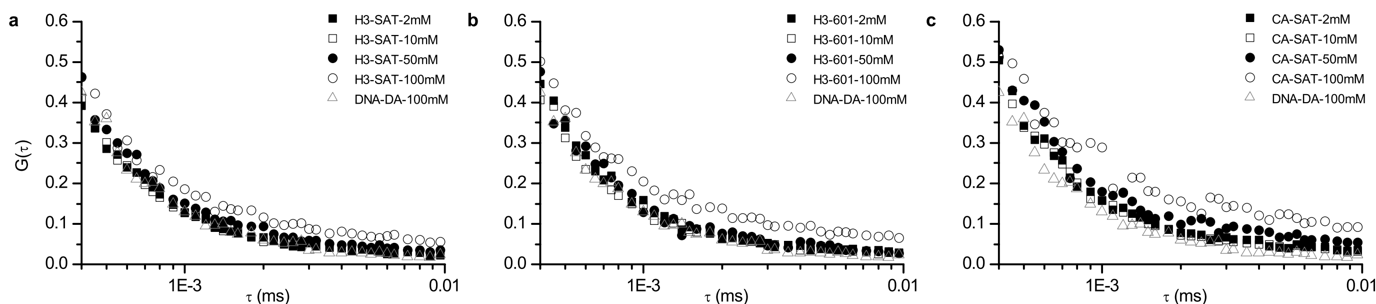

Figure S6. The rate constants for FRET decrease (k_1, a) and increase (k_{-1}, b) dynamics, respectively, on a 10~30μs timescale.

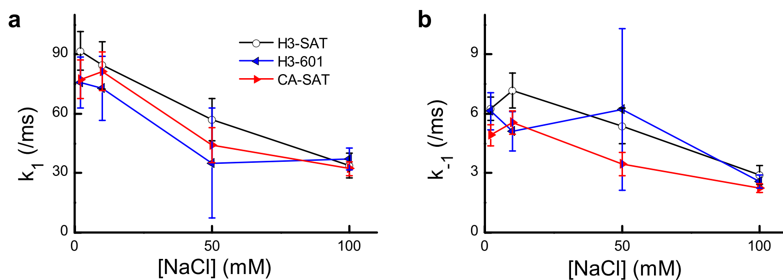

Table S1. FRET data statistics

	[NaCl]	2mM		10mM		50mM		100mM	
H3-SAT	Total	1237		1140		786		614	
	Analyzed	1221	98.7%	1132	99.3%	780	99.2%	607	98.9%
H3-601	Total	1013		962		734		609	
	Analyzed	489	48.3%	447	46.5%	331	45.1%	292	47.9%
CA-SAT	Total	999		1052		565		485	
	Analyzed	998	99.9%	1051	99.9%	562	99.5%	480	99.0%

Table S2. MLE performance on simulated traces

	ε_{close}	ε_{open}	k_{open}	k_{close}
Given	0.723	0.566	0.364	1.177
MLE estimates	0.742	0.579	0.375	1.125
Error (%)	2.25	2.47	2.92	4.62
s. d. (%)	1.9	4.7	56	39

s.d. = standard deviation

MLE performed on 25 simulated traces with a 500ms effective data range on average. The photon emission rate is 32kHz.

Table S3. MLE estimated FRET efficiencies (average ε_{close} and ε_{open})

	ε_{close}					ε_{open}					$\Delta\varepsilon$
[NaCl]	2mm	10mm	50mm	100mm	Avg	2mm	10mm	50mm	100mm	Avg	ε_{close} - ε_{open}
H3-SAT	0.739	0.732	0.745	0.741	**0.739** ±0.005	0.625	0.611	0.628	0.607	**0.618** ±0.010	**0.122** ±0.012
H3-601	0.726	0.747	0.736	0.735	**0.736** ±0.009	0.580	0.616	0.596	0.601	**0.598** ±0.015	**0.138** ±0.017
CA-SAT	0.683	0.665	0.685	0.702	**0.684** ±0.015	0.544	0.510	0.555	0.555	**0.541** ±0.021	**0.143** ±0.026

Table S4. Average FRET efficiencies from MLE estimation and from ATTO647N normalized intensity (calculated from 1ms time-binned traces)

	ε_{avg} from MLE				ε_{avg} from binned traces			
[NaCl]	2mM	10mM	50mM	100mM	2mM	10mM	50mM	100mM
H3-SAT	0.698	0.688	0.696	0.690	0.698	0.688	0.696	0.690
H3-601	0.682	0.705	0.687	0.685	0.681	0.704	0.687	0.683
CA-SAT	0.643	0.624	0.639	0.646	0.643	0.623	0.638	0.646

Table S5. The Student's t-test results for the FRET efficiency differences between the canonical (H3-SAT) and CENP-A (CA-SAT) nucleosomes.

	ε_{open}				ε_{close}			
[NaCl]	2mM	10mM	50mM	100mM	2mM	10mM	50mM	100mM
p-value*	6.2e-7	1.2e-9	1.3e-5	1.4e-3	9.2e-7	4.3e-12	1.7e-6	1.2e-3

*p-value is the probability by which the CENP-A nucleosome has a FRET efficiency not lower than the canonical nucleosome. The values are essentially negligible, indicating that the FRET efficiencies are always lower with the CENP-A nucleosome.

Table S6. DNA sequences and labeling positions (The dyad is squared. The single strand linker region is underscored. Cy3, ATTO647N, and biotin are marked green, red, and grey respectively.)

Human α-satellite sequence－forward strand	5'-/Biotin/CAACGAAATC CTCCGAGAGG TCCAAATATC CACCTGCAGA TTCTACCAAA AGTGTATTTG. GAAACTGCTC CATCAAAAGG CATGTTCAGC TCTGT/Cy3/AGTG AAACTCCATC ATCACAAAGA ATATTCTGAG AATGCTTCCG TTTGCCTTTT ATATGAACTT CCTTCCT
Human α-satellite sequence－reverse strand	5'-/ATTO647N/AGGAAGGAAG TTCATATAAA AGGCAAACGG AAGCATTCTC. AGAATATTCT TTGTGATGAT GGAGTTTCAC TCACAGAGCT GAACATG.CCT TTTGATGGAG CAGTTTCCAA ATACACTTTT GGTAGAATCT GCAGGTGGAT ATTTGGA
601 sequence－forward strand	5'-/Biotin/CAACGAAATC CTCCGAGAGG ATCGAGAATC CCGGTGCCGA GGCCGCTCAA TTGGTCGTAG. ACAGCTCTAG CACCGCTTAA ACGCACGTAC GCGCT/Cy3/TCCC CCGCGTT.TTA ACCGCCAAGG GGATTACTCC CTAGTCTCCA GGCACGTGTC AGATATATAC ATCCGAT
601 sequence－reverse strand	5'-/ATTO647N/ATCGGATGTA TATATCTGAC ACGTGCCTGG AGACTAGGGA. GTAATCCCCT TGGCGGTTAA AACGCGGGGG ACAGCGCGTA CGTGCGT.TTA AGCGGTGCTA GAGCTGTCTA CGACCAATTG AGCGGCCTCG GCACCGGGAT TCTCGAT
DNA－D－forward strand	5'-/Biotin/ATC ATCACAAAGA ATATTCTGAG AATGCTTCCG TTTGCCTTTT ATAT/Cy3/AACTT CCTTCCT
DNA－D－reverse strand	5'-AGGAAGGAAG TTCATATAAA AGGCAAACGG AAGCATTCTC
DNA－DA－forward strand	5'-/Biotin/ATC ATCACAAAGA ATATTCTGAG AATGCTTCCG TTTGCCTTTT ATAT/Cy3/AACTT CCTTCCT
DNA－DA－reverse strand	5'-/ATTO647N/AGGAAGGAAG TTCATATAAA AGGCAAACGG AAGCATTCTC

Computer code for FCS spectra calculation (written by Sijie Wei)

This code is written in IDL (Interactive Data Language) (Exelis, Boulder CO). Refer to the IDL manual for variable/array type/definition and descriptions of how to use a user-defined function and the built-in functions. The comments below are to calculate donor, acceptor, and donor-acceptor FCS spectra from a single FRET trace. For further details on the algorithm, refer to the references 3 and 4.

```
FUNCTION F2Cor, prn
;This function calculate the FCS spectra of FRET donor, acceptor, donor-acceptor from a set
of two photon streams (donor and acceptor).
;prn should be set to 0 for a single FRET trace

common parameter6, FCS_bin_number, FCS_lag, SCH_Ks, SCH_Ke, period_photon_number_p,
period_photon_time_d1, period_photon_time_a2, period_photon_time_interval_d1,
period_photon_time_interval_a2, base_unit
;These arrays and variables contain inputs (data and FCS parameters as commented below)



;INPUT:
;FCS_bin_number is the number of the correlation lag points. For descriptions on the "lag
points", see the references 3&4.
;FCS_bin_number=8*(number of k levels)=8*20=160 in our implementation. For descriptions on
the "k levels", see the references 3&4.
;FCS_lag = [8,9,10,11,12,13,14,15,16,18,20,22,24,26,28,30,32,36,...]. The size of this array
is 8x20=160. For further information, see the references 3&4.

;SCH_Ks is the first k level of the SCH (simple correlation histogram) method in IDL and was
set to 0 in our implementation.
;SCH_Ke is the last k level of the SCH method in IDL and was set to 10 in our implementation.

;period_photon_number_p=ulonarr(period_number,3)
;period_photon_number_p[*,0:2] stores the numbers of total, donor, and acceptor photons for
all the FRET segments.

;period_photon_time_d1=ulonarr(period_max_photon_number1,period_number)
;period_photon_time_a2=ulonarr(period_max_photon_number2,period_number)
;For a single FRET trace, period_number is set to 1.
;period_max_photon_number1 defines the number of photons in the donor channel
;period_max_photon_number2 defines the number of photons in the acceptor channel
;period_photon_time_d1[*,0] stores the arrival times of donor photons.
;period_photon_time_a2[*,0] stores the arrival times of acceptor photons.

;period_photon_time_interval_d1=ulonarr(period_max_photon_number1,period_number)
;period_photon_time_interval_a2=ulonarr(period_max_photon_number2,period_number)
;period_photon_time_interval_d1[*,0] stores the time intervals of donor photons.
;period_photon_time_interval_a2[*,0] stores the time intervals of acceptor photons.

;base_unit=1.0



;OUTPUT: FCS_hist_period=fltarr(6,FCS_bin_number)
;FCS_hist_period[0:2,0:FCS_bin_number-1] stores FCS spectra for the FCS time (lag) range of
[0:FCS_bin_number-1] for donor, acceptor and donor-acceptor, respectively for 0:2.
```

```
FCS_hist_period=fltarr(6,FCS_bin_number)

delay=ulonarr(FCS_lag[SCH_Ke*8])
G0=fltarr(FCS_lag[SCH_Ke*8])
GTr=fltarr(SCH_Ke*8)
GBM=fltarr(FCS_bin_number)
Mdir=fltarr(FCS_bin_number)
Mdel=fltarr(FCS_bin_number)

temp_L_period=float(period_photon_time_d1[period_photon_number_p[prn,1]-1l,prn])/base_uni
t

lag_to=value_locate(FCS_lag,floor(temp_L_period/5.0))
K_to=lag_to/8

L_period=floor(temp_L_period/2.0^float(K_to))*2.0^float(K_to)

  p_to=0l
  p_to_done=0
  ptn=period_photon_number_p[prn,1]-1l
  while not p_to_done do begin

    if float(period_photon_time_d1[ptn,prn])/base_unit le L_period then begin
      p_to_done=1
p_to=long(ptn)

endif else begin

      ptn=ptn-1l
if ptn lt 0 then p_to_done=1

endelse

  endwhile

  m=0l
  delay[*]=0l
  G0[*]=0.0

  for pn=0l,p_to do begin
m=m+1l

for k=m, 1l, -1l do delay[k]=delay[k-1l] +
ulong(float(period_photon_time_interval_d1[pn,prn])/base_unit)

while delay[m] ge FCS_lag[SCH_Ke*8] do begin
    m=m-1l
endwhile

for k=1l,m do G0[delay[k]]=G0[delay[k]]+1.0

  endfor

  GTr[*]=0.0

  GTr[0:7]=G0[8:15]
  for kn=1, SCH_Ke-1 do begin
```

```
for l=8,15 do begin

GTr[kn*8+l-8]=G0[FCS_lag[kn*8+l-8]]

for i=1,2.0^float(kn)-1 do begin

        GTr[kn*8+l-8]=GTr[kn*8+l-8]+(G0[FCS_lag[kn*8+l-8]-i]+
G0[FCS_lag[kn*8+l-8]+i])*(2.0^float(kn)-float(i))/(2.0^float(kn))

Endfor

    endfor
  endfor

  GBM[*]=0.0
  Mdir[*]=0.0
  Mdel[*]=0.0
  lag_done=0
  kn=0

  while not lag_done do begin

    I_number=ulong(L_period/(2.0^float(kn)))
    I=intarr(I_number)

temp11=histogram(float(period_photon_time_d1
[0:period_photon_number_p[prn,1]-11,prn])/base_unit,  BINSIZE=2.0^float(kn),MIN=0.0)

    I[0:I_number-1]=temp11[0:I_number-1]

    for l=8,15 do begin

      if kn eq K_to and kn*8+l-8 eq lag_to then begin
        lag_done=1
        break
      endif

      Mdir[kn*8+l-8]=float(total(I[0:I_number-1-l]))/float(I_number-l)
      Mdel[kn*8+l-8]=float(total(I[l:I_number-1]))/float(I_number-l)

if kn lt SCH_Ke then begin

        FCS_hist_period[0,kn*8+l-8]=FCS_hist_period[0,kn*8+l-8]
+GTr[kn*8+l-8]/Mdir[kn*8+l-8]/Mdel[kn*8+l-8]/float(I_number-l)-1.0

endif else begin

        GBM[(kn)*8+l-8]=total(float(I[0:I_number-1-l])
*float(I[l:I_number-1]))/float(I_number-l)

        FCS_hist_period[0,kn*8+l-8]=FCS_hist_period[0,kn*8+l-8]
+GBM[(kn)*8+l-8]/(Mdir[kn*8+l-8]*Mdel[kn*8+l-8])-1.0

      endelse

    endfor
    kn=kn+1
```

```
  endwhile

FCS_hist_period[3,0:lag_to]=1.0


  temp_L_period=float(period_photon_time_a2
[period_photon_number_p[prn,2]-1l,prn])/base_unit

  lag_to=value_locate(FCS_lag,floor(temp_L_period/5.0))
  K_to=lag_to/8
  L_period=floor(temp_L_period/2.0^float(K_to))*2.0^float(K_to)

  p_to=0l
  p_to_done=0
  ptn=period_photon_number_p[prn,2]-1l

  while not p_to_done do begin

    if float(period_photon_time_a2[ptn,prn])/base_unit le L_period then begin
      p_to_done=1
p_to=long(ptn)

endif else begin

      ptn=ptn-1l
if ptn lt 0 then p_to_done=1

endelse

  endwhile

  m=0l
  delay[*]=0l
  G0[*]=0.0

  for pn=0l,p_to do begin
m=m+1l;

for k=m, 1l, -1l do delay[k]=delay[k-1l]+
ulong(float(period_photon_time_interval_a2[pn,prn])/base_unit)

    while delay[m] ge FCS_lag[SCH_Ke*8] do begin
      m=m-1l
endwhile

    for k=1l,m do G0[delay[k]]=G0[delay[k]]+1.0
  endfor

  GTr[*]=0.0

  GTr[0:7]=G0[8:15]
  for kn=1, SCH_Ke-1 do begin
for l=8,15 do begin

GTr[kn*8+l-8]=G0[FCS_lag[kn*8+l-8]]

for i=1,2.0^float(kn)-1 do begin
```

```
        GTr[kn*8+l-8]=GTr[kn*8+l-8]+(G0[FCS_lag[kn*8+l-8]-i]
+G0[FCS_lag[kn*8+l-8]+i])*(2.0^float(kn)-float(i))/(2.0^float(kn))

      endfor
    endfor
  endfor


  GBM[*]=0.0
  Mdir[*]=0.0
  Mdel[*]=0.0
  lag_done=0
  kn=0
  while not lag_done do begin

    I_number=ulong(L_period/(2.0^float(kn)))
    I=intarr(I_number)

temp11=histogram(float(period_photon_time_a2
[0:period_photon_number_p[prn,2]-1,prn])/base_unit, BINSIZE=2.0^float(kn),MIN=0.0)

    I[0:I_number-1]=temp11[0:I_number-1]

    for l=8,15 do begin

      if kn eq K_to and kn*8+l-8 eq lag_to then begin
        lag_done=1
        break
      endif

      Mdir[kn*8+l-8]=float(total(I[0:I_number-1-l]))/float(I_number-l)
      Mdel[kn*8+l-8]=float(total(I[l:I_number-1]))/float(I_number-l)

if kn lt SCH_Ke then begin

        FCS_hist_period[1,kn*8+l-8]=FCS_hist_period[1,kn*8+l-8]
+GTr[kn*8+l-8]/Mdir[kn*8+l-8]/Mdel[kn*8+l-8]/float(I_number-l)-1.0

endif else begin

        GBM[(kn)*8+l-8]=total(float(I[0:I_number-1-l])
*float(I[l:I_number-1]))/float(I_number-l)

        FCS_hist_period[1,kn*8+l-8]=FCS_hist_period[1,kn*8+l-8]
+GBM[(kn)*8+l-8]/(Mdir[kn*8+l-8]*Mdel[kn*8+l-8])-1.0

      endelse

    endfor
    kn=kn+1

  endwhile

  FCS_hist_period[4,0:lag_to]=1.0


  temp_L_period=min([float(period_photon_time_a2
```

```
[period_photon_number_p [prn,2]-1,prn])/base_unit,
float(period_photon_time_d1[period_photon_number_p[prn,1]-1,prn])/base_unit])

  lag_to=value_locate(FCS_lag,floor(temp_L_period/5.0))

  K_to=lag_to/8
  L_period=floor(temp_L_period/2.0^float(K_to))*2.0^float(K_to)

  GBM[*]=0.0
  Mdir[*]=0.0
  Mdel[*]=0.0
  lag_done=0
  kn=0
  while not lag_done do begin

    I_number=ulong(L_period/(2.0^float(kn)))
    I=intarr(I_number)
    II=intarr(I_number)

    temp11=histogram(float(period_photon_time_d1
[0:period_photon_number_p[prn,1]-11,prn])/base_unit, BINSIZE=2.0^float(kn),MIN=0.0)
    I[0:I_number-1]=temp11[0:I_number-1]

    temp22=histogram(float(period_photon_time_a2
[0:period_photon_number_p[prn,2]-11,prn])/base_unit, BINSIZE=2.0^float(kn),MIN=0.0)
    II[0:I_number-1]=temp22[0:I_number-1]

    for l=8,15 do begin

      if kn eq K_to and kn*8+l-8 eq lag_to then begin
        lag_done=1
        break
      endif

      Mdir[kn*8+l-8]=float(total(I[0:I_number-1-l]))/float(I_number-l)
      Mdel[kn*8+l-8]=float(total(II[l:I_number-1]))/float(I_number-l)

GBM[(kn)*8+l-8]=total(float(I[0:I_number-1-l]) *float(II[l:I_number-1]))/float(I_number-l)

      FCS_hist_period[2,kn*8+l-8]=FCS_hist_period[2,kn*8+l-8]
+GBM[(kn)*8+l-8]/(Mdir[kn*8+l-8]*Mdel[kn*8+l-8])-1.0

    endfor
    kn=kn+1

  endwhile

  FCS_hist_period[5,0:lag_to]=1.0

  return,FCS_hist_period
END
```

Computer code for likelihood calculation (for MLE, written by Sijie Wei)

This code is written in IDL (Interactive Data Language) (Exelis, Boulder CO). Refer to the IDL manual for variable/array type/definition and descriptions of how to use a user-defined function and the built-in functions. The comments below are to calculate -1 times the value of the likelihood function with a set of dynamics parameters and a combined photon stream (donor + acceptor). Use the AMOEBA function built in IDL to maximize the likelihood function. For further details on the likelihood function, refer to the references 5 and 6.

```
FUNCTION MLE, A
;This function calculates the -1*likelihood as defined in the method with a given dynamics
parameter set A and a combined photon trajectory. For further information, see the method and
the references cited therein. MLE was done with a built-in function minimization sub-routine
(AMOEBA) in IDL. See the IDL manual for detailed information on the AMOEBA routine.

common parameter4, keeper
common parameter2, molecule_number, photon_number, photon_color, photon_time_interval,
input_unit

;INPUT:
;A[0] : state 1, unfolded state with lower FRET efficiency
;A[1] : state 2, folded state with higher FRET efficiency
;A[2] : equilibrium population density of the state 1
;A[3] : the sum of the opening/closing rates (k1 + k2)

;keeper = 2.115d
;If underflow happens during calculation, try to increase this value by 0.1

;molecule_number = length of the FRET trace in ms / 25ms.
;photon_number=ulonarr(molecule_number,3)
;photon_number[*,0:2] should store the number of total, donor, and acceptor photons in a FRET
trace segmented by molecule_number.

;photon_color=intarr(max_photon_number,molecule_number)
;photon_color should store the color of all the photons in a FRET trace segmented by
molecule_number.

;photon_time_interval=ulonarr(max_photon_number,molecule_number)
;photon_time_interval should store the time intervals of the photons (donor and acceptor
combined in one array)

;input_unit=0.000050 (microseconds). This is the time unit of the data.

;OUTPUT: the logarithmic likelihood of observing the photon stream with the dynamics parameter
set A

  e=fltarr(3)
  e[1]=A[0]
  e[2]=A[1]

  p=fltarr(3)
  p[1]=A[2]
  p[2]=1.0-p[1]

  k=fltarr(3)
  k[0]=A[3]
  k[1]=p[2]*k[0]
  k[2]=p[1]*k[0]
```

```
  Phi=dblarr(2,2,3)
  Phi[0,0,2]=p[1]*E[1]+p[2]*E[2]
  Phi[0,1,2]=(E[2]-E[1])*p[2]
  Phi[1,0,2]=(E[2]-E[1])*p[1]
  Phi[1,1,2]=p[1]*E[2]+p[2]*E[1]

  Phi[0,0,1]=1.0-Phi[0,0,2]
  Phi[0,1,1]=0.0-Phi[0,1,2]
  Phi[1,0,1]=0.0-Phi[1,0,2]
  Phi[1,1,1]=1.0-Phi[1,1,2]

  Tau=dblarr(2,2)
  Tau[0,0]=1.0
  Tau[0,1]=0.0
  Tau[1,0]=0.0

  St=dblarr(1,2)
  En=dblarr(2,1)
  En=[1.0,0.0]

  L=dblarr(molecule_number)

  temp0=dblarr(2,2)

  for mn=0, molecule_number-1 do begin
    St=[[1.0],[0.0]]

    for pnc=1, photon_number[mn,0]-1 do begin
pn=photon_number[mn,0]-pnc

Tau[1,1]=exp((-1.0)*k[0]*float(photon_time_interval[pn,mn])* input_unit)

temp0=Phi[*,*,photon_color[pn,mn]]#Tau

St=St#temp0*keeper
    endfor

St=St#phi[*,*,photon_color[0,mn]]
L[mn]=St#En

  endfor

  Ltot=dblarr(1)
  Ltot[0]=total(ALOG10(L),1)

  RETURN,-Ltot

END
```

References

1. Harp, J.M., Uberbacher, E.C., Roberson, A.E., Palmer, E.L., Gewiess, A. and Bunick, G.J. (1996) X-ray diffraction analysis of crystals containing twofold symmetric nucleosome core particles. *Acta Crystallogr D Biol Crystallogr*, **52**, 283-288.
2. Lowary, P.T. and Widom, J. (1998) New DNA sequence rules for high affinity binding to histone octamer and sequence-directed nucleosome positioning. *J Mol Biol*, **276**, 19-42.
3. Schaub, E. (2012) F2Cor: fast 2-stage correlation algorithm for FCS and DLS. *Opt Express*, **20**, 2184-2195.
4. Schaub, E. (2013) High countrate real-time FCS using F2Cor. *Opt Express*, **21**, 23543-23555.
5. Chung, H.S., Gopich, I.V., McHale, K., Cellmer, T., Louis, J.M. and Eaton, W.A. (2011) Extracting rate coefficients from single-molecule photon trajectories and FRET efficiency histograms for a fast-folding protein. *J Phys Chem A*, **115**, 3642-3656.
6. Gopich, I.V. and Szabo, A. (2009) Decoding the pattern of photon colors in single-molecule FRET. *J Phys Chem B*, **113**, 10965-10973.