Electronic Supplementary Information

Hierarchical Self-Assembly and Controlled Disassembly of a Cavitand-based Host-Guest Supramolecular Polymers.

Daniele Zuccaccia,[a] Roberta Pinalli,[b] Rita De Zorzi,[c] Monica Semeraro,[d] Alberto Credi,*[d] Cristiano Zuccaccia,[e] Alceo Macchioni,[e], Silvano Geremia,*[c] and Enrico Dalcanale*[b]

[a]Dipartimento di Chimica, Fisica e Ambiente, Università di Udine, Via Cotonificio 108, 33100 Udine, Italy

[b]Dipartimento di Chimica and Unità INSTM, UdR Parma, Università di Parma, Viale delle Scienze 17/A, 43124 Parma, Italy

[c]Dipartimento di Scienze Chimiche e Farmaceutiche,Università di Trieste, via Giorgieri, 1, 34127 Trieste, Italy

[d]Dipartimento di Chimica Industriale "Toso Montanari", Università di Bologna, Viale del Risorgimento 4, 40136 Bologna, Italy

[e]Dipartimento di Chimica, Biologia e Biotecnologia and CIRCC, Università di Perugia, Via Elce di Sotto 8, 06123 Perugia, Italy

Table of contents:

1. Octave scripts

1.1 Calculation of the hydrodynamic translational diffusion coefficients

Di.m: Script for the calculation of the hydrodynamic translational diffusion coefficient D_t by fitting data obtained from HYDRO with Perrin's equation.

```
clear
clc
close all
pkg load optim
a=1;                    ## Monomer dimension, parallel to the elongation direction
b=1;                    ## Monomer dimension, perpendicular to the elongation direction
load Dhydro             ## Input values from HYDRO
Dcalcinit(1)=Dhydro(2,1);
for k=2:10000
   Perrin(k)=a/b*k;
   chi(k)=sqrt(abs((Perrin(k)^2)-1))/Perrin(k);
   S(k)=(2/chi(k))*atanh(chi(k));
   Dcalcinit(k)=Dhydro(2,1)/(2*(k^(1/3))*(Perrin(k)^(2/3))/S(k));
end
Dconstott=nlinfit(Dhydro(1,1:11),Dhydro(2,1:11),@dfun,Dhydro(2,1))
Dcalc(1)=Dconstott;
for i=2:10000
   Dcalc(i)=Dconstott/(2*(i^(1/3))*(Perrin(i)^(2/3))/S(i));
end
save Dcalc Dcalc        ## Output values of Di
```

dfun.m: Function from Perrin's equation.

```
function Di=dfun(beta0,X)
D0=abs(beta0(1,1));
Z=size(X,2);
a=1;                    ## Monomer dimension, parallel to the elongation direction
b=1;                    ## Monomer dimension, perpendicular to the elongation direction
Di(1,1)=D0;
for j=2:Z
   Perrin(j)=a/b*j;
   chi(j)=sqrt(abs((Perrin(j)^2)-1))/Perrin(j);
   S(j)=(2/chi(j))*atanh(chi(j));
   Di(1,j)=D0/(2*(j^(1/3))*(Perrin(j)^(2/3))/S(j));
end
```

1.2 SEK model

SEK_model.m: Optimization of the polymerization constant from PGSE data, using a Sequential Equal K (SEK) model.

```
## Optimization of the SEK constant for each series of data corresponding to PGSE
## measurements with different starting concentrations of the monomer
clear
clc
close all
pkg load optim
pkg load struct
pkg load io
pkg load statistics
'SEK model, optimization over single series'
## ------ INPUT: ------
## 1) Monomer starting concentrations and experimental Δ parameter values
load monomer_conc
## 2) Values of the Field Gradients G applied
load G
## 3) Experimental resonance intensities of the samples
load I
## 4) Resonance intensities of the internal standard
load TMSS
## 5) Diffusion coefficients of oligomers
load Dcalc
## ------------------------
## Selection of the series (with different monomer concentration):
ser=input('Series number:');
TMSS1=0;
LNTMSS=0;
G1=0;
G2=0;
Iexp=0;
LNIexp=0;
Icalc=0;
LNIcalc=0;
A=0;
CT=monomer_conc(1,ser);
Delta=monomer_conc(2,ser);
Cost=((42.5756e6)^2)*((5e-3)^2)*((Delta*1e-3)-((5e-3)/3));
Xt=0;
Yt=0;
for i=1:size(G,1)
   if G(i,ser)==0
   else
      if TMSS(i, ser)<1e4
      else
         TMSS1(i,1)=TMSS(i,ser);
         LNTMSS(i,1)=log(TMSS1(i,1));
      end
      G1(i,1)=G(i,ser);
      G2(i,1)=G1(i,1)^2;
      Iexp(i,1)=I(i,ser);
      LNIexp(i,1)=log(Iexp(i,1));
end
end
Xt(1:size(LNTMSS,1),1)=1;
Xt(1:size(LNTMSS,1),2)=G2(1:size(LNTMSS,1),1);
Yt(1:size(LNTMSS,1),1)=LNTMSS(1:size(LNTMSS,1),1);
## Linear regression on data of the internal standard (TMSS)
kappa=LinearRegression(Xt, Yt);
ksperim=-kappa(2)/(Cost*1.07e-5);
##
## Initial values for I0 and K(SEK):
```

```
I0=Iexp(1,1);
Kstart=1e4;
##
dati=0;
dati(1:size(G1,1),1)=G1(1:size(G1,1),1);
dati(1:size(G1,1),2)=CT;
dati(1:size(G1,1),3)=Cost;
dati(1:size(G1,1),4)=ksperim;
dati(1:size(G1,1),5)=ser;
dati(1:size(G1,1),6)=Iexp(1:size(G1,1),1);
X=0;
Y=0;
beta0=0;
X(1:size(dati,1),1:4)=dati(1:size(dati,1),1:4);
X(1:size(dati,1),5)=1;
Y(1:size(dati,1),1)=dati(1:size(dati,1),6);
beta0(1,1)=Kstart;
beta0(2,1)=I0;
save Start dati beta0
beta1=0;
## Non linear optimization:
beta1=nlinfit(X, Y, @SEKoptimfun, beta0)
##
## ------ OUTPUT: ------
## 1) Optimized value of the thermodynamic constant K(SEK)
Kfinal=abs(beta1(1,1))
## 2) Optimized value of sample intensity for G=0 (I0)
I0optim=abs(beta1(2,1))
save Optimized Kfinal I0optim
C1=0;
X0(1)=0;
if CT<1/K
   X0(2)=CT;
else
   X0(2)=1/K;
end
## 3) Distribution of the polymeric species, calculated using optimized K(SEK)
##        3.a) monomer concentration:
save tmp_data Kfinal CT
[C1, fval, info]=fzero(@SEKdistrall, X0);
##        3.b) distribution of oligomers up to i=10000:
A=0;
if or (info==1, info==2, info==3)
   A(1,1)=abs(C1);
   for i=2:10000;
      A(i,1)=Kfinal*A(1,1)*A(i-1,1);
   end
   save DISTRIB A
   if or(info==2, info==3)
      'WARNING!'
   end
else
   'ERROR! Not converging'
end
## 4) Resonance intensities calculated from the polymeric distribution
for j=1:size(G1,1)
   sum=0;
   for i=1:size(A,1)
      expo=exp(-dati(j,4)*dati(j,3)*Dcalc(i)*(dati(j,1)^2));
      sum=sum+(A(i,1)*i/dati(j,2))*expo;
   end
   Icalc(j,1)=I0optim*sum;
   LNIcalc(j,1)=log(Icalc(j,1));
end
save IandG G1 G2 Iexp Icalc LNIexp LNIcalc
```

```
## 5) Residuals of intensity and monomer concentration
R=0;
for i=1:size(G1,1)
   R=R+(Iexp(i,1)-Icalc(i,1))^2;
end
ResidualI=abs(sqrt(R)/size(G1,1))
monomer=0;
for i=1:size(A,1)
   monomer=monomer+A(i,1)*i;
end
Residual_monomer=abs(CT-monomer);
save residue ResidualI Residual_monomer
## ------------------------
## ------ FIGURES ------
## FIGURE 1: plot of intensities (I) against gradient field values (G)
## Experimental values: black line, marker +
## Calculated values: red line, marker o
figure(1)
plot(G1,Iexp,'k+-')
hold on
plot(G1,Icalc,'ro-')
## FIGURE 2: plot of logarithm of intensities (ln(I)) against gradient field values
## squared (G^2)
## Experimental values: black line, marker +
## Calculated values: red line, marker o
figure(2)
plot(G2,LNIexp,'k+-')
hold on
plot(G2,LNIcalc,'ro-')
## FIGURE 3: distribution of the polymeric species, concentration against polymerization
## index (up to i=50)
figure(3)
plot(1:50,A(1:50,1),'k-')
```

SEKoptimfun.m: Function from the PGSE equation.

```
function I=SEKoptimfun(beta0,X)
K=abs(beta0(1,1));            ## Initial value of the SEK constant
I0=abs(beta0(2,1));           ## Initial value of the intensity at G=0
G1=0;
G1=X(1:size(X,1),1);          ## Input values of the field gradient
CT=X(1,2); ## Input value of the total concentration of the monomer
Cost=X(1,3); ## Input value of PGSE constant
ksperim=X(1,4);               ## Input value of the experimental constant determined by
                              ## linear regression of the internal standard data
I=0;
load Dcalc                    ## Hydrodynamic translational diffusion coefficients
save tmp_data K CT
X0(1)=0;
if CT<1/K
   X0(2)=CT;
else
   X0(2)=1/K;
end
[C1,fval,info]=fzero(@SEKdistribution,X0);        ## Concentration of the monomer
                                                  ## according to the SEK distribution
if or(info==1,info==2,info==3)
   A(1,1)=abs(C1);
   for i=2:10000;
      A(i,1)=K*A(1,1)*A(i-1,1);            ## Concentration of the polymeric species
   end
   save SEK A
   if or(info==2,info==3)
      'WARNING!'
   end
else
   'ERROR! Not converging'
   K
   load SEK
end
for j=1:size(G1,1)
   sum=0;
   for i=1:10000
      expo=exp(-ksperim*Cost*Dcalc(i)*(G1(j)^2));
      sum=sum+(A(i,1)*i/CT)*expo;
   end
   I(j,1)=I0*sum;
end
A=0;
```

SEKdistribution.m: Function for the calculation of the monomer concentration from defined values of total monomer concentration and polymerization constant according to the SEK model.

```
function y=SEKdistribution(x)
load tmp_data
y=CT*((1-K*x)^2)-x;
```

1.3 AK model

AK_model.m: Optimization of the polymerization constant from PGSE data, using an Attenuated K (AK) model.

```
## Optimization of the AK constant for each series of data corresponding to PGSE
## measurements with different starting concentrations of the monomer
clear
clc
close all
pkg load optim
pkg load struct
pkg load io
pkg load statistics
'AK model, optimization over single series'
## ------ INPUT: ------
## 1) Monomer starting concentrations and experimental Δ parameter values
load monomer_conc
## 2) Values of the Field Gradients G applied
load G
## 3) Experimental resonance intensities of the samples
load I
## 4) Resonance intensities of the internal standard
load TMSS
## 5) Diffusion coefficients of oligomers
load Dcalc
## ------------------------
## Selection of the series (with different monomer concentration):
ser=input('Series number:');
TMSS1=0;
LNTMSS=0;
G1=0;
G2=0;
Iexp=0;
LNIexp=0;
Icalc=0;
LNIcalc=0;
A=0;
CT=monomer_conc(1,ser);
Delta=monomer_conc(2,ser);
Cost=((42.5756e6)^2)*((5e-3)^2)*((Delta*1e-3)-((5e-3)/3));
Xt=0;
Yt=0;
for i=1:size(G,1)
   if G(i,ser)==0
   else
      if TMSS(i, ser)<1e4
      else
         TMSS1(i,1)=TMSS(i,ser);
         LNTMSS(i,1)=log(TMSS1(i,1));
      end
      G1(i,1)=G(i,ser);
      G2(i,1)=G1(i,1)^2;
      Iexp(i,1)=I(i,ser);
      LNIexp(i,1)=log(Iexp(i,1));
   end
end
Xt(1:size(LNTMSS,1),1)=1;
Xt(1:size(LNTMSS,1),2)=G2(1:size(LNTMSS,1),1);
Yt(1:size(LNTMSS,1),1)=LNTMSS(1:size(LNTMSS,1),1);
## Linear regression on data of the internal standard (TMSS)
kappa=LinearRegression(Xt, Yt);
ksperim=-kappa(2)/(Cost*1.07e-5);
##
## Initial values for I0 and K(AK):
```

```
I0=Iexp(1,1);
Kstart=1e4;
##
dati=0;
dati(1:size(G1,1),1)=G1(1:size(G1,1),1);
dati(1:size(G1,1),2)=CT;
dati(1:size(G1,1),3)=Cost;
dati(1:size(G1,1),4)=ksperim;
dati(1:size(G1,1),5)=ser;
dati(1:size(G1,1),6)=Iexp(1:size(G1,1),1);
X=0;
Y=0;
beta0=0;
X(1:size(dati,1),1:4)=dati(1:size(dati,1),1:4);
X(1:size(dati,1),5)=1;
Y(1:size(dati,1),1)=dati(1:size(dati,1),6);
beta0(1,1)=Kstart;
beta0(2,1)=I0;
save Start dati beta0
beta1=0;
## Non linear optimization:
beta1=nlinfit(X, Y, @AKoptimfun, beta0)
##
## ------ OUTPUT: ------
## 1) Optimized value of the thermodynamic constant K(AK)
Kfinal=abs(beta1(1,1))
## 2) Optimized value of sample intensity for G=0 (I0)
I0optim=abs(beta1(2,1))
save Optimized Kfinal I0optim
C1=0;
X0(1)=0;
X0(2)=CT;
## 3) Distribution of the polymeric species, calculated using optimized K(AK)
##        3.a) monomer concentration:
save tmp_data Kfinal CT
[C1, fval, info]=fzero(@AKdistribution, X0);
##        3.b) distribution of oligomers up to i=10000:
A=0;
if or (info==1, info==2, info==3)
   A(1,1)=abs(C1);
   for i=2:10000;
      A(i,1)=Kfinal*A(1,1)*A(i-1,1)/i;
   end
   save DISTRIB A
   if or(info==2, info==3)
      'WARNING!'
   end
else
   'ERROR! Not converging'
end
## 4) Resonance intensities calculated from the polymeric distribution
for j=1:size(G1,1)
   sum=0;
   for i=1:size(A,1)
      expo=exp(-dati(j,4)*dati(j,3)*Dcalc(i)*(dati(j,1)^2));
      sum=sum+(A(i,1)*i/dati(j,2))*expo;
   end
   Icalc(j,1)=I0optim*sum;
   LNIcalc(j,1)=log(Icalc(j,1));
end
save IandG G1 G2 Iexp Icalc LNIexp LNIcalc
## 5) Residuals of intensity and monomer concentration
R=0;
for i=1:size(G1,1)
   R=R+(Iexp(i,1)-Icalc(i,1))^2;
```

```
end
ResidualI=abs(sqrt(R)/size(G1,1))
monomer=0;
for i=1:size(A,1)
   monomer=monomer+A(i,1)*i;
end
Residual_monomer=abs(CT-monomer);
save residue ResidualI Residual_monomer
## ------------------------
## ------ FIGURES ------
## FIGURE 1: plot of intensities (I) against gradient field values (G)
## Experimental values: black line, marker +
## Calculated values: red line, marker o
figure(1)
plot(G1,Iexp,'k+-')
hold on
plot(G1,Icalc,'ro-')
## FIGURE 2: plot of logarithm of intensities (ln(I)) against gradient field values
## squared (G^2)
## Experimental values: black line, marker +
## Calculated values: red line, marker o
figure(2)
plot(G2,LNIexp,'k+-')
hold on
plot(G2,LNIcalc,'ro-')
## FIGURE 3: distribution of the polymeric species, concentration against polymerization
## index (up to i=50)
figure(3)
plot(1:50,A(1:50,1),'k-')
```

AKoptimfun.m: Function from the PGSE equation.

```
function I=AKoptimfun(beta0,X)
K=abs(beta0(1,1));            ## Initial value of the AK constant
I0=abs(beta0(2,1));           ## Initial value of the intensity at G=0
G1=0;
G1=X(1:size(X,1),1);          ## Input values of the field gradient
CT=X(1,2);                    ## Input value of the total concentration of the monomer
Cost=X(1,3);                  ## Input value of PGSE constant
ksperim=X(1,4);               ## Input value of the experimental constant determined by
                              ## linear regression of the internal standard data
I=0;
load Dcalc                    ## Hydrodynamic translational diffusion coefficients
save tmp_data K CT
X0(1)=0;
X0(2)=CT;
[C1,fval,info]=fzero(@AKdistribution,X0);         ## Concentration of the monomer
                                                  ## according to the AK distribution
if or(info==1,info==2,info==3)
   A(1,1)=abs(C1);
   for i=2:10000;
      A(i,1)=(K/i)*A(1,1)*A(i-1,1);        ## Concentration of the polymeric species
   end
   save AK A
   if or(info==2,info==3)
      'WARNING!'
   end
else
   'ERROR! Not converging'
   K
   load AK
end
for j=1:size(G1,1)
   sum=0;
   for i=1:10000
      expo=exp(-ksperim*Cost*Dcalc(i)*(G1(j)^2));
      sum=sum+(A(i,1)*i/CT)*expo;
   end
   I(j,1)=I0*sum;
end
A=0;
```

AKdistribution.m: Function for the calculation of the monomer concentration from defined values of total monomer concentration and polymerization constant according to the AK model.

```
function y=AKdistribution(x)
load tmp_data
y=CT-(x*exp(K*x));
```

1.4 SEK-TAi model

SEKTAi_model.m: Optimization of the polymerization constant from PGSE data, using a SEK with aggregation increasing with *i* model.

```
## Optimization of the SEK constant and the aggregation constant KA, with Kai=KA*i, for
## each series of data corresponding to PGSE measurements with different starting
## concentrations of the monomer
clear
clc
close all
pkg load optim
pkg load struct
pkg load io
pkg load statistics
'SEK-TAi model, optimization over single series'
## ------ INPUT: ------
## 1) Monomer starting concentrations and experimental Δ parameter values
load monomer_conc
## 2) Values of the Field Gradients G applied
load G
## 3) Experimental resonance intensities of the samples
load I
## 4) Resonance intensities of the internal standard
load TMSS
## 5) Diffusion coefficients of oligomers
load Dcalc
## ------------------------
## Selection of the series (with different monomer concentration):
ser=input('Series number:');
TMSS1=0;
LNTMSS=0;
G1=0;
G2=0;
Iexp=0;
LNIexp=0;
Icalc=0;
LNIcalc=0;
CT=monomer_conc(1,ser);
Delta=monomer_conc(2,ser);
Cost=((42.5756e6)^2)*((5e-3)^2)*((Delta*1e-3)-((5e-3)/3));
Xt=0;
Yt=0;
for i=1:size(G,1)
   if G(i,ser)==0
   else
      if TMSS(i, ser)<1e4
      else
         TMSS1(i,1)=TMSS(i,ser);
         LNTMSS(i,1)=log(TMSS1(i,1));
      end
      G1(i,1)=G(i,ser);
      G2(i,1)=G1(i,1)^2;
      Iexp(i,1)=I(i,ser);
      LNIexp(i,1)=log(Iexp(i,1));
   end
end
Xt(1:size(LNTMSS,1),1)=1;
Xt(1:size(LNTMSS,1),2)=G2(1:size(LNTMSS,1),1);
Yt(1:size(LNTMSS,1),1)=LNTMSS(1:size(LNTMSS,1),1);
## Linear regression on data of the internal standard (TMSS)
kappa=LinearRegression(Xt, Yt);
ksperim=-kappa(2)/(Cost*1.07e-5);
##
## Initial values for I0, K(SEK) and KA (TAi):
```

```
I0=Iexp(1,1);
Kstart=1e4;
KAstart=1e10;
##
dati=0;
dati(1:size(G1,1),1)=G1(1:size(G1,1),1);
dati(1:size(G1,1),2)=CT;
dati(1:size(G1,1),3)=Cost;
dati(1:size(G1,1),4)=ksperim;
dati(1:size(G1,1),5)=ser;
dati(1:size(G1,1),6)=Iexp(1:size(G1,1),1);
X=0;
Y=0;
beta0=0;
X(1:size(dati,1),1:4)=dati(1:size(dati,1),1:4);
X(1:size(dati,1),5)=1;
Y(1:size(dati,1),1)=dati(1:size(dati,1),6);
beta0(1,1)=Kstart;
beta0(2,1)=KAstart;
beta0(3,1)=I0;
save Start dati beta0
beta1=0;
## Non linear optimization:
beta1=nlinfit(X, Y, @SEKTAioptimfun, beta0)
##
## ------ OUTPUT: ------
## 1) Optimized value of the thermodynamic constants K(SEK) and KA (TAi)
Kfinal=abs(beta1(1,1))
KAfinal=abs(beta1(2,1))
## 2) Optimized value of sample intensity for G=0 (I0)
I0optim=abs(beta1(3,1))
save Optimized Kfinal KAfinal I0optim
C1=0;
K=Kfinal;
KA=KAfinal;
X0(1)=0;
if CT<1/K
   X0(2)=CT;
else
   X0(2)=1/K;
end
## 3) Distribution of polymeric species, calculated using optimized K (SEK) and KA (TAi)
##        3.a) monomer concentration:
save tmp_data K KA CT
[C1, fval, info]=fzero(@SEKTAidistrib, X0);
##        3.b) distribution of oligomers up to i=10000:
A=0;
AA=0;
if or (info==1, info==2, info==3)
   A(1,1)=abs(C1);
   for i=2:10000;
      A(i,1)=Kfinal*A(1,1)*A(i-1,1);
   end
##        3.c) distribution of tetrameric aggregates up to i=2500:
   for i=1:2500
      AA(i,1)=KA*i*(A(i,1)^4);
   end
   save DISTRIB A AA
   if or(info==2, info==3)
      'WARNING!'
   end
else
   'ERROR! Not converging'
end
## 4) Resonance intensities calculated from the polymeric distribution
```

```
for j=1:size(G1,1)
   sum=0;
   for i=1:size(A,1)
      expo=exp(-dati(j,4)*dati(j,3)*Dcalc(i)*(dati(j,1)^2));
      sum=sum+(A(i,1)*i/dati(j,2))*expo;
   end
   for i=1:size(AA,1)
      expo=exp(-dati(j,4)*dati(j,3)*Dcalc(4*i)*(dati(j,1)^2));
      sum=sum+(AA(i,1)*4*i/dati(j,2))*expo;
   end
   Icalc(j,1)=I0optim*sum;
   LNIcalc(j,1)=log(Icalc(j,1));
end
save IandG G1 G2 Iexp Icalc LNIexp LNIcalc
## 5) Residuals of intensity and monomer concentration
R=0;
for i=1:size(G1,1)
   R=R+(Iexp(i,1)-Icalc(i,1))^2;
end
ResidualI=abs(sqrt(R)/size(G1,1))
monomer=0;
for i=1:size(A,1)
   monomer=monomer+A(i,1)*i;
end
for i=1:size(AA,1)
   monomer=monomer+AA(i,1)*4*i;
end
Residual_monomer=abs(CT-monomer);
save residue ResidualI Residual_monomer
## -------------------------
## ------ FIGURES ------
## FIGURE 1: plot of intensities (I) against gradient field values (G)
## Experimental values: black line, marker +
## Calculated values: red line, marker o
figure(1)
plot(G1,Iexp,'k+-')
hold on
plot(G1,Icalc,'ro-')
## FIGURE 2: plot of logarithm of intensities (ln(I)) against gradient field values
## squared (G^2)
## Experimental values: black line, marker +
## Calculated values: red line, marker o
figure(2)
plot(G2,LNIexp,'k+-')
hold on
plot(G2,LNIcalc,'ro-')
## FIGURE 3: distribution of the polymeric species and the tetrameric aggregates,
## concentration against polymerization index (up to i=50)
## Polymeric species: black line
## Tetrameric aggregates: red line
figure(3)
plot(1:50,A(1:50,1),'k-')
hold on
plot(1:50,AA(1:50,1),'r-')
```

SEKTAioptimfun.m: Function from the PGSE equation.

```
function I=SEKTAioptimfun(beta0,X)
K=abs(beta0(1,1));     ## Initial value of the SEK constant
KA=abs(beta0(2,1));    ## Initial value of the TAi constant
I0=abs(beta0(3,1));    ## Initial value of the intensity at G=0
G1=0;
G1=X(1:size(X,1),1);   ## Input values of the field gradient
CT=X(1,2);             ## Input value of the total concentration of the monomer
```

```
Cost=X(1,3);           ## Input value of PGSE constant
ksperim=X(1,4);        ## Input value of the experimental constant determined by
                       ## linear regression of the internal standard data
I=0;
load Dcalc             ## Hydrodynamic translational diffusion coefficients
save tmp_data K KA CT
X0(1)=0;
if CT<1/K
   X0(2)=CT;
else
   X0(2)=1/K;
end
[C1,fval,info]=fzero(@SEKTAidistrib,X0);          ## Concentration of the monomer
                                                  ## according to SEKTAi distribution
if or(info==1,info==2,info==3)
   A(1,1)=abs(C1);
   for i=2:10000;
      A(i,1)=K*A(1,1)*A(i-1,1);             ## Concentration of the polymeric species
   end
   for i=1:2500
      AA(i,1)=KA*i*(A(i,1)^4);              ## Concentration of the tetrameric aggregates
   end
   save SEKTAi A AA
   if or(info==2,info==3)
      'WARNING!'
   end
else
   'ERROR! Not converging'
   load SEKTAi
end
for j=1:size(G1,1)
   sum=0;
   for i=1:10000
      expo=exp(-ksperim*Cost*Dcalc(i)*(G1(j,1)^2));
      sum=sum+(A(i,1)*i/CT)*expo;
   end
   for i=1:size(AA,1)
      expo=exp(-ksperim*Cost*Dcalc(4*i)*(G1(j,1)^2));
      sum=sum+(AA(i,1)*4*i/CT)*expo;
   end
   I(j,1)=I0*sum;
end
A=0;
AA=0;
```

<u>SEKTAidistrib.m</u>: Function for the calculation of the monomer concentration from defined values of total monomer concentration and polymerization constants according to the SEK-TAi model.

```
function y=SEKTAidistrib(x)
load tmp_data
y=CT*((1-(K*x))^3)*((1+(K*x))^3)*((1+((K*x)^2))^3)-(x*(1-
(K*x))*((1+(K*x))^3)*((1+((K*x)^2))^3))-(4*KA*x^4)*(1+((K*x)^4));
```

1.5 SEK-TAi^4 model

SEKTAi4_model.m: Optimization of the polymerization constant from PGSE data, using a SEK with aggregation increasing with i^4 model.

```
## Optimization of the SEK constant and the aggregation constant KA, with KAi=KA*(i^4),
## for each series of data corresponding to PGSE measurements with different starting
## concentrations of the monomer
clear
clc
close all
pkg load optim
pkg load struct
pkg load io
pkg load statistics
'SEK-TAi4 model, optimization over single series'
## ------ INPUT: ------
## 1) Monomer starting concentrations and experimental Δ parameter values
load monomer_conc
## 2) Values of the Field Gradients G applied
load G
## 3) Experimental resonance intensities of the samples
load I
## 4) Resonance intensities of the internal standard
load TMSS
## 5) Diffusion coefficients of oligomers
load Dcalc
## ------------------------
## Selection of the series (with different monomer concentration):
ser=input('Series number:');
TMSS1=0;
LNTMSS=0;
G1=0;
G2=0;
Iexp=0;
LNIexp=0;
Icalc=0;
LNIcalc=0;
A=0;
CT=monomer_conc(1,ser);
Delta=monomer_conc(2,ser);
Cost=((42.5756e6)^2)*((5e-3)^2)*((Delta*1e-3)-((5e-3)/3));
Xt=0;
Yt=0;
for i=1:size(G,1)
   if G(i,ser)==0
   else
      if TMSS(i,ser)<1e4
      else
         TMSS1(i,1)=TMSS(i,ser);
         LNTMSS(i,1)=log(TMSS1(i,1));
      end
      G1(i,1)=G(i,ser);
      G2(i,1)=G1(i,1)^2;
      Iexp(i,1)=I(i,ser);
      LNIexp(i,1)=log(Iexp(i,1));
   end
end
Xt(1:size(LNTMSS,1),1)=1;
Xt(1:size(LNTMSS,1),2)=G2(1:size(LNTMSS,1),1);
Yt(1:size(LNTMSS,1),1)=LNTMSS(1:size(LNTMSS,1),1);
## Linear regression on data of the internal standard (TMSS)
kappa=LinearRegression(Xt, Yt);
ksperim=-kappa(2)/(Cost*1.07e-5);
##
```

```
## Initial values for I0, K (SEK) and KA (TAi4):
I0=Iexp(1,1);
Kstart=1e4;
KAstart=1e10;
##
dati=0;
dati(1:size(G1,1),1)=G1(1:size(G1,1),1);
dati(1:size(G1,1),2)=CT;
dati(1:size(G1,1),3)=Cost;
dati(1:size(G1,1),4)=ksperim;
dati(1:size(G1,1),5)=ser;
dati(1:size(G1,1),6)=Iexp(1:size(G1,1),1);
X=0;
Y=0;
beta0=0;
X(1:size(dati,1),1:4)=dati(1:size(dati,1),1:4);
X(1:size(dati,1),5)=1;
Y(1:size(dati,1),1)=dati(1:size(dati,1),6);
beta0(1,1)=Kstart;
beta0(2,1)=KAstart;
beta0(3,1)=I0;
save Start dati beta0
beta1=0;
## Non linear optimization:
beta1=nlinfit(X, Y, @SEKTAi4optimfun, beta0)
##
## ------ OUTPUT: ------
## 1) Optimized value of the thermodynamic constants K(SEK) and KA (TAi4)
Kfinal=abs(beta1(1,1))
KAfinal=abs(beta1(2,1))
## 2) Optimized value of sample intensity for G=0 (I0)
I0optim=abs(beta1(3,1))
save Optimized Kfinal KAfinal I0optim
C1=0;
K=Kfinal;
KA=KAfinal;
X0(1)=0;
if CT<1/K
   X0(2)=CT;
else
   X0(2)=1/K;
end
## 3) Distribution of polymeric species, calculated using optimized K (SEK) and KA (TAi4)
##        3.a) monomer concentration:
save tmp_data K KA CT
[C1, fval, info]=fzero(@SEKTAi4distrib, X0);
##        3.b) distribution of oligomers up to i=10000:
A=0;
AA=0;
if or (info==1, info==2, info==3)
   A(1,1)=abs(C1);
   for i=2:10000;
      A(i,1)=Kfinal*A(1,1)*A(i-1,1);
   end
##        3.c) distribution of tetrameric aggregates up to i=2500:
   for i=1:2500
      AA(i,1)=KA*(i^4)*(A(i,1)^4);
   end
   save DISTRIB A AA
   if or(info==2, info==3)
      'WARNING!'
   end
else
   'ERROR! Not converging'
end
```

```
## 4) Resonance intensities calculated from the polymeric distribution
for j=1:size(G1,1)
   sum=0;
   for i=1:size(A,1)
      expo=exp(-dati(j,4)*dati(j,3)*Dcalc(i)*(dati(j,1)^2));
      sum=sum+(A(i,1)*i/dati(j,2))*expo;
   end
   for i=1:size(AA,1)
      expo=exp(-dati(j,4)*dati(j,3)*Dcalc(4*i)*(dati(j,1)^2));
      sum=sum+(AA(i,1)*4*i/dati(j,2))*expo;
   end
   Icalc(j,1)=I0optim*sum;
   LNIcalc(j,1)=log(Icalc(j,1));
end
save IandG G1 G2 Iexp Icalc LNIexp LNIcalc
## 5) Residuals of intensity and monomer concentration
R=0;
for i=1:size(G1,1)
   R=R+(Iexp(i,1)-Icalc(i,1))^2;
end
ResidualI=abs(sqrt(R)/size(G1,1))
monomer=0;
for i=1:size(A,1)
   monomer=monomer+A(i,1)*i;
end
for i=1:size(AA,1)
   monomer=monomer+AA(i,1)*4*i;
end
Residual_monomer=abs(CT-monomer);
save residue ResidualI Residual_monomer
## -------------------------
## ------ FIGURES ------
## FIGURE 1: plot of intensities (I) against gradient field values (G)
## Experimental values: black line, marker +
## Calculated values: red line, marker o
figure(1)
plot(G1,Iexp,'k+-')
hold on
plot(G1,Icalc,'ro-')
## FIGURE 2: plot of logarithm of intensities (ln(I)) against gradient field values
## squared (G^2)
## Experimental values: black line, marker +
## Calculated values: red line, marker o
figure(2)
plot(G2,LNIexp,'k+-')
hold on
plot(G2,LNIcalc,'ro-')
## FIGURE 3: distribution of the polymeric species and the tetrameric aggregates,
## concentration against polymerization index (up to i=50)
## Polymeric species: black line
## Tetrameric aggregates: red line
figure(3)
plot(1:50,A(1:50,1),'k-')
hold on
plot(1:50,AA(1:50,1),'r-')
```

SEKTAi4optimfun.m: Function from the PGSE equation.

```
function I=SEKTAi4optimfun(beta0,X)
K=abs(beta0(1,1));            ## Initial value of the SEK constant
KA=abs(beta0(2,1));           ## Initial value of the TAi4 constant
I0=abs(beta0(3,1));           ## Initial value of the intensity at G=0
G1=0;
G1=X(1:size(X,1),1);          ## Input values of the field gradient
```

```
CT=X(1,2);                    ## Input value of the total concentration of the monomer
Cost=X(1,3);                  ## Input value of PGSE constant
ksperim=X(1,4);               ## Input value of the experimental constant determined by
                              ## linear regression of the internal standard data
I=0;
load Dcalc                    ## Hydrodynamic translational diffusion coefficients
save tmp_data K KA CT
X0(1)=0;
if CT<1/K
   X0(2)=CT;
else
   X0(2)=1/K;
end
[C1,fval,info]=fzero(@SEKTAi4distrib,X0);         ## Concentration of the monomer
                                                  ## according to SEKTAi4 distribution
if or(info==1,info==2,info==3)
   A(1,1)=abs(C1);
   for i=2:10000;
      A(i,1)=K*A(1,1)*A(i-1,1);            ## Concentration of the polymeric species
   end
   for i=1:2500
      AA(i,1)=KA*(i^4)*(A(i,1)^4);         ## Concentration of the tetrameric aggregates
   end
   save SEKTAi4 A AA
   if or(info==2,info==3)
      'WARNING!'
   end
else
   'ERROR! Not converging'
   load SEKTAi4
end
for j=1:size(G1,1)
   sum=0;
   for i=1:10000
      expo=exp(-ksperim*Cost*Dcalc(i)*(G1(j,1)^2));
      sum=sum+(A(i,1)*i/CT)*expo;
   end
   for i=1:size(AA,1)
      expo=exp(-ksperim*Cost*Dcalc(4*i)*(G1(j,1)^2));
      sum=sum+(AA(i,1)*4*i/CT)*expo;
   end
   I(j,1)=I0*sum;
end
A=0;
AA=0;
```

SEKTAi4distrib.m: Function for the calculation of the monomer concentration from defined values of total monomer concentration and polymerization constants according to the SEK-TAi^4 model.

```
function y=SEKTAi4distrib(x)
load tmp_data
y=CT*((1-(K*x)^4)^6)-x*((1-K*x)^4)*((1+K*x)^6)*((1+(K*x)^2)^6)-
4*KA*(x^4)*(((K*x)^16)+26*((K*x)^12)+66*((K*x)^8)+26*((K*x)^4)+1);
```

2. Supplementary Figures

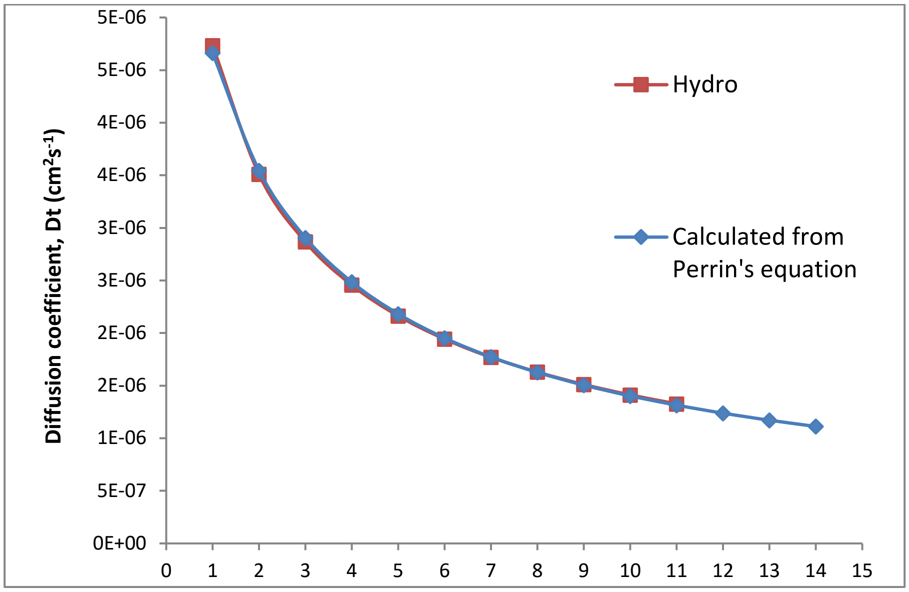

Figure S1. Comparison between the translational diffusion coefficients derived from HYDRO and those calculated using Perrin's equations.

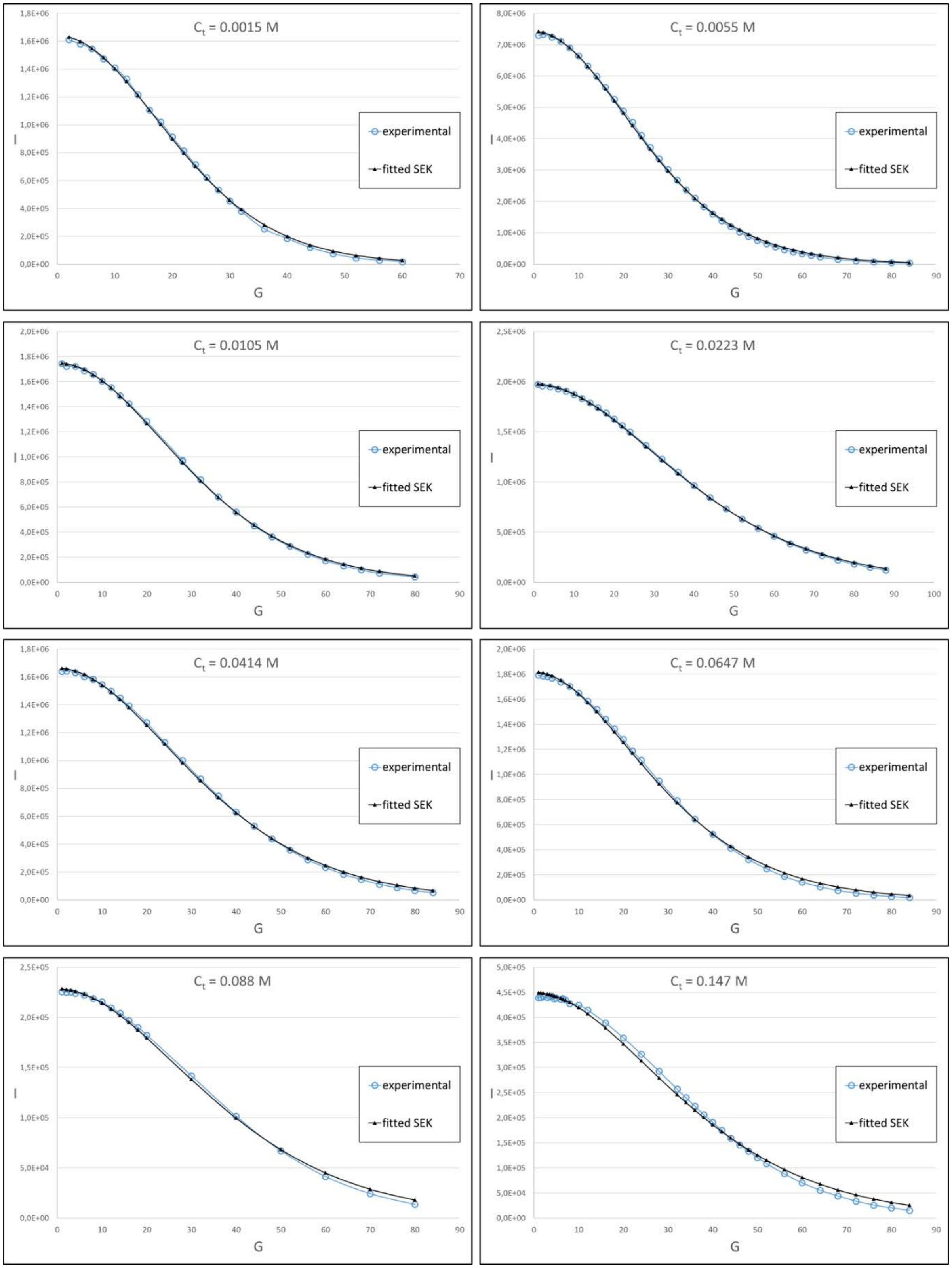

Figure S2: Experimental data (blue circles) and fitted data (black triangles) using the SEK model for **1**I at different concentrations in $CDCl_3$. Data are reported as intensity of the sample signal, I, against gradient field, G.

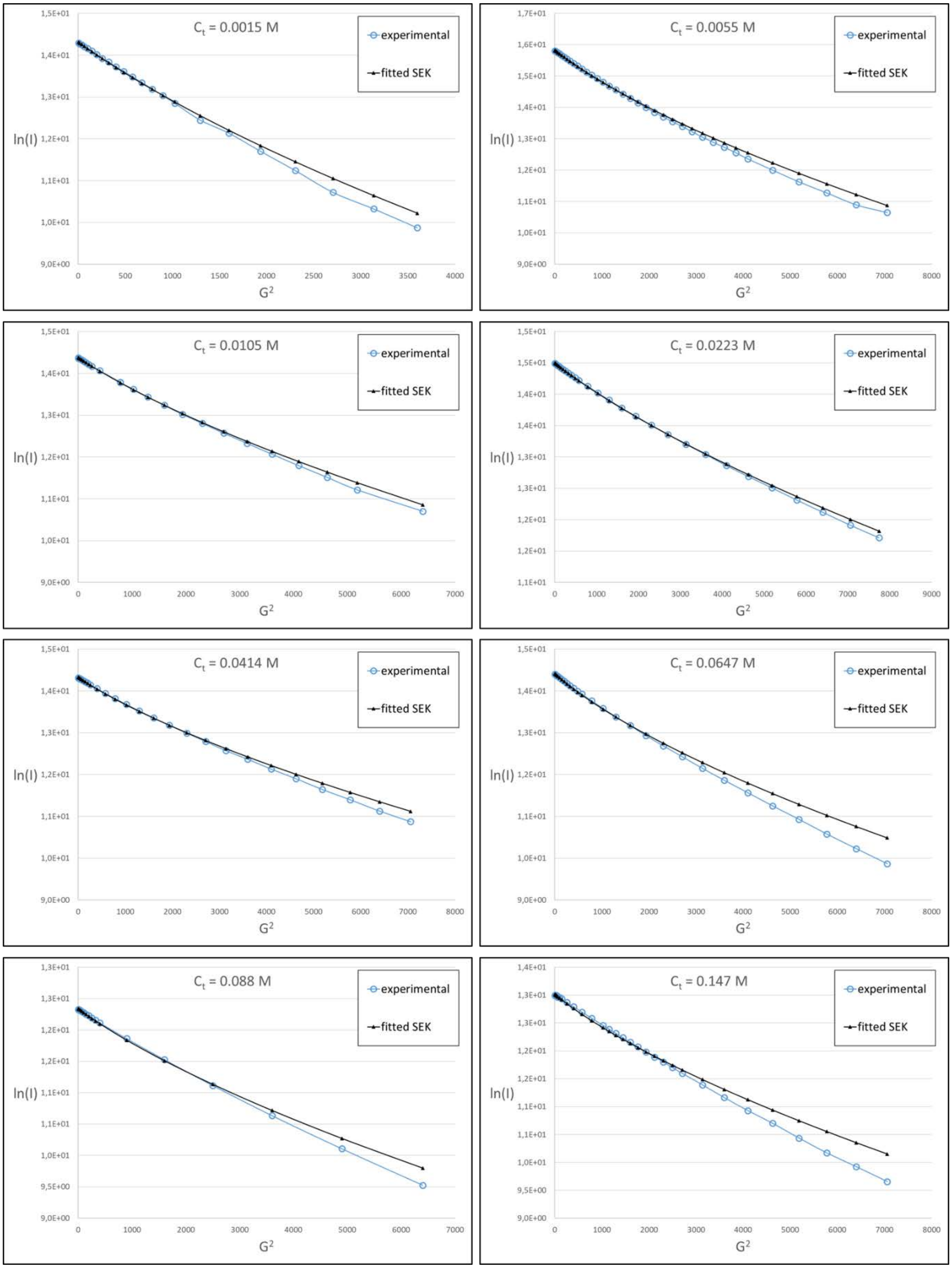

Figure S3: Experimental data (blue circles) and fitted data (black triangles) using the SEK model for **1**I at different concentrations in $CDCl_3$. Data are reported as logarithm of the intensity of the sample signal, ln(I), against squared gradient field, G^2.

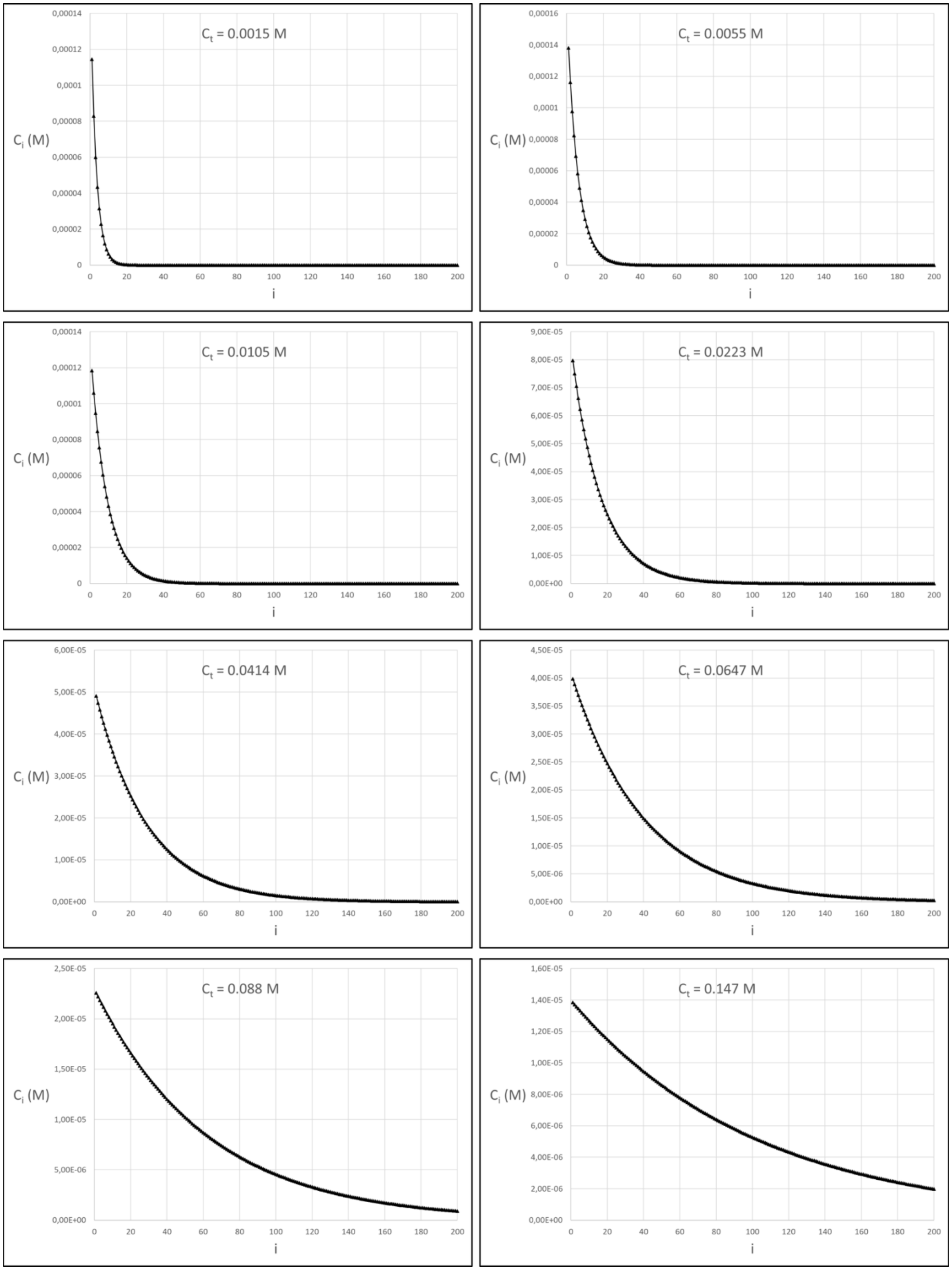

Figure S4: Polymeric distributions for each concentration, obtained using the SEK model, with a K_{SEK} value optimized for each series of data to fit the experimental PGSE measurements.

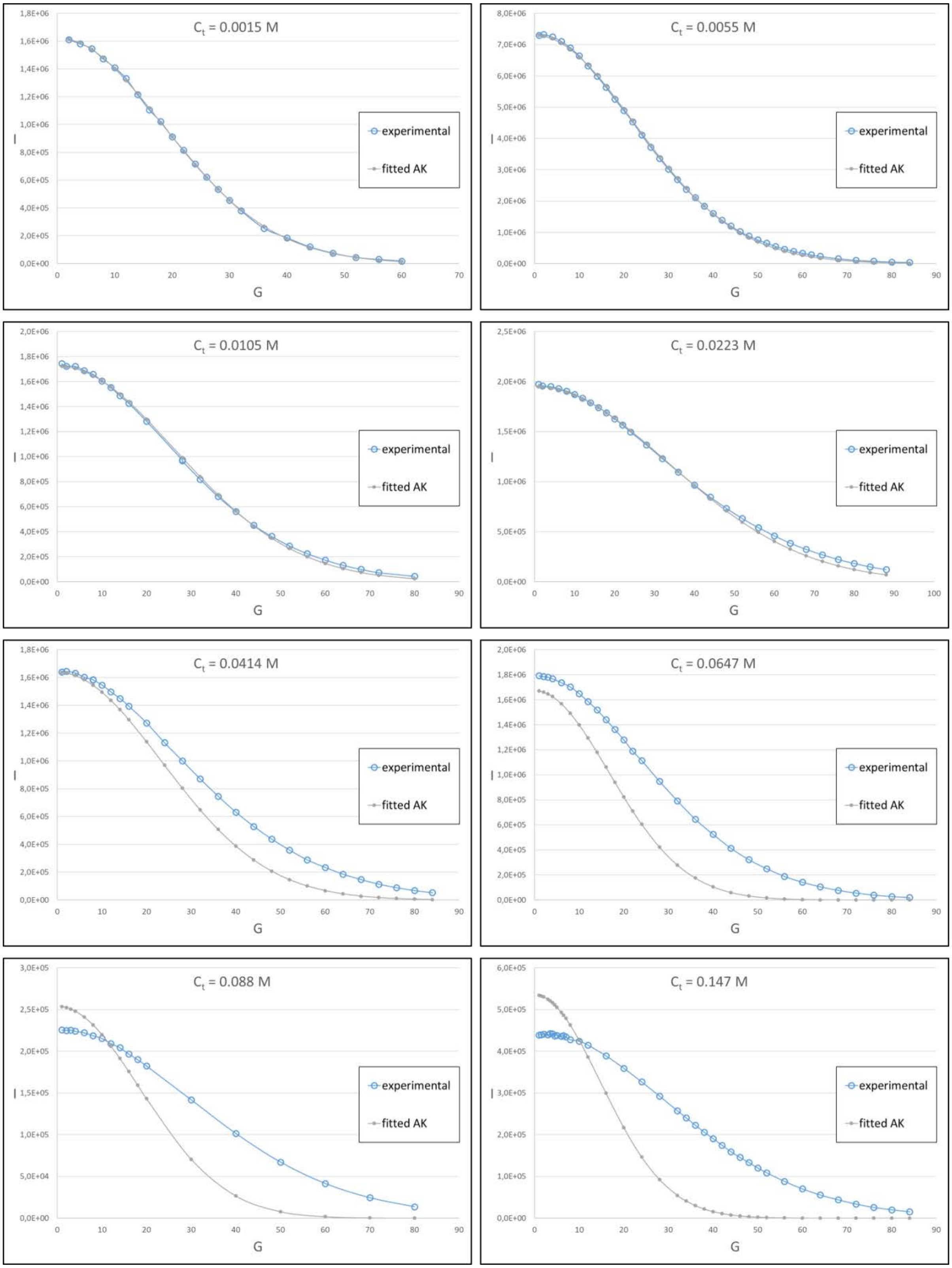

Figure S5: Experimental data (blue circles) and fitted data (grey circles) using the AK model for **1**I at different concentrations in $CDCl_3$. Data are reported as intensity of the sample signal, I, against gradient field, G.

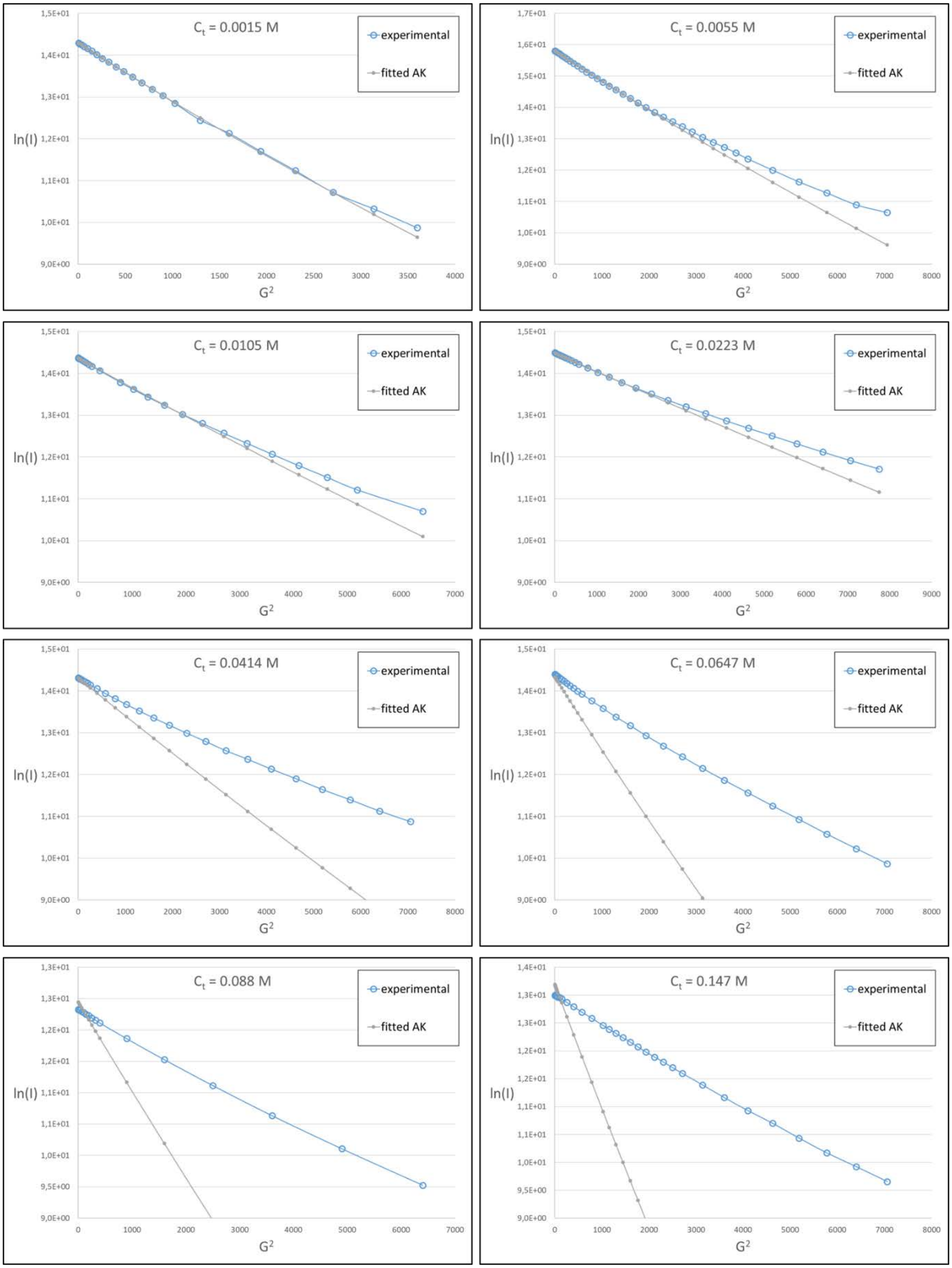

Figure S6: Experimental data (blue circles) and fitted data (grey circles) using the AK model for **1**I at different concentrations in $CDCl_3$. Data are reported as logarithm of the intensity of the sample signal, ln(I), against squared gradient field, G^2.

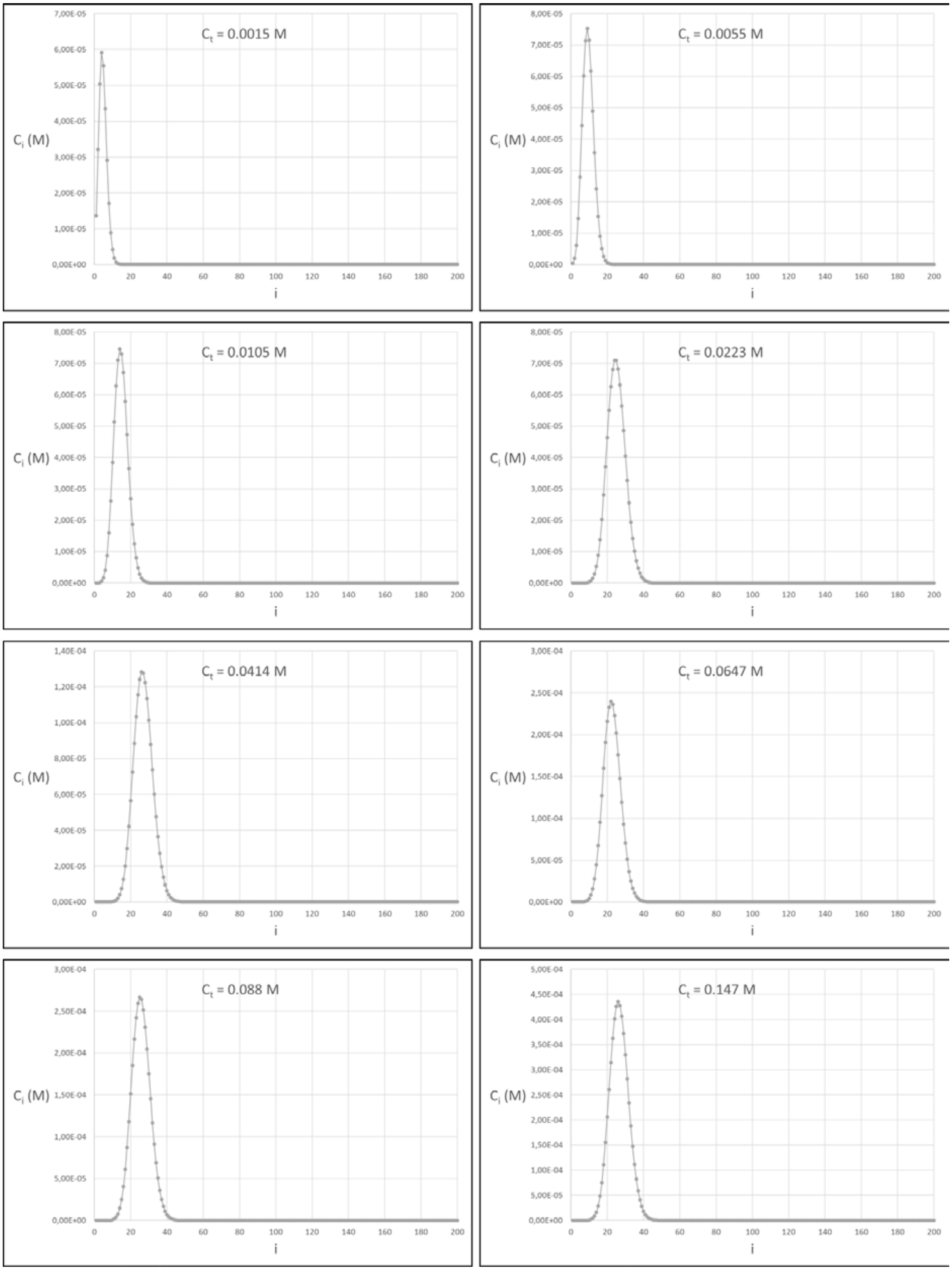

Figure S7: Polymeric distributions for each concentration, obtained using the AK model, with a K_{AK} value optimized for each series of data to fit the experimental PGSE measurements.

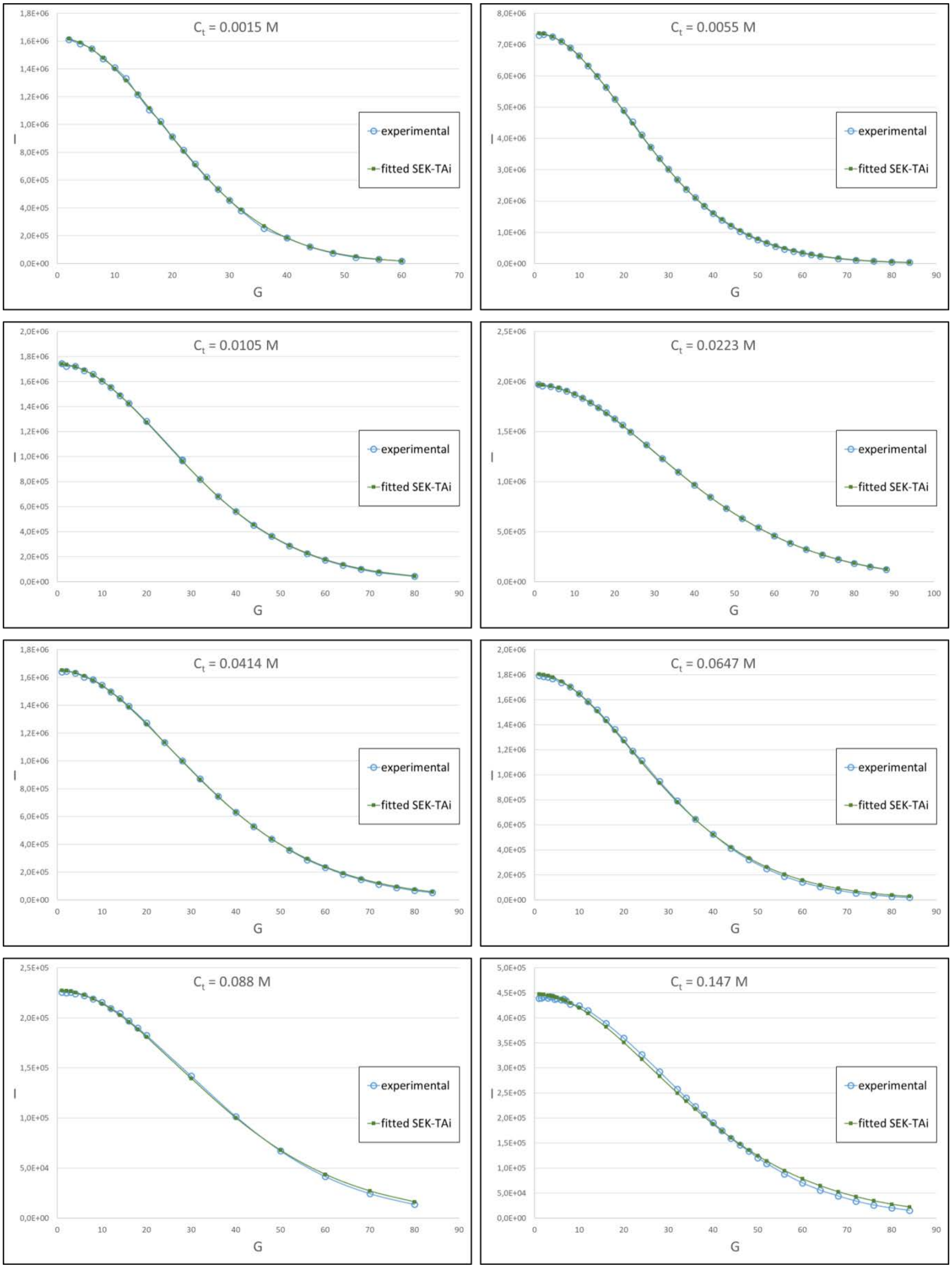

Figure S8: Experimental data (blue circles) and fitted data (green squares) using the SEK-TAi model for **1**I at different concentrations in $CDCl_3$. Data are reported as intensity of the sample signal, I, against gradient field, G.

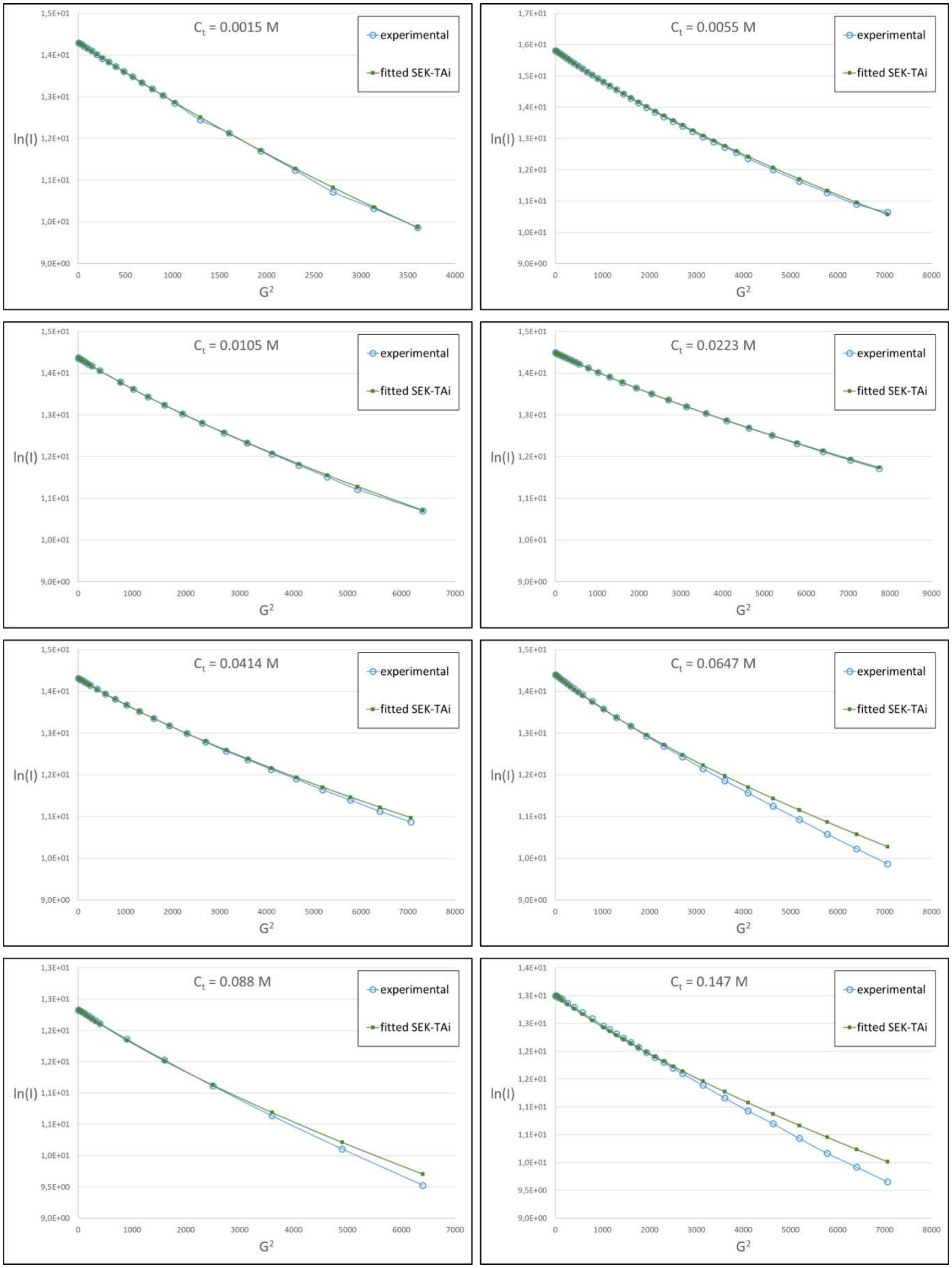

Figure S9: Experimental data (blue circles) and fitted data (green squares) using the SEK-TAi model for **1**I at different concentrations in $CDCl_3$. Data are reported as logarithm of the intensity of the sample signal, ln(I), against squared gradient field, G^2.

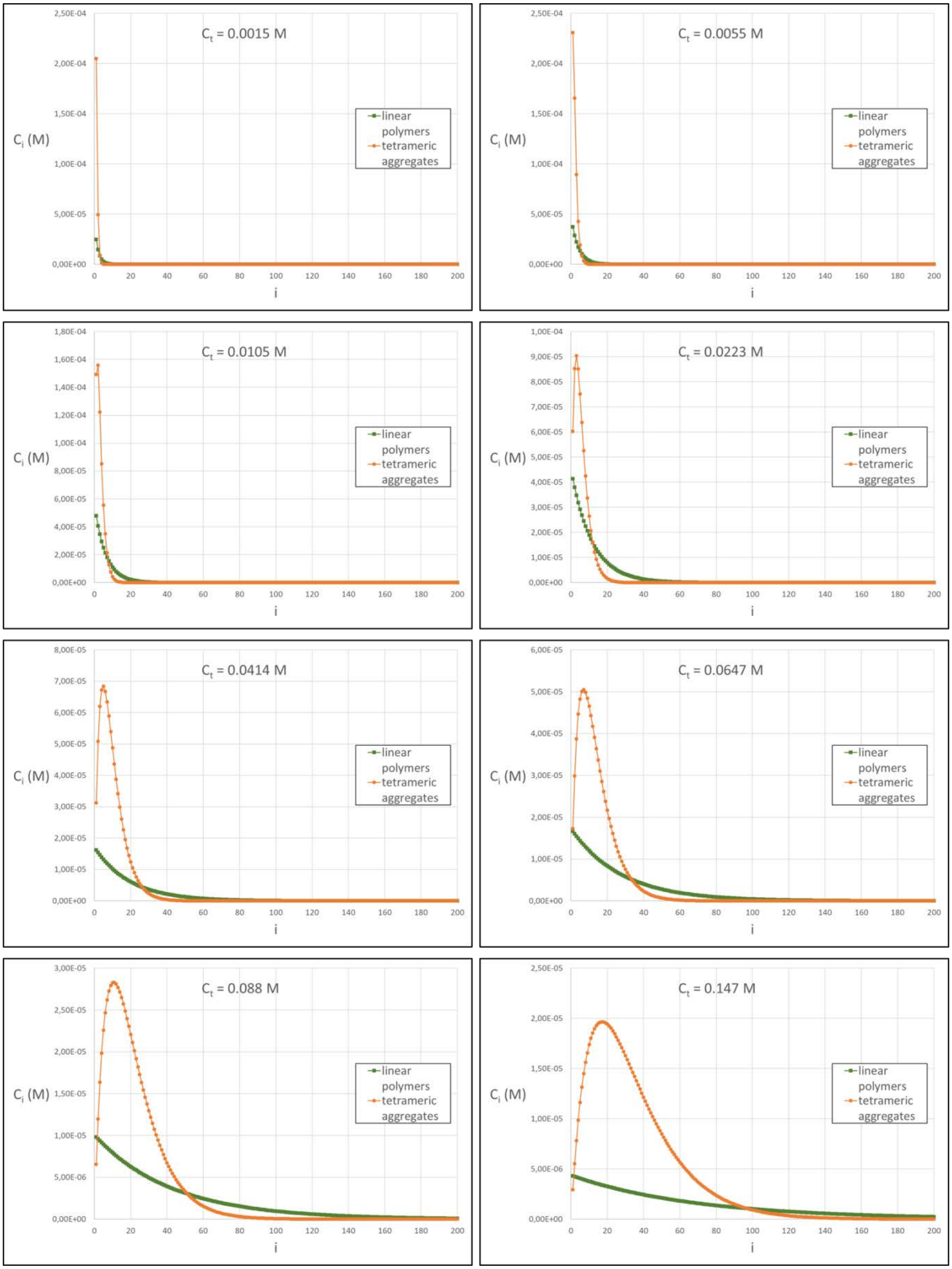

Figure S10: Polymeric (green squares) and aggregate distributions (orange circles) for each concentration, obtained using the SEK-TAi model, with K_{SEK} and K_{agg} values optimized for each series of data to fit the experimental PGSE measurements.

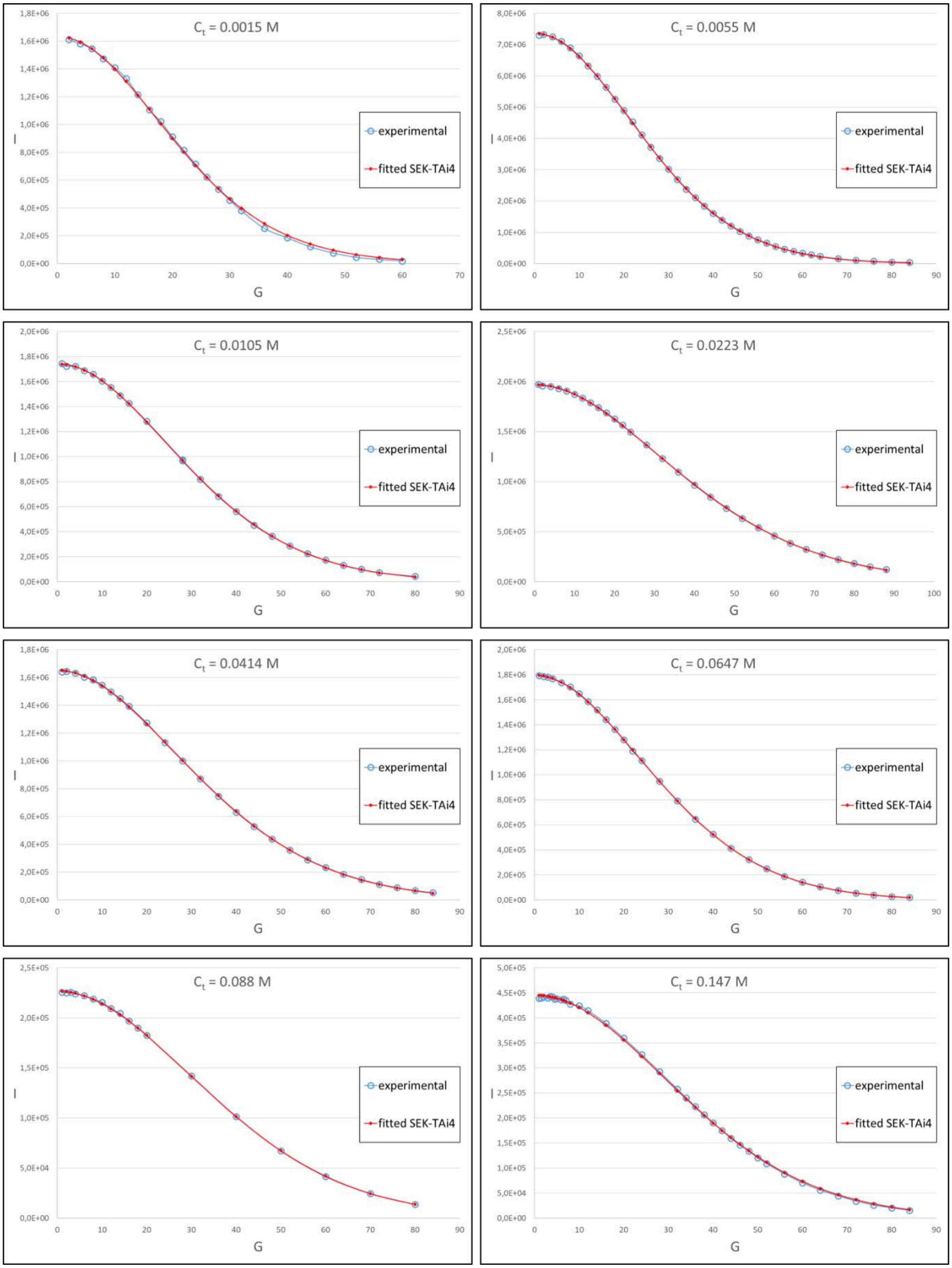

Figure S11: Experimental data (blue circles) and fitted data (red diamonds) using the SEK-TAi[4] model for **1**I at different concentrations in $CDCl_3$. Data are reported as intensity of the sample signal, I, against gradient field, G.

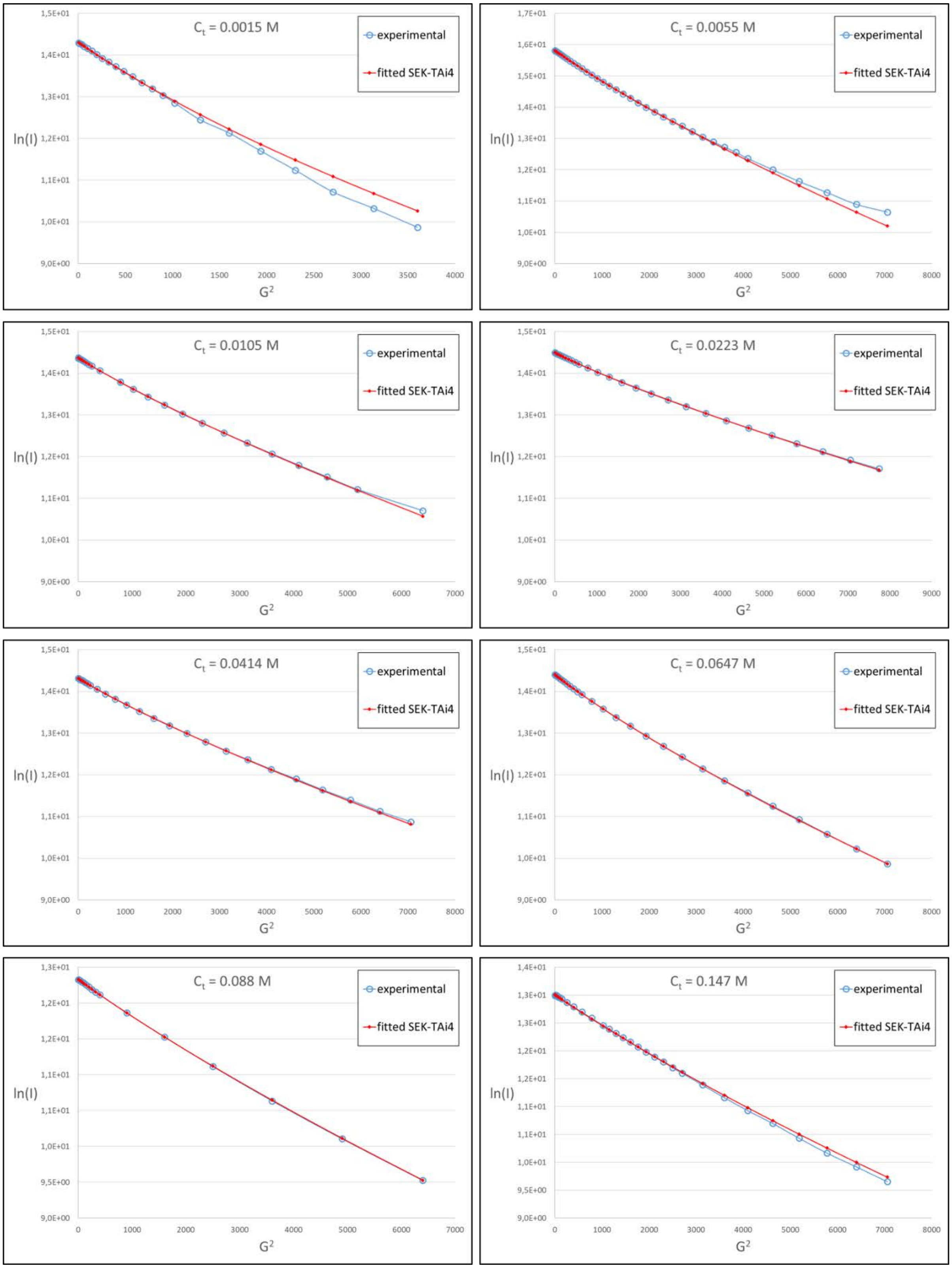

Figure S12: Experimental data (blue circles) and fitted data (red diamonds) using the SEK-TAi[4] model for **1**I at different concentrations in $CDCl_3$. Data are reported as logarithm of the intensity of the sample signal, ln(I), against squared gradient field, G^2.

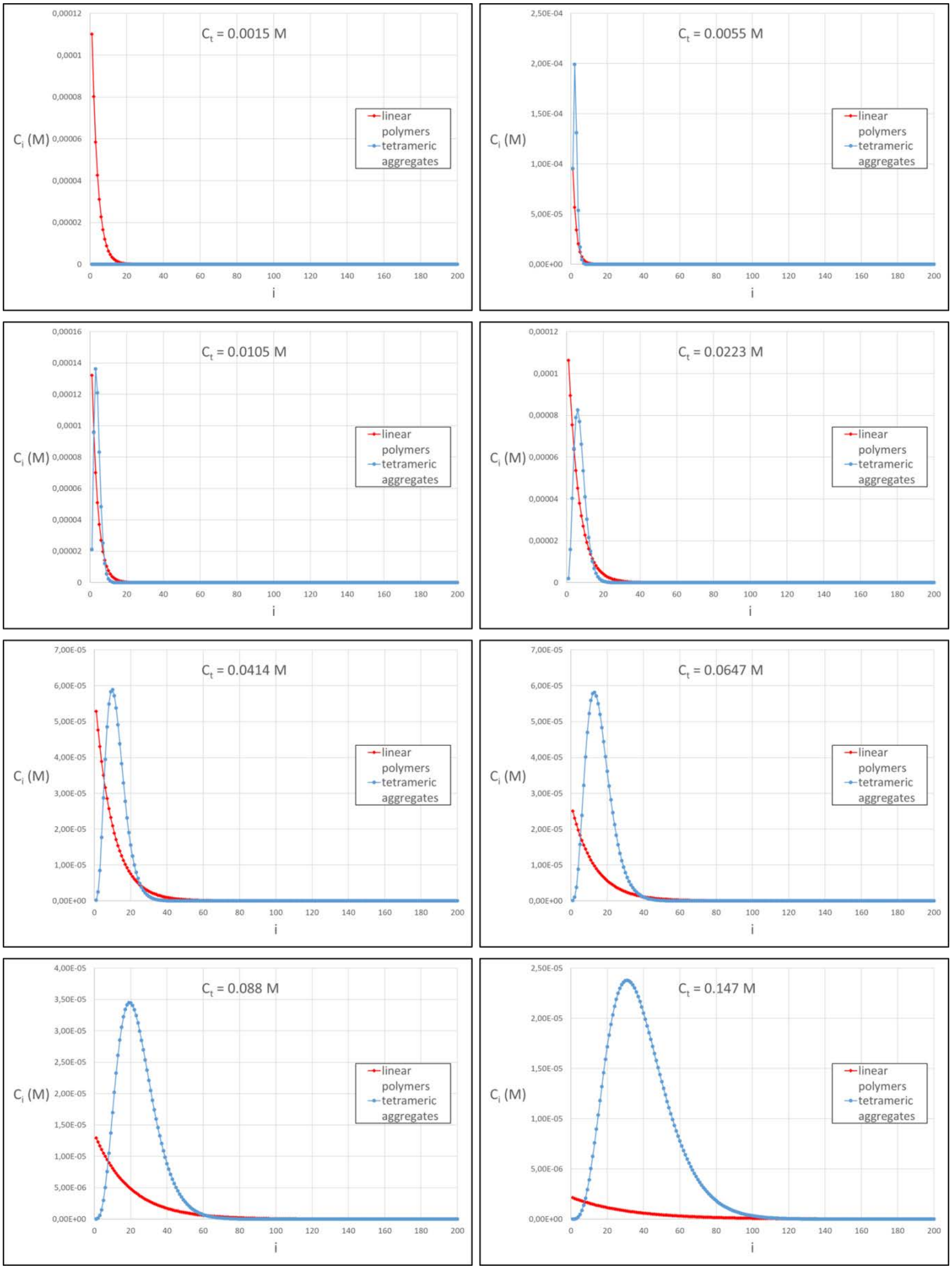

Figure S13: Polymeric (red diamonds) and aggregate distributions (blue circles) for each concentration, obtained using the SEK-TAi[4] model, with K_{SEK} and K_{agg} values optimized for each series of data to fit the experimental PGSE measurements.

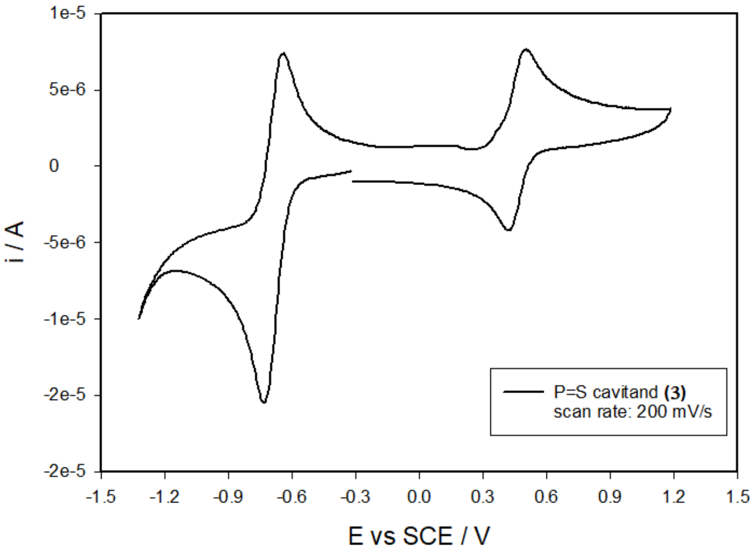

Figure S14: Cyclic voltammogram for the first reduction of the methylpyridinium unit of compound **3**$^+$ (0.732 mM, CH_2Cl_2 with 73.2 mM TBAP, glassy carbon electrode, 200 mV/s). The reversible wave at +0.46 V vs SCE is that of ferrocene used as an internal standard.

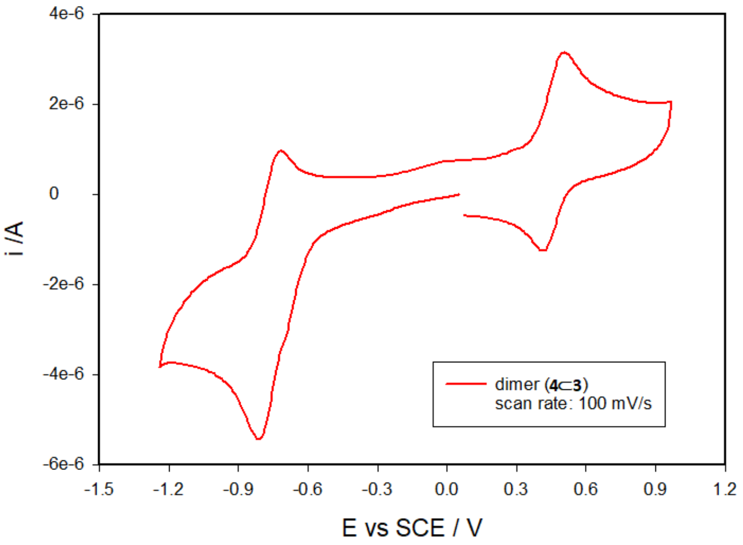

Figure S15: Cyclic voltammogram for the first reduction of the MePy$^+$ unit of compound **3**$^+$ 0.512 mM in the presence of the cavitand **4** 0.670 mM(CH_2Cl_2 with 67.1 mM TBAP, glassy carbon electrode, 100 mV/s). The reversible wave at +0.46 V vs SCE is that of ferrocene used as an internal standard.

3. Supplementary Tables

Table S1: Values of the thermodynamic constants for four polymerization models (SEK, AK, SEK-TAi, SEK-TAi^4), optimized with PGSE data obtained at different monomer concentrations, C_t. For the simple polymerization models (SEK and AK) values of the polymerization constants (K_{SEK} and K_{AK}, respectively) are shown, for polymerization-aggregation models (SEK-TAi and SEK-TAi^4) the polymerization (K_{SEK}) and aggregation constants (K_{agg}) are reported.

C_t	**SEK model**	**AK model**	**SEK-TAi model**		**SEK-TAi^4 model**	
	K_{SEK} (M^{-1})	**K_{AK}** (M^{-1})	**K_{SEK}** (M^{-1})	**K_{agg}** (M^{-3})	**K_{SEK}** (M^{-1})	**K_{agg}** (M^{-3})
1.5 mM	6,32E+03	3,43E+05	2,38E+04	5,40E+14	6,63E+03	1,52E+05
5.5 mM	6,10E+03	2,30E+07	2,08E+04	1,19E+14	6,34E+03	1,19E+12
10.5 mM	7,55E+03	3,35E+09	1,77E+04	2,82E+13	5,51E+03	6,95E+10
22.3 mM	1,18E+04	8,13E+13	2,22E+04	2,06E+13	7,93E+03	1,54E+10
41.1 mM	1,97E+04	2,77E+14	5,85E+04	4,48E+14	1,71E+04	3,04E+10
64.7 mM	2,45E+04	2,41E+12	5,79E+04	2,24E+14	3,70E+04	2,27E+11
88 mM	4,36E+04	1,28E+14	9,95E+04	7,09E+14	7,34E+04	3,89E+11
147 mM	7,15E+04	6,07E+13	2,29E+05	8,59E+15	4,55E+05	6,13E+13

Table S2: Average degree of polymerization calculated for polymer (and aggregate) distributions obtained with different polymerization (and aggregation) models, for increasing starting concentrations of monomer. The thermodynamic constants used in the calculation are those obtained from the fitting of each series of PGSE data (Table S1).

C_t	**SEK model**	**AK model**	**SEK-TAi model**	**SEK-TAi^4 model**
1.5 mM	3,62	4,74	4,61	3,69
5.5 mM	6,31	9,50	7,56	7,43
10.5 mM	9,42	14,69	10,77	10,10
22.3 mM	16,72	25,01	18,60	17,26
41.1 mM	29,04	26,86	33,88	32,41
64.7 mM	40,28	22,65	46,03	50,59
88 mM	62,44	25,71	70,88	79,07
147 mM	103,04	26,56	120,87	146,10