# Deconvolution of a Multi-Component Interaction Network Using Systems Chemistry – Supporting Information

*Soumyadip Ghosh*[†], *Pritam Mukhopadhyay*[‡], *Lyle Isaacs*[†,*]

[†]*Department of Chemistry and Biochemistry, University of Maryland, College Park, MD 20742, and* [‡]*School of Physical Sciences, Jawaharlal Nehru University, New Delhi 110067, India.*

**Table of Contents** **Pages**

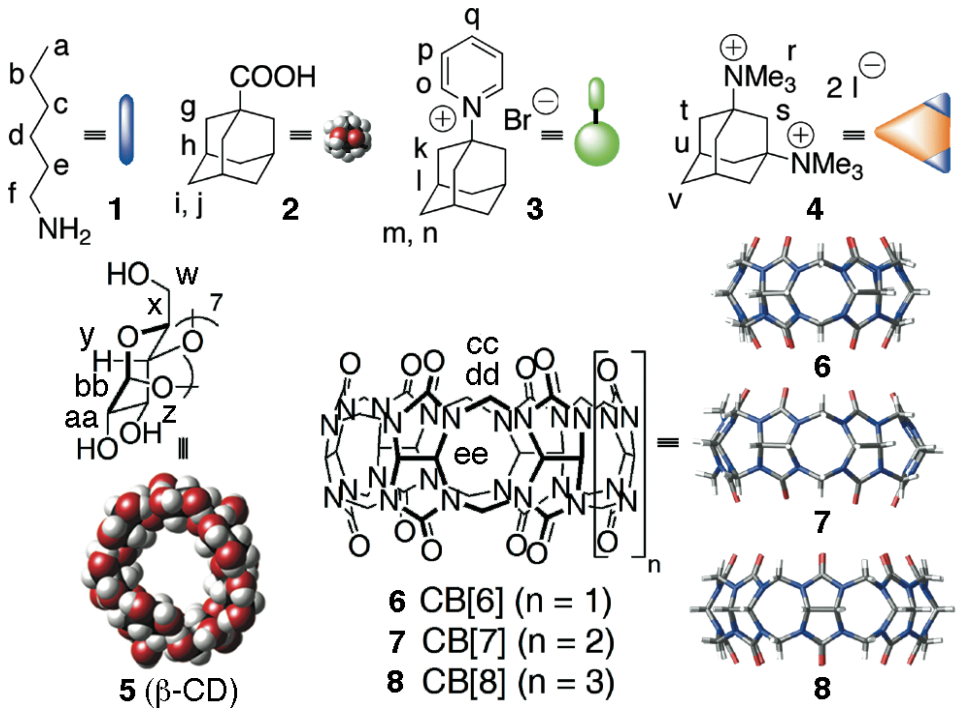



***Figure S1.*** Structures of the compounds used in the study.

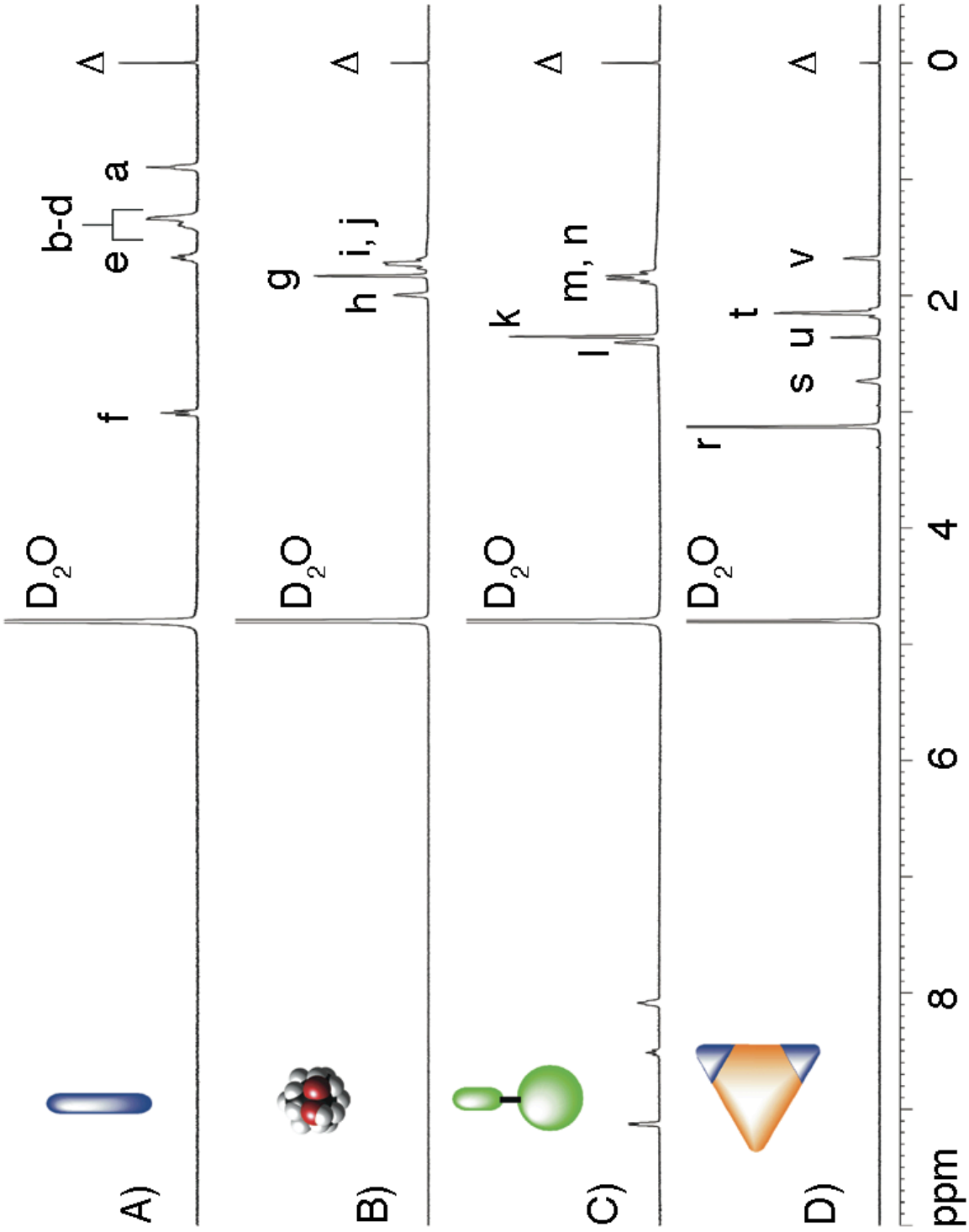



***Figure S2.*** $^{1}$H NMR spectra (400 MHz, $D_2O$, pD 7.4, 298 K, 1 mM) recorded for: A) **1**, B) **2**, C) **3**, and D) **4**. $(CD_3)_3SiCD_2CD_2CO_2D$ (Δ) is used as internal standard.

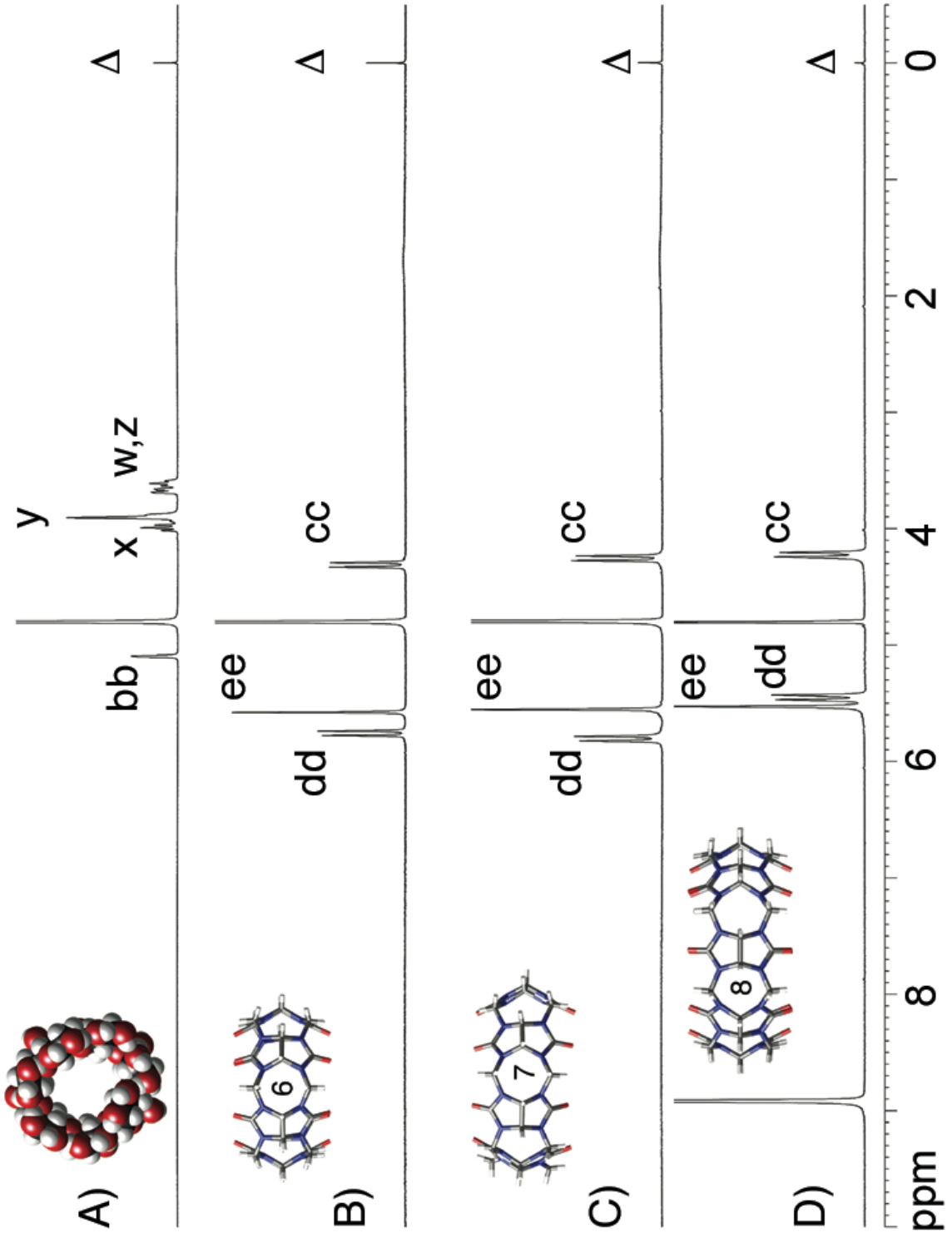



***Figure S3.*** $^{1}$H NMR spectra (400 MHz, $D_2O$, pD 7.4, 298 K, 1 mM) recorded for: A) **5**, B) **6**, C) **7**, and D) **8** (400 MHz, $D_2O$/DCl (1:1), 298 K, 1 mM). $(CD_3)_3SiCD_2CD_2CO_2D$ (Δ) is used as internal standard.

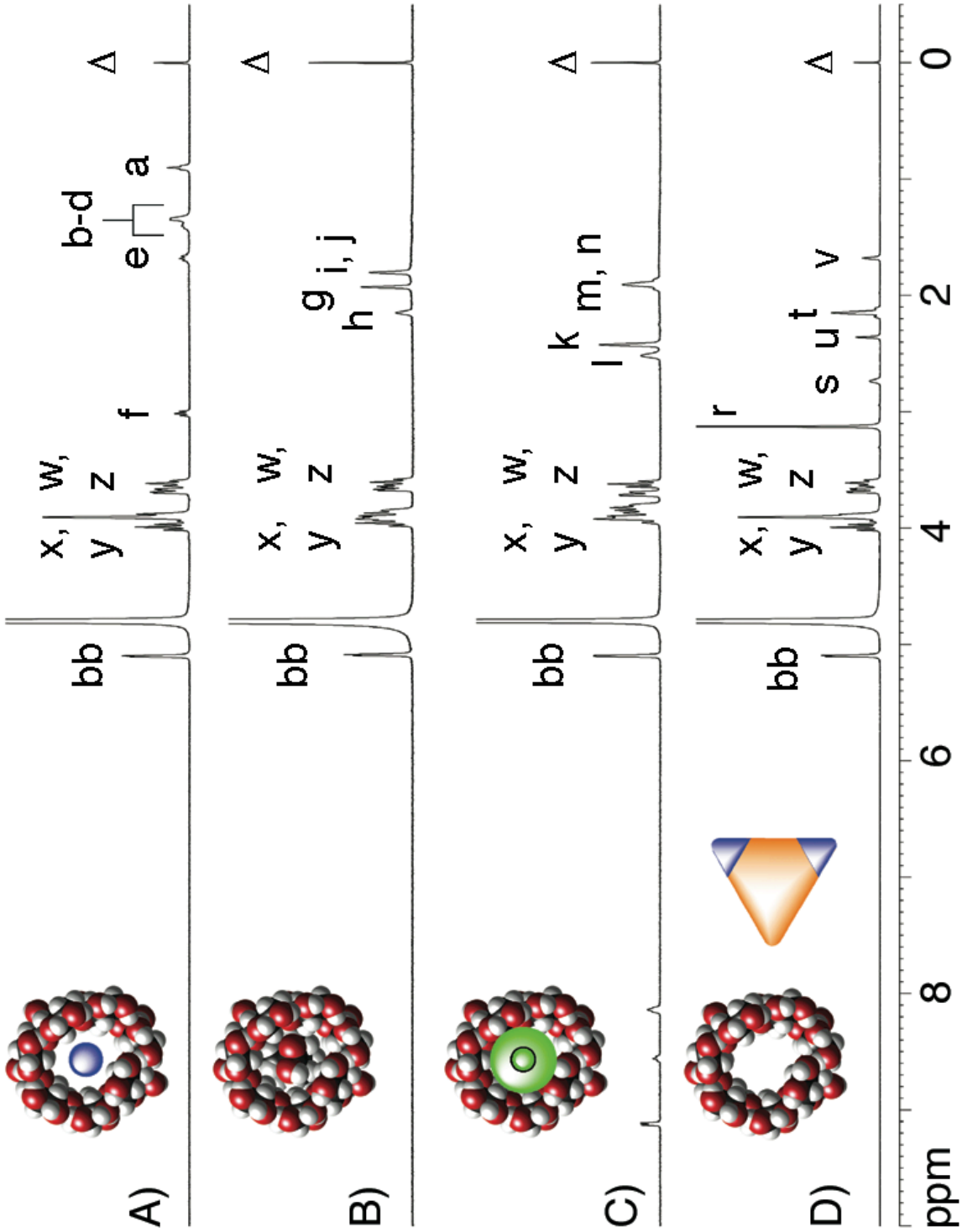



***Figure S4.*** $^{1}$H NMR spectra (400 MHz, $D_2O$, pD 7.4, 298 K, 1 mM) recorded for an equimolar mixture of: A) **5** and **1**, B) **5** and **2**, C) **5** and **3**, and D) **5** and **4**. $(CD_3)_3SiCD_2CD_2CO_2D$ (Δ) is used as internal standard.

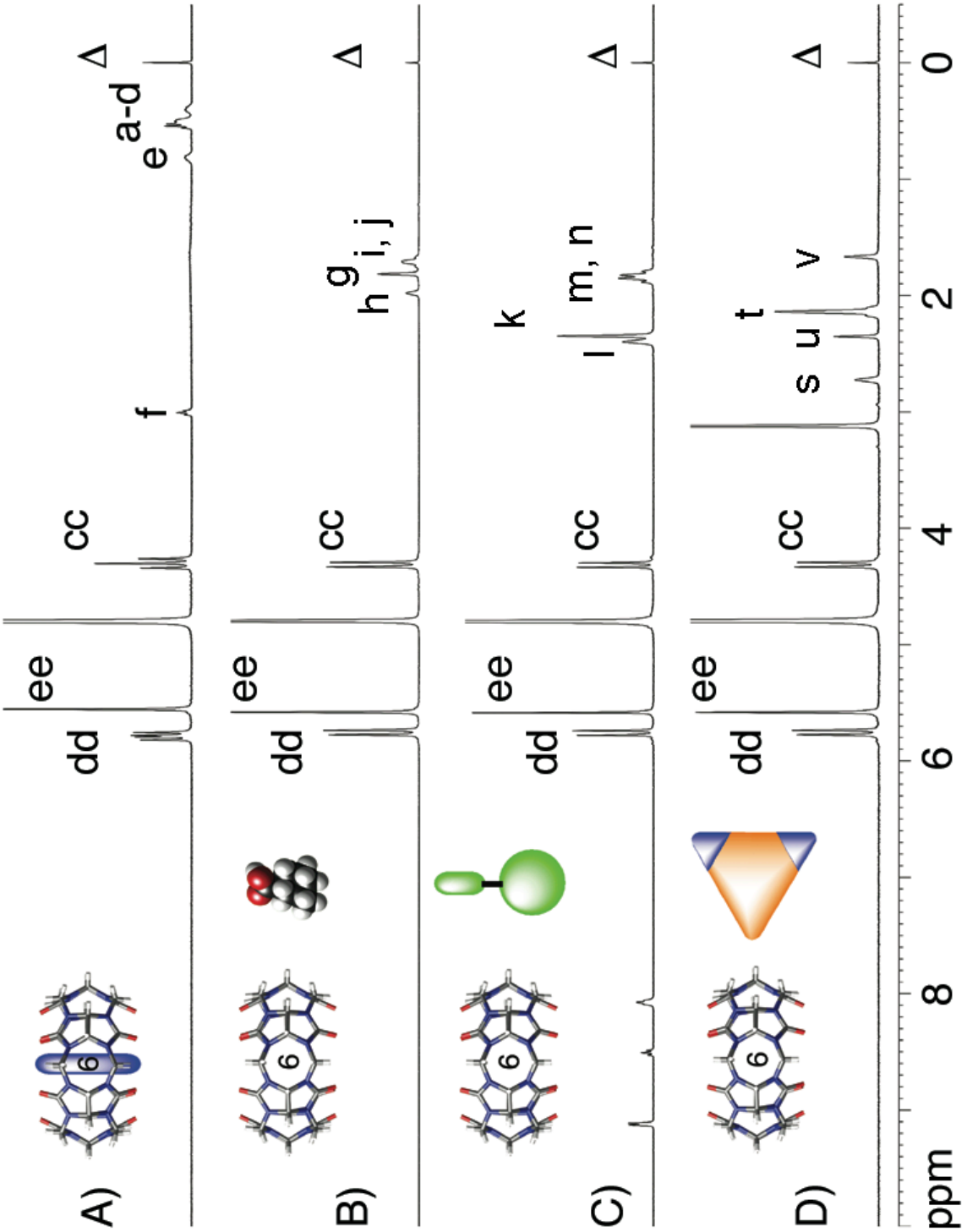



***Figure S5.*** $^{1}$H NMR spectra (400 MHz, $D_2O$, pD 7.4, 298 K, 1 mM) recorded for an equimolar mixture of: A) **6** and **1**, B) **6** and **2**, C) **6** and **3**, and D) **6** and **4**. $(CD_3)_3SiCD_2CD_2CO_2D$ (Δ) is used as internal standard.

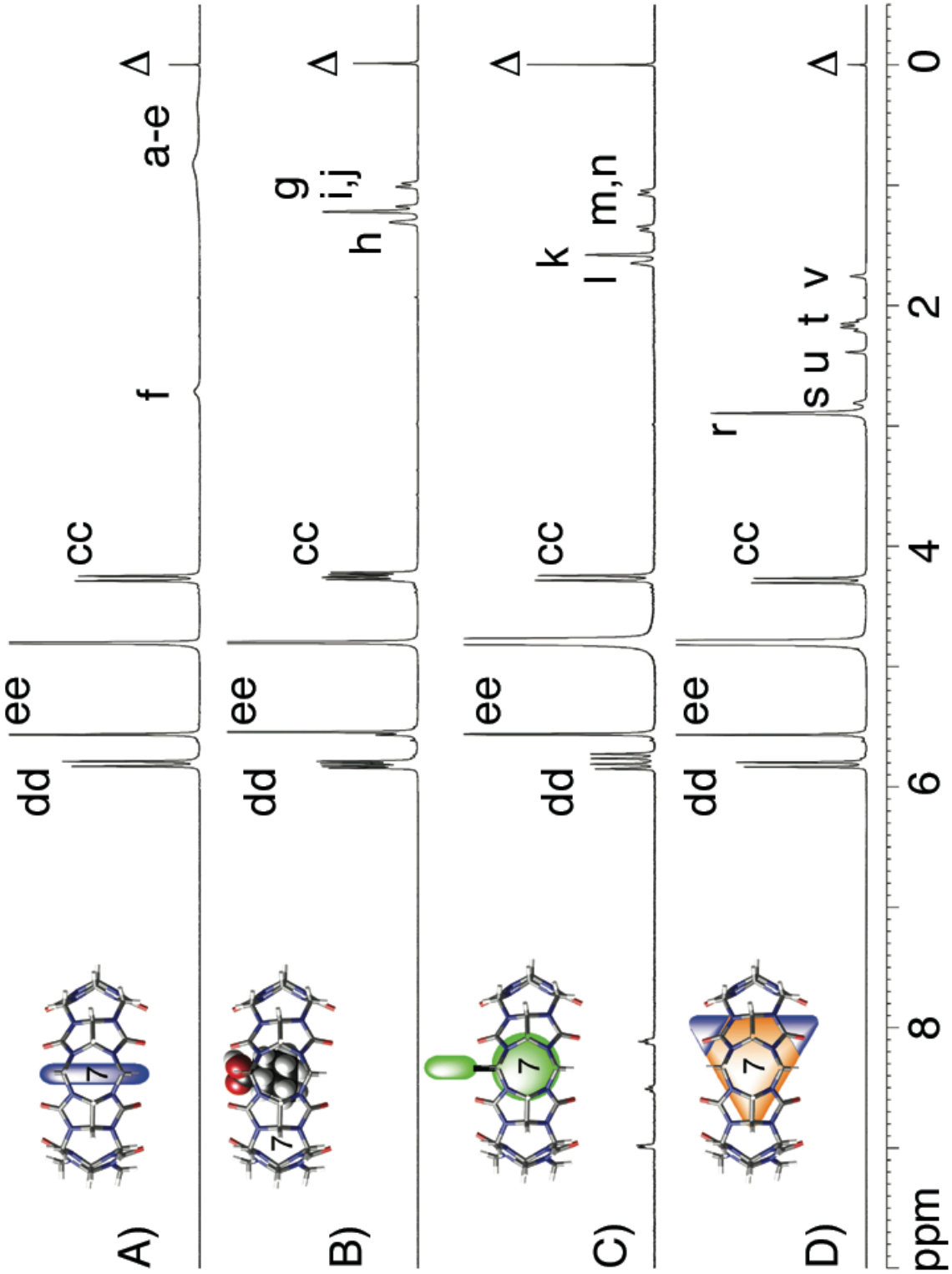



***Figure S6.*** $^{1}$H NMR spectra (400 MHz, $D_2O$, pD 7.4, 298 K, 1 mM) recorded for an equimolar mixture of: A) **7** and **1**, B) **7** and **2**, C) **7** and **3**, and D) **7** and **4**. $(CD_3)_3SiCD_2CD_2CO_2D$ (Δ) is used as internal standard.

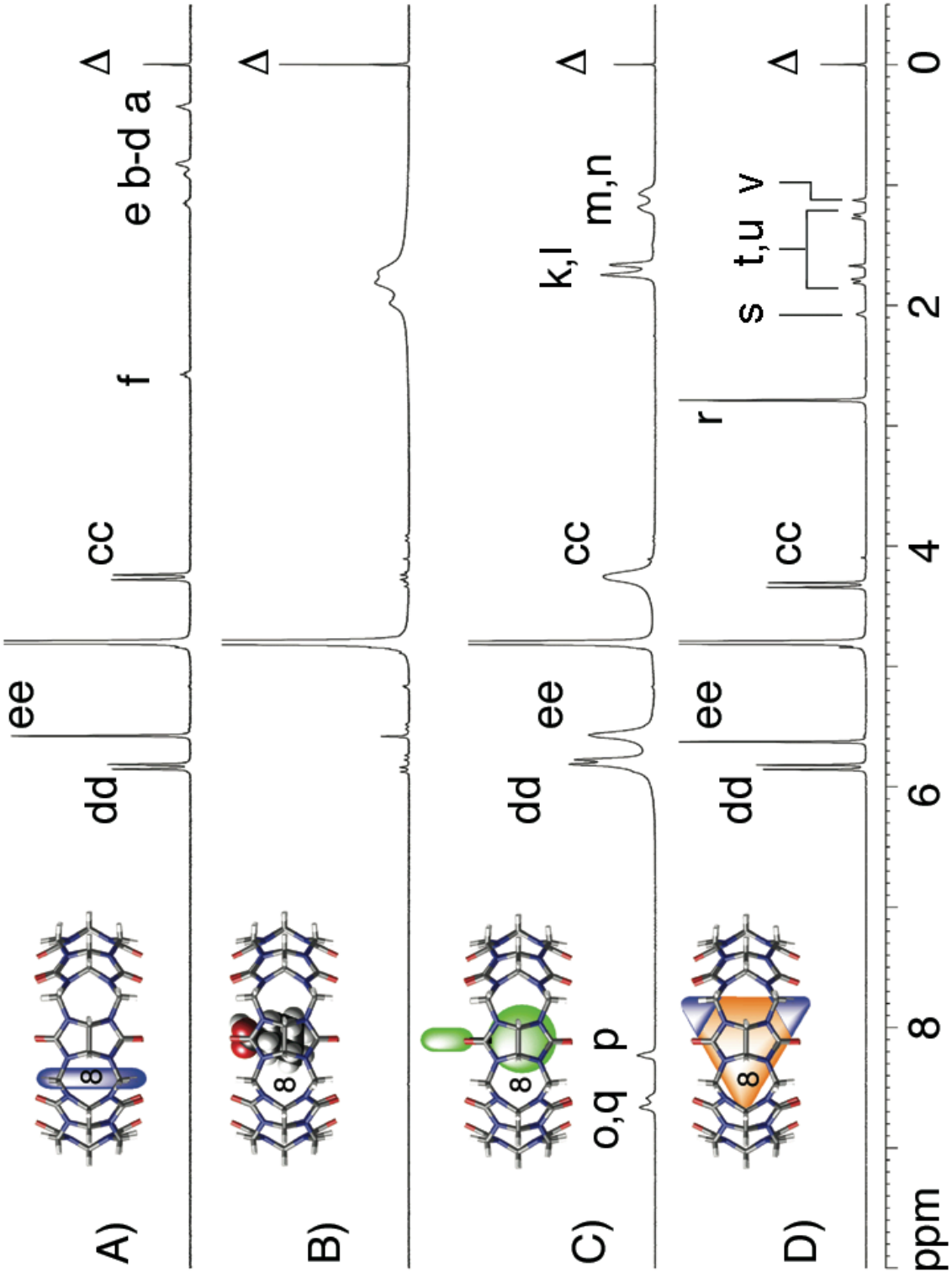



***Figure S7.*** $^{1}$H NMR spectra (400 MHz, $D_2O$, pD 7.4, 298 K, 1 mM) recorded for an equimolar mixture of: A) **8** and **1**, B) **8** and **2**, C) **8** and **3**, and D) **8** and **4**. $(CD_3)_3SiCD_2CD_2CO_2D$ (Δ) is used as internal standard.

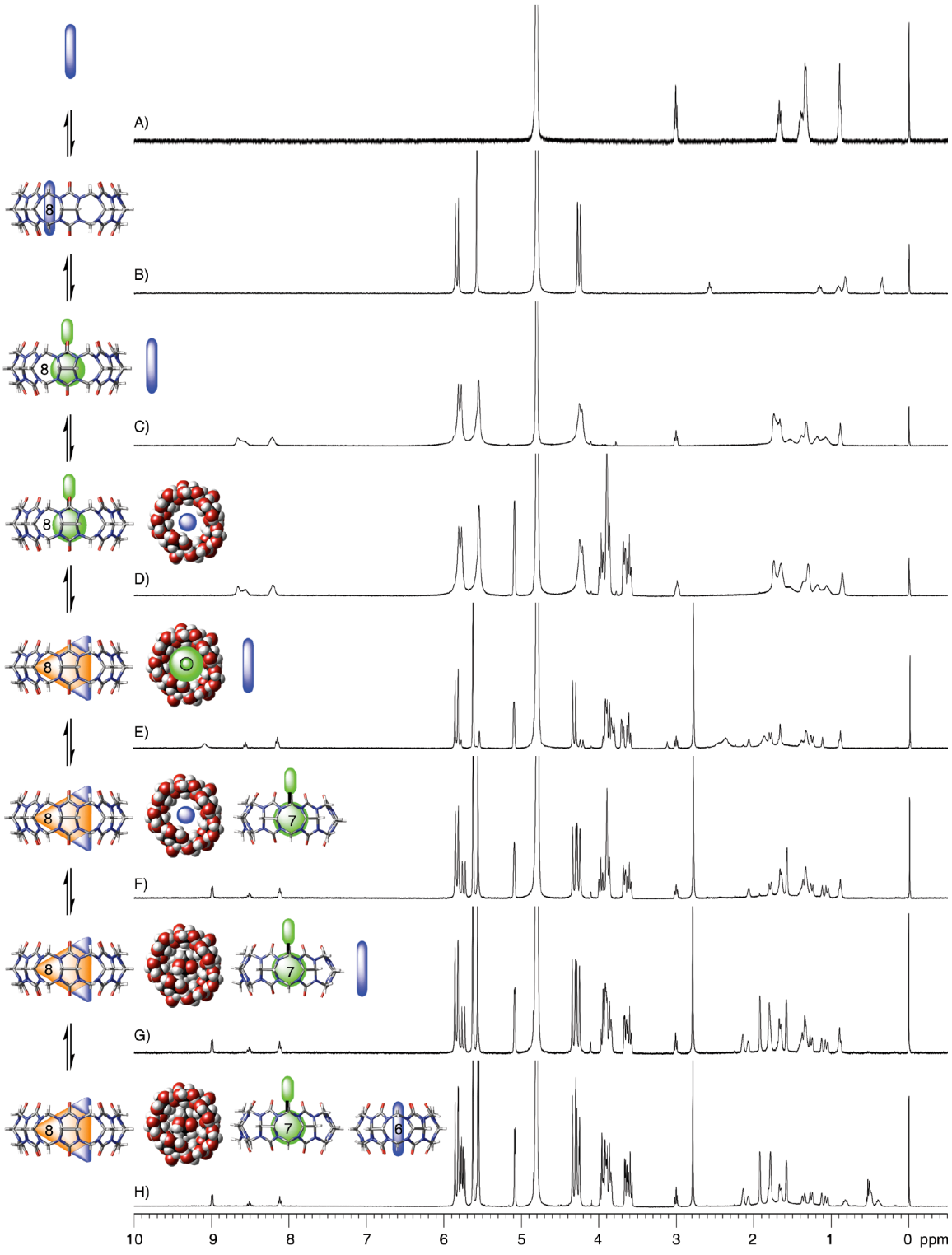



***Figure S8.*** $^{1}$H NMR spectra (400 MHz, $D_2O$, pD 7.4, 298 K, 1 mM) recorded for an equimolar mixture after addition of 1 eq. of: A) **1**, B) **8**, C) **3**, D) **5**, E) **4,** F) **7**, G) **2**, H) **6**.

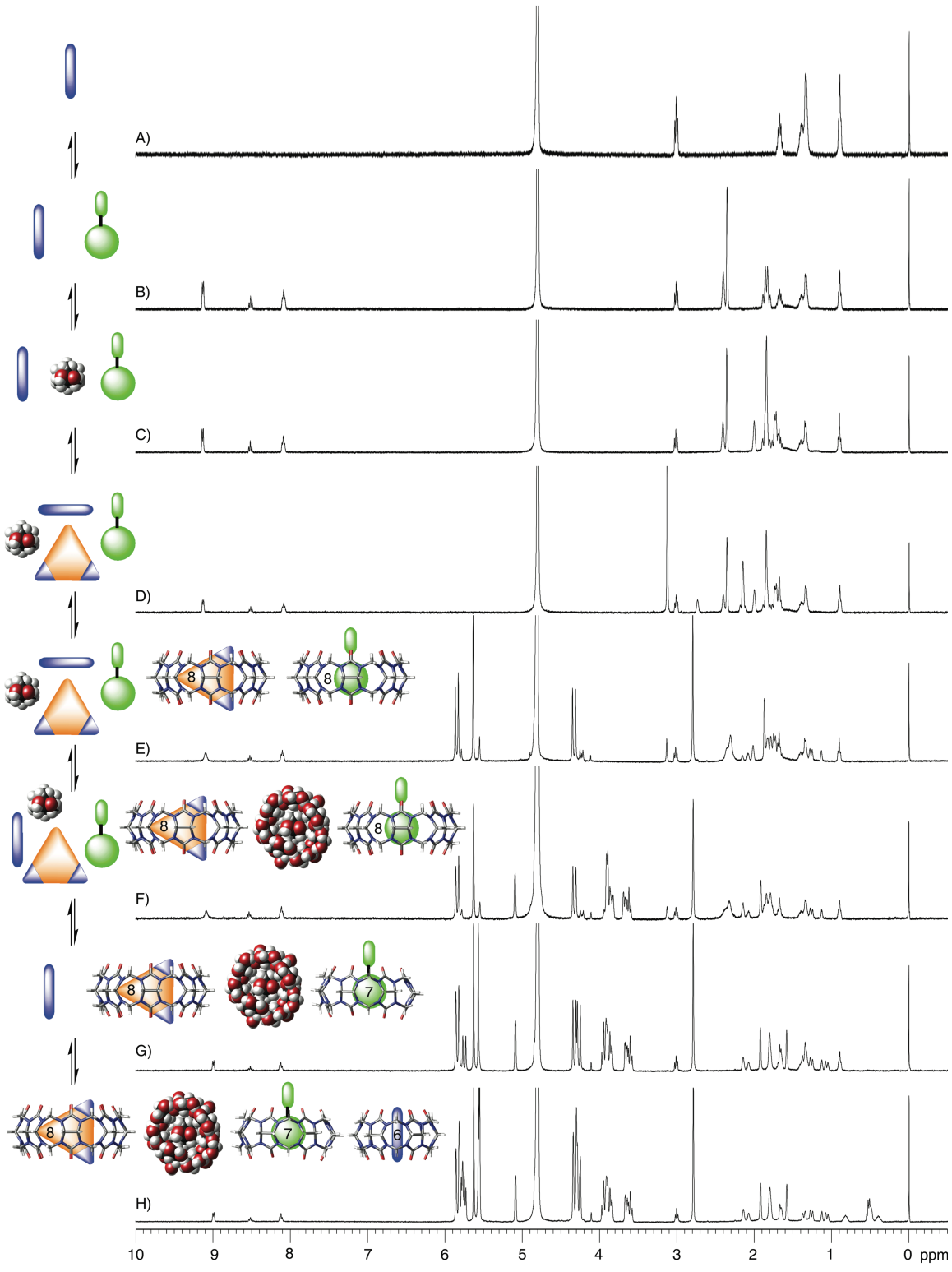



***Figure S9.*** $^{1}$H NMR spectra (400 MHz, $D_2O$, pD 7.4, 298 K, 1 mM) recorded for an equimolar mixture after addition of 1 eq. of: A) **1**, B) **3**, C) **2**, D) **4**, E) **8,** F) **5**, G) **7**, and H) **6**.

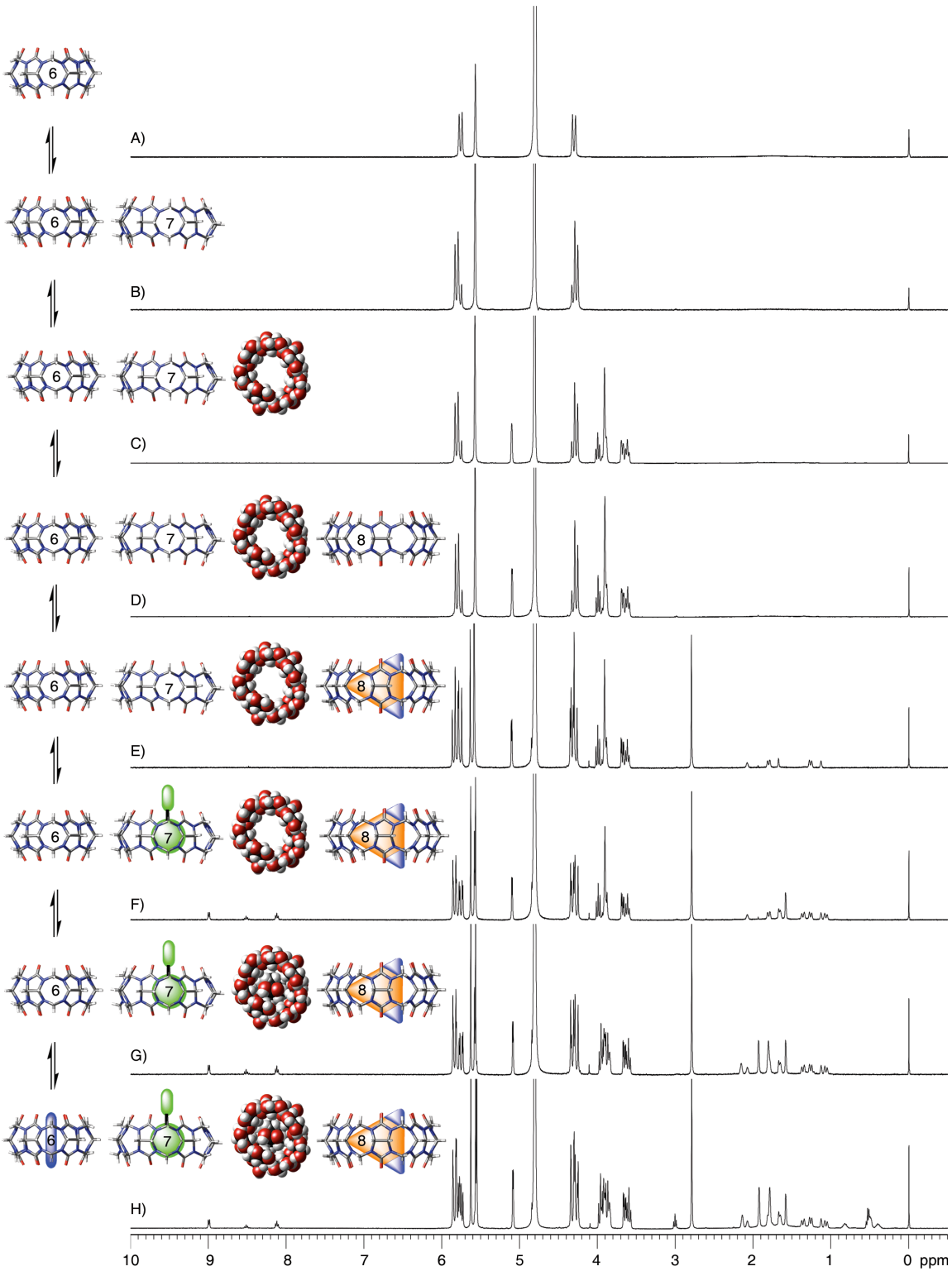



***Figure S10.*** $^{1}$H NMR spectra (400 MHz, $D_2O$, pD 7.4, 298 K, 1 mM) recorded for an equimolar mixture after addition of 1 eq. of: A) **6**, B) **7**, C) **5**, D) **8**, E) **4,** F) **3**, G) **2**, and H) **1**.

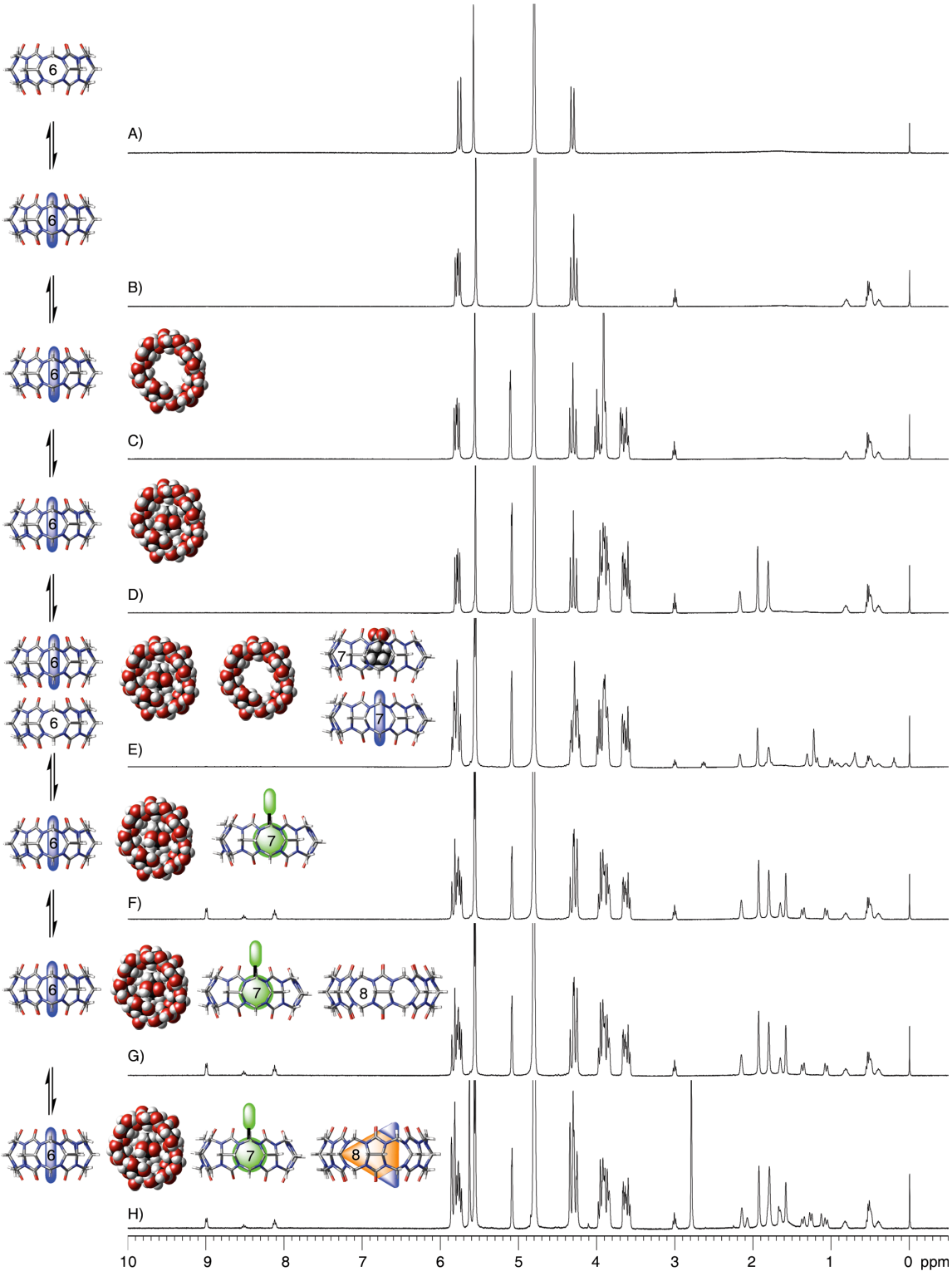



***Figure S11.*** $^{1}$H NMR spectra (400 MHz, $D_2O$, pD 7.4, 298 K, 1 mM) recorded for an equimolar mixture after addition of 1 eq. of: A) **6**, B) **1**, C) **5**, D) **2**, E) **7,** F) **3**, G) **8**, and H) **4**.

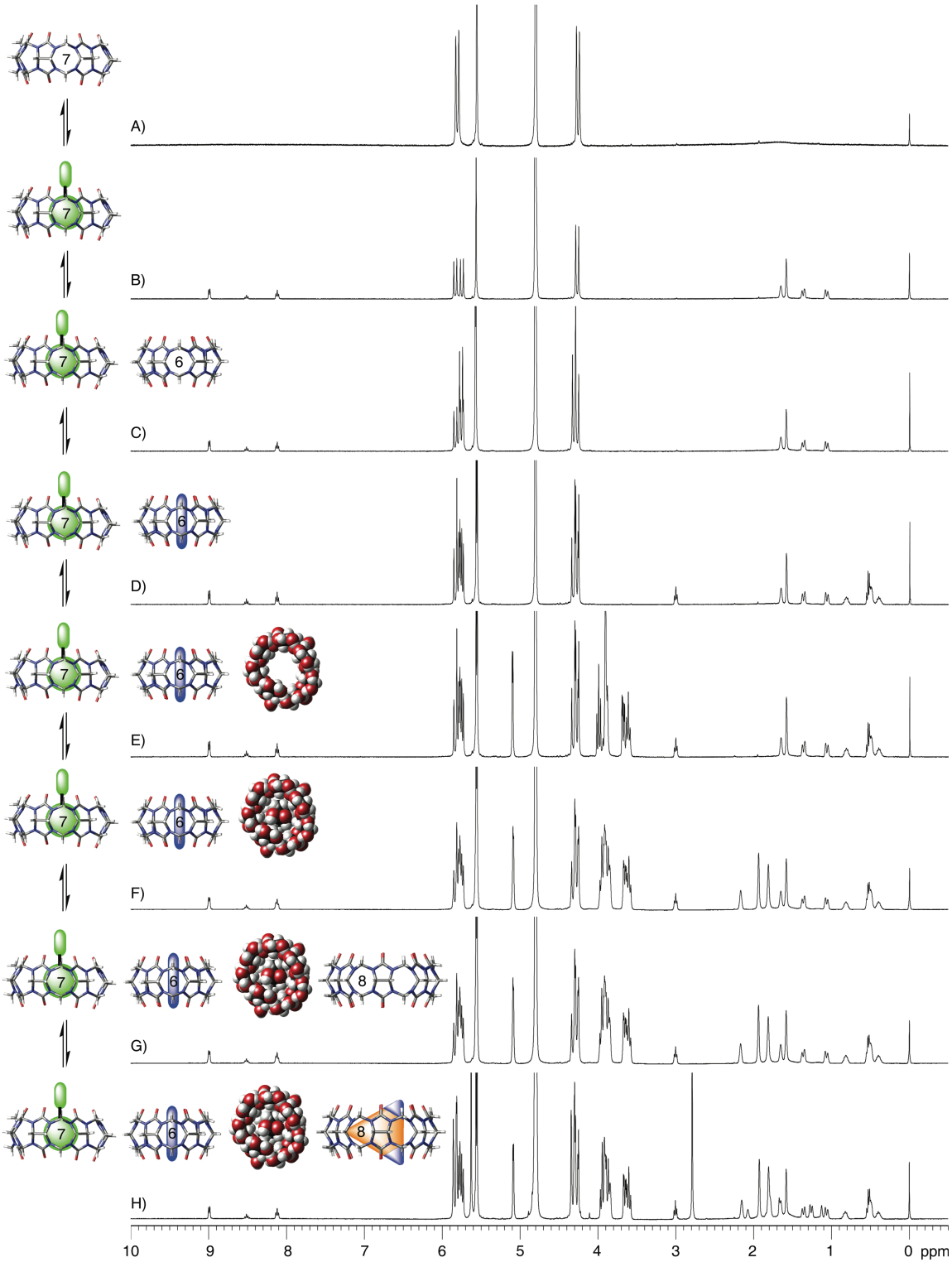



***Figure S12.*** $^{1}$H NMR spectra (400 MHz, $D_2O$, pD 7.4, 298 K, 1 mM) recorded for an equimolar mixture after addition of 1 eq. of: A) **7**, B) **3**, C) **6**, D) **1**, E) **5,** F) **2**, G) **8**, and H) **4**.

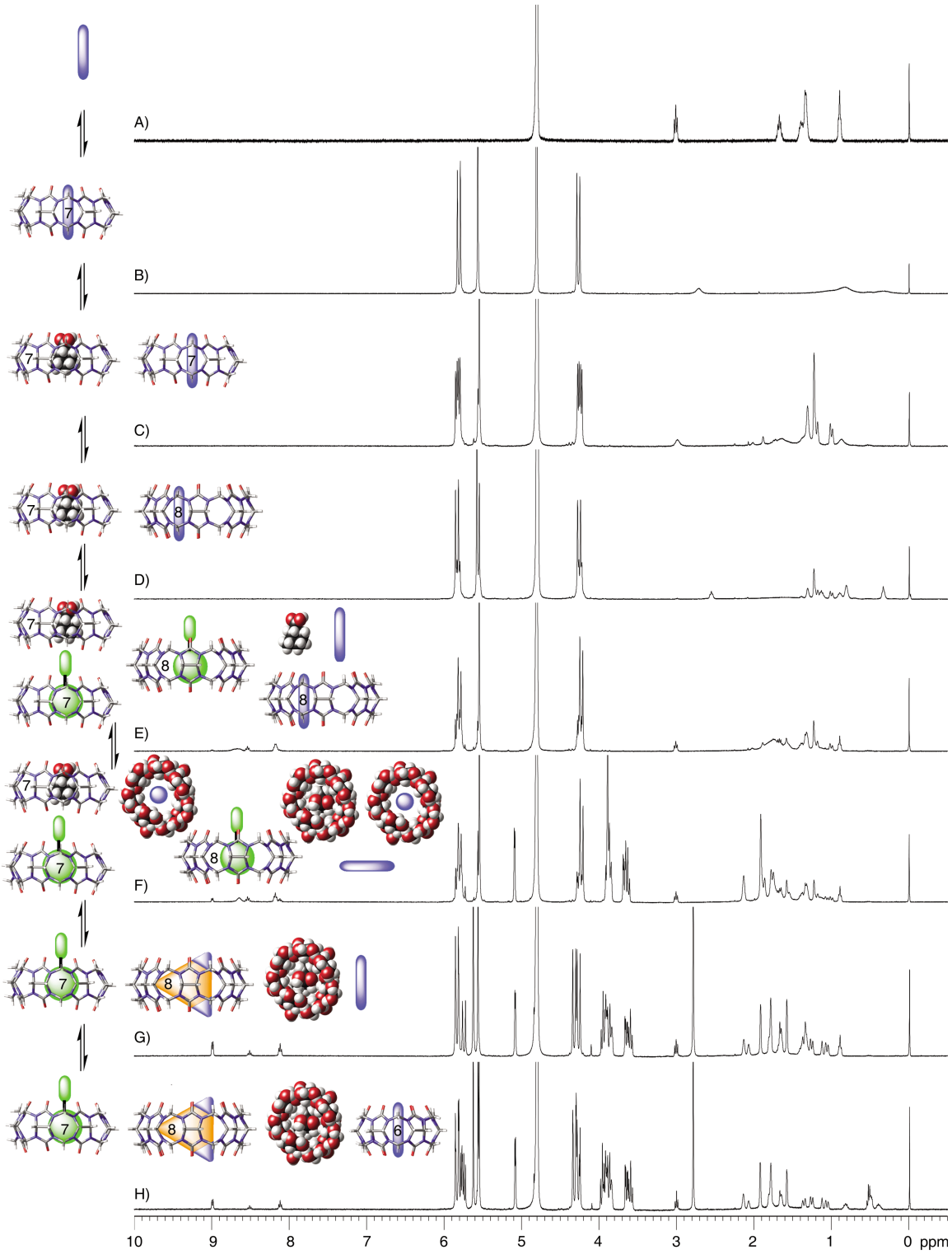



***Figure S13.*** $^{1}$H NMR spectra (400 MHz, $D_2O$, pD 7.4, 298 K, 1 mM) recorded for an equimolar mixture after addition of 1 eq. of: A) **1**, B) **7**, C) **2**, D) **8**, E) **3,** F) **5**, G) **4**, and H) **6**.

**A Sample Determination of Key Parameters In the Step-by-step Formation of a Four-Component System From Gepasi Output Files.**

The Gepasi Output files were imported into Microsoft Excel. Sample Microsoft Excel (.xls) files are deposited in the supporting information.

*Column A – M: Data obtained from Gepasi output file*.
Column A – D: Initial composition of the system in each state.
Column E – L: Steady state composition of the system in each state.

Formulas written below were applied to all cells under those column.

Column O: Total number of components present in each state.
Formula: O2 = SUM(A2:D2)

*Determination of whether a particular state is self-sorted or non self-sorted:*
This has been determined based on the assumption that all components in a self-sorted state have mole fractions either less than 0.1 or more than 0.9. Accordingly non-self-sorted states have one or more components in the range $0.1 \leq \chi \leq 0.9$.

Column Q – X: Formula: Q2 = IF(E2>0.1,1,0)
Column Z – AG: Formula Z2 = IF(E2<0.9,1,0)
Column AI: Formula: AI2 = SUM(Q2:X2:Z2:AG2)
Column AL: Formula: AL2 = IF(AK2=8,1,0)

Column AN – AQ: Steady state concentration of AM, AN, BM, and BN (same as column I2 – L2)

*Calculation of the free energy of each state:*
Column AS – AV: Formula: AS2 = PRODUCT(I2,LN(AN2))
Column AX: Formula: AX2 = SUM(AS2:AV2)
Column AZ: Formula: AZ2 =PRODUCT(AX2,298,0.00198)
Column BB: Formula: BB1 = =PRODUCT(AZ2,-1)

The Gepasi output file for the simulation of an eight component system was processed in a similar manner.

**Matlab Code Used for the Processing of the Gepasi Output Files for the Stepwise Formation of the Four Component Mixture.**

---

% Lines that start with >> are Matlab commands and % are just comments.
% All sixteen possible states ($2^4$) that occur in a stepwise formation of a four-component system can be concisely represented by a [4 × 16] matrix.

```
>> a = [0 0 0 0; 0 0 1 0; 0 0 0 1; 1 0 0 0; 0 1 0 0; 1 1 0 0; 1 0 1 0; 0 1 1 0; 1 0 0 1; 0 1 0 1; 0 0 1
    1; 1 1 1 0; 1 1 0 1; 1 0 1 1; 0 1 1 1; 1 1 1 1]
```

% For example:

| | | | | | |
|---|---|---|---|---|---|
| a = | 0 | 0 | 0 | 0 | no component |
| | 0 | 0 | 1 | 0 | M |
| | 0 | 0 | 0 | 1 | N |
| | 1 | 0 | 0 | 0 | A |
| | 0 | 1 | 0 | 0 | B |
| | 1 | 1 | 0 | 0 | A+B |
| | 1 | 0 | 1 | 0 | A+M |
| | 0 | 1 | 1 | 0 | B+M |
| | 1 | 0 | 0 | 1 | A+N |
| | 0 | 1 | 0 | 1 | B+N |
| | 0 | 0 | 1 | 1 | M+N |
| | 1 | 1 | 1 | 0 | A+B+M |
| | 1 | 1 | 0 | 1 | A+B+N |
| | 1 | 0 | 1 | 1 | A+M+N |
| | 0 | 1 | 1 | 1 | B+M+N |
| | 1 | 1 | 1 | 1 | A+B+M+N |

% where "one" represent the presence of a component and "zero" represent absence of a component in a particular state. Elements each row vector represents full configuration of one particular state and each column vector represent one particular component.

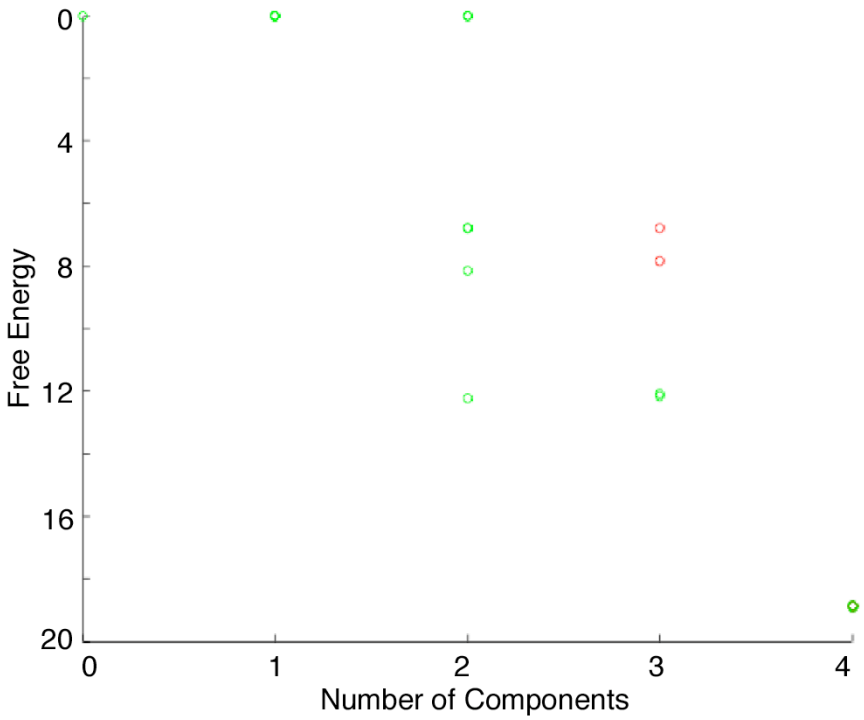
% Next, we define every possible connection between states uniquely. If addition of one or more elements changes the system from one particular state to another; there will be a total of 256 such possible transitions which can be represented by a [16 × 16] matrix.

```
>> z=zeros(16,16)
>> for J=1:16,
for I=J:16,
b=abs(a(I,:)-a(J,:));
z(I,J)=sum(b);
end
end
```

```
>> z
z =
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    1    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    1    2    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    1    2    2    0    0    0    0    0    0    0    0    0    0    0    0    0
    1    2    2    2    0    0    0    0    0    0    0    0    0    0    0    0
    2    3    3    1    1    0    0    0    0    0    0    0    0    0    0    0
    2    1    3    1    3    2    0    0    0    0    0    0    0    0    0    0
    2    1    3    3    1    2    2    0    0    0    0    0    0    0    0    0
    2    3    1    1    3    2    2    4    0    0    0    0    0    0    0    0
    2    3    1    3    1    2    4    2    2    0    0    0    0    0    0    0
    2    1    1    3    3    4    2    2    2    2    0    0    0    0    0    0
    3    2    4    2    2    1    1    1    3    3    3    0    0    0    0    0
    3    4    2    2    2    1    3    3    1    1    3    2    0    0    0    0
    3    2    2    2    4    3    1    3    1    3    1    2    2    0    0    0
    3    2    2    4    2    3    3    1    3    1    1    2    2    2    0    0
    4    3    3    3    3    2    2    2    2    2    2    1    1    1    1    0
```

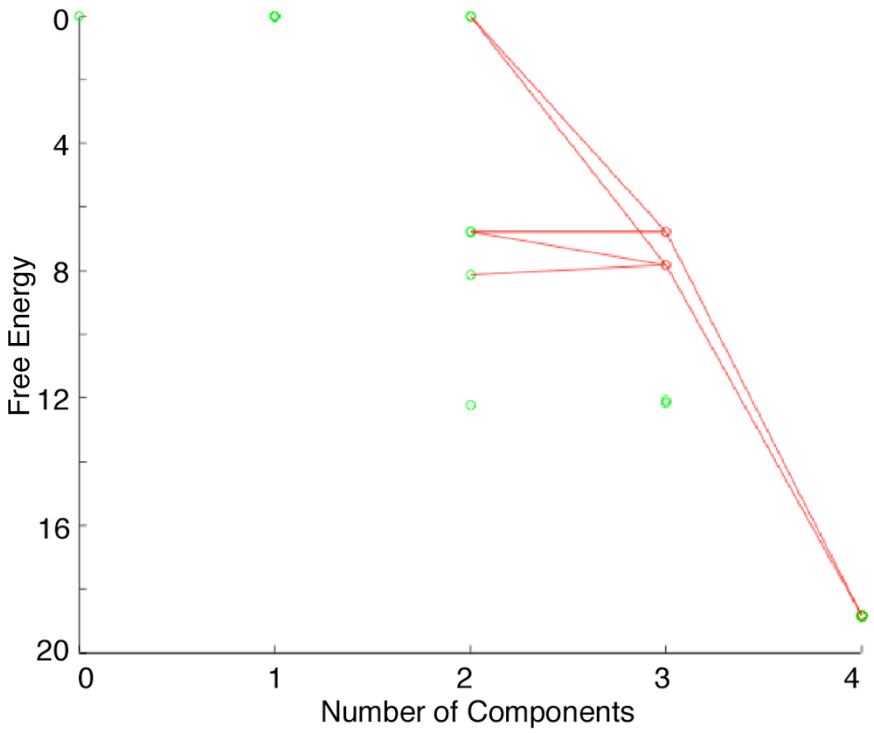
% $z_{i,j}$ (element in $i^{th}$ row and $j^{th}$ column) in the lower triangle represents the number of components need to be added in the transformation from one state to another. In the experiment we added one component at a time i.e. $z_{i,j} = 1$.

```
% We introduce three more column vectors m, n, and p:
% m represents total number of components in each state (corresponding to each row of a).
>> m = [0; 1; 1; 1; 1; 2; 2; 2; 2; 2; 2; 3; 3; 3; 3; 4]

% n represents free energy of each state (corresponding to each row of a).
>> n = [0; 0; 0; 0; 0; 0; -12.2272; -8.1436; -6.7716; -6.7716; 0; -12.1026; -6.7930; -12.1737;
     -7.8253; -18.8742]

% p represents whether a state is self-sorted or not (corresponding to each row of a).  A self-
sorted state is represented by “1” and non self-sorted state is represented by “0”.
>> p = [1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 0; 1; 0; 1]

% The following set of commands plots self-sorted and non self-sorted states separately.

>> hold on
>> for k=1:16,
X(1)=M(k);
Y(1)=N(k);
if p(k)==1;
plot(X,Y,'go');
else plot(X,Y,'ro');
end
end
```




% The following set of commands connects two adjacent states if at least one of the is non self-sorted (two red circles or one red and one green circle)

```
>> hold on
>> for u=1:16,
for v=1:16,
X(1)=m(v);
Y(1)=n(v);
X(2)=m(u);
Y(2)=n(u);
if(z(v,u)==1)&&((p(v)==0)||(p(u)==0))
plot(X,Y,'r')
end
end
end
```




% The following set of commands connects two adjacent self-sorted states (green circles) by a green line.

```
>> hold on
>> for u=1:16,
for v=1:16,
X(1)=M(v);
Y(1)=N(v);
X(2)=M(u);
Y(2)=N(u);
if((z(v,u)==1)&&(P(v)==1)&&(P(u)==1))
plot(X,Y,'g')
end
end
end
```

% Final Plot:

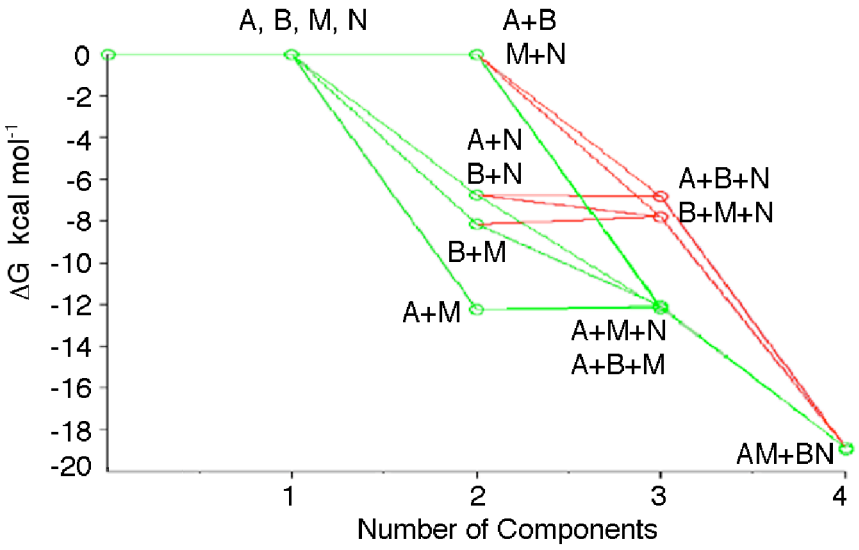



***Figure S14:*** A plot showing free energies of all the states in the stepwise formation of four-component system. The overall free energy of the system decreases as more components are added.

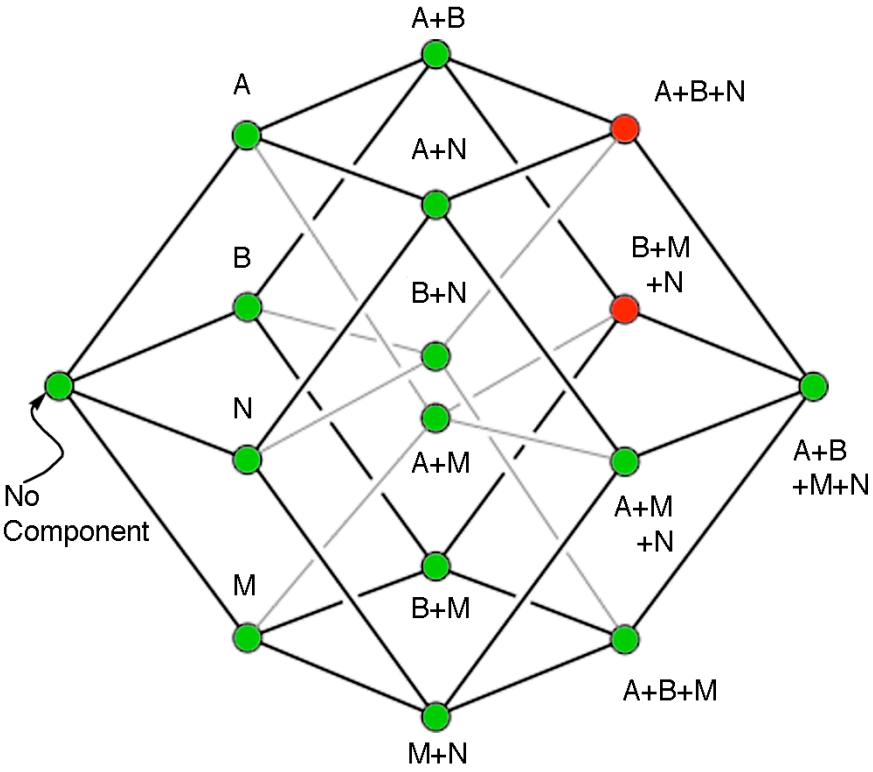



***Figure S15:*** A four dimensional hypercube representing all the states in the stepwise formation of four component system.

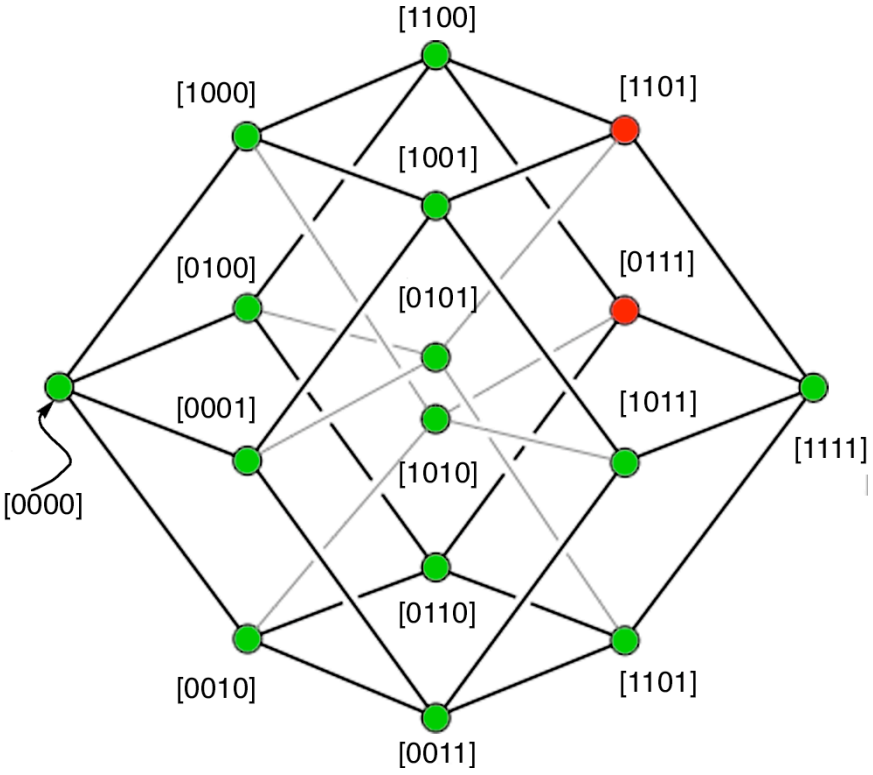



***Figure S16:*** A four dimensional hypercube representing all the states in a binary form in the stepwise formation of four component system.

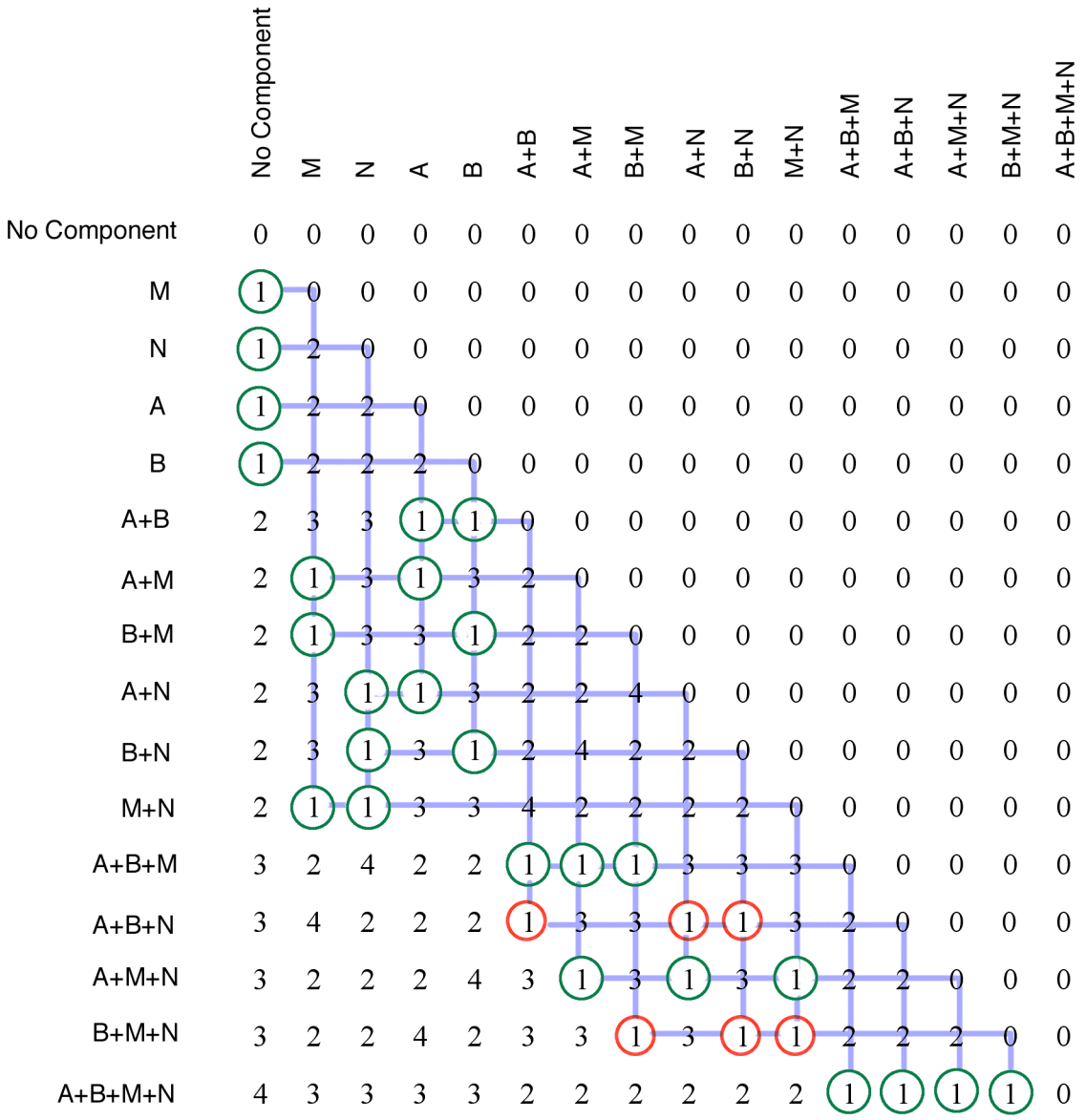



***Figure S17:*** Depiction of the connectivity among states in the stepwise formation of a four component system. A element $z_{i,j}$ in matrix z indicates the number of components need to be added in the transformation from one state to another. In the experiment we added one component at a time i.e. $z_{i,j} = 1$. Green and red circles represent self-sorted and non self-sorted states respectively. Connectivities are represented by blue lines.

Matlab codes for Figure 8a, and 8b.

---

```
clear all

% Lines starting with % are just comments.

% These are the known values obtained from the simulation of the stepwise formation of a eight
component system.
>> a = composition_8comp;
>> m = components_8comp;
>> n = delta_G_8comp;
>> p = self_sorting;

% This plots self-sorted and non self-sorted states separately.
>> for j= 1:256,
for i= j:256,
b= abs(A(i,:)-A(j,:));
z(i,j)= sum(b);
end
end
>> hold on
>> for k= 1:256,
X(1)= m(k);
Y(1)= n(k);
if p(k)==1;
plot(X,Y,'go');
else plot(X,Y,'ro');
end
end

% This connects between two adjacent self-sorted states (green circles) by a green line.
>> hold on
>> for u = 1:256,
for v = 1:256,
X(1) = M(v);
Y(1) = N(v);
X(2) = M(u);
Y(2) = N(u);
if ((Z(v,u) == 1) && (P(v) == 1) && (P(u) == 1))                                    (1)
plot(X,Y,'g')
end
end
end

% This connects two adjacent states if at least one of the is non self-sorted (two red circles or one
red and one green circle).
>> hold on
```

```
>> for u= 1:256,
for v= 1:256,
X(1)= M(v);
Y(1)= N(v);
X(2)= M(U);
Y(2)= N(U);
if (Z(V,U) == 1) && ((P(V) == 0) || (P(U) == 0))
Z=floor(Z)
plot(X,Y,'r')
end
end
end
```

---

Model for Figure 8c – 8f.

---

Figure 8c
Command in (1) is replaced by:

```
 if((z(v,u)==1)&&(p(v)==1)&&(p(u)==1))
plot(X,Y,'g')
```

Figure 8d
Command in (1) is replaced by:

```
if((z(v,u)==1)&&(p(v)==0)&&(p(u)==0))
plot(X,Y,'r')
```

Figure 8e
Command in (1) is replaced by:

```
if((z(v,u)==1)&&(p(v)==1)&&(p(u)==0))
plot(X,Y,'b')
```

Figure 8f
Command in (1) is replaced by:

```
if((z(v,u)==1)&&(p(v)==0)&&(p(u)==1))
plot(X,Y,'k')
```

---

Model for Figure 10a – 10d.

---

**Addition of all four guests followed by all four hosts:**

```
>> a = composition_8comp;
>> m = components_8comp;
>> n = delta_G_8comp;
>> p = self_sorting;
```

```
>> whos
  Name      Size                Bytes  Class

  a        31x8                1984  double array
  m         31x1                248  double array
  n        31x1                248  double array
  p        31x1                248  double array

Grand total is 341 elements using 2728 bytes

>> for j= 1:31,
for i= j:31,
b= abs(a(i,:)-a(j,:));
z(i,j)= sum(b);
end
end
>> hold on
>> for k= 1:31,
X(1)= m(k);
Y(1)= n(k);
if p(k)==1;
plot(X,Y,'go');
else plot(X,Y,'ro');
end
end
>> for u = 1:31,
for v = 1:31,
X(1) = m(v);
Y(1) = n(v);
X(2) = m(u);
Y(2) = n(u);
if ((z(v,u) == 1) && (p(v) == 1) && (p(u) == 1))
plot(X,Y,'g')
end
end
end
>> for u= 1:31,
for v= 1:31,
X(1)= m(v);
Y(1)= n(v);
X(2)= m(u);
Y(2)= n(u);
if (z(v,u) == 1) && ((p(v) == 0) || (p(u) == 0))
plot(X,Y,'r')
end
end
end
```

**Alternate Addition Sequences:**

```
>> a = composition_8comp;
>> m = components_8comp;
>> n = delta_G_8comp;
>> p = self_sorting;

>> whos
  Name      Size                Bytes  Class

  a       126x8                8064  double array
  m        126x1                1008  double array
  n       126x1                1008  double array
  p       126x1                1008  double array

Grand total is 1386 elements using 11088 bytes

>> for j= 1:126,
for i= j:126,
b= abs(a(i,:)-a(j,:));
z(i,j)= sum(b);
end
end
>> hold on
>> for k= 1:126,
X(1)= m(k);
Y(1)= n(k);
if p(k)==1;
plot(X,Y,'go');
else plot(X,Y,'ro');
end
end
>> for u = 1:126
for v = 1:126
X(1) = m(v);
Y(1) = n(v);
X(2) = m(u);
Y(2) = n(u);
if ((z(v,u) == 1) && (p(v) == 1) && (p(u) == 1))
plot(X,Y,'g')
end
end
end
>> for u= 1:126
for v= 1:126
```

```
X(1)= m(v);
Y(1)= n(v);
X(2)= m(u);
Y(2)= n(u);
if (z(v,u) == 1) && ((p(v) == 0) || (p(u) == 0))
plot(X,Y,'r')
end
end
end
```

---

Model for Figure 11.

---

```
clear all

W1= deltaG(1,:);
W2= deltaG(2:9,:);
W3= deltaG(10:37,:);
W4= deltaG(38:93,:);
W5= deltaG(94:163,:);
W6= deltaG(164:219,:);
W7= deltaG(220:247,:);
W8= deltaG(248:255,:);
W9= deltaG(256,:);

g1=reshape(W1,6*1,1);
>> g2=reshape(W2,6*8,1);
>> g3=reshape(W3,6*28,1);
>> g4=reshape(W4,6*56,1);
>> g5=reshape(W5,6*70,1);
>> g6=reshape(W6,6*56,1);
>> g7=reshape(W7,6*28,1);
>> g8=reshape(W8,6*8,1);
>> g9=reshape(W9,6*1,1);

>> hist(g5,20)
>> N=  hist(g5,20);

>> [N,X]=hist(g5);
>> plot(N,X)
>> plot(X,N)
>>

>> N1= hist(g1);
>> N2= hist(g2);
>> N3= hist(g3);
```

```
>> N4= hist(g4);
>> N5= hist(g5);
>> N6= hist(g6);
>> N7= hist(g7);
>> N8= hist(g8);
>> N9= hist(g9);
>> [N1,X]= hist(g1);

>> plot(X,N1)
>> [N2,X]= hist(g2);
>> [N3,X]= hist(g3);
>> [N4,X]= hist(g4);
>> [N5,X]= hist(g5);
>> [N6,X]= hist(g6);
>> [N7,X]= hist(g7);
>> [N8,X]= hist(g8);
>> [N9,X]= hist(g9);

>> N= [N1;N2;N3;N4;N5;N6;N7;N8;N9]
>> X=linspace(-50,5,10);
X =

  -50.0000  -43.8889  -37.7778  -31.6667  -25.5556  -19.4444  -13.3333   -7.2222   -1.1111
5.0000

>> [N9,X]= hist(g9,X);
>> [N9,X]= hist(g9,X);
>> hist(g9,X)

>> subplot(9,1,1)
hist(g1,X)
subplot(9,1,2)
hist(g2,X)
subplot(9,1,3)
hist(g3,X)
subplot(9,1,4)
hist(g4,X)
subplot(9,1,5)
hist(g5,X)
subplot(9,1,6)
hist(g6,X)
subplot(9,1,7)
hist(g7,X)
subplot(9,1,8)
hist(g8,X)
subplot(9,1,9)
hist(g9,X)
```