Supporting information for

Coassembly of Peptides Derived from β-Sheet Regions of β-Amyloid

Nicholas L. Truex and James S. Nowick*

Department of Chemistry, University of California, Irvine, Irvine, CA 92697-2025

*To whom correspondence should be addressed: jsnowick@uci.edu

I. Supplemental Figures

II. Materials and Methods

I. Supplemental Figures

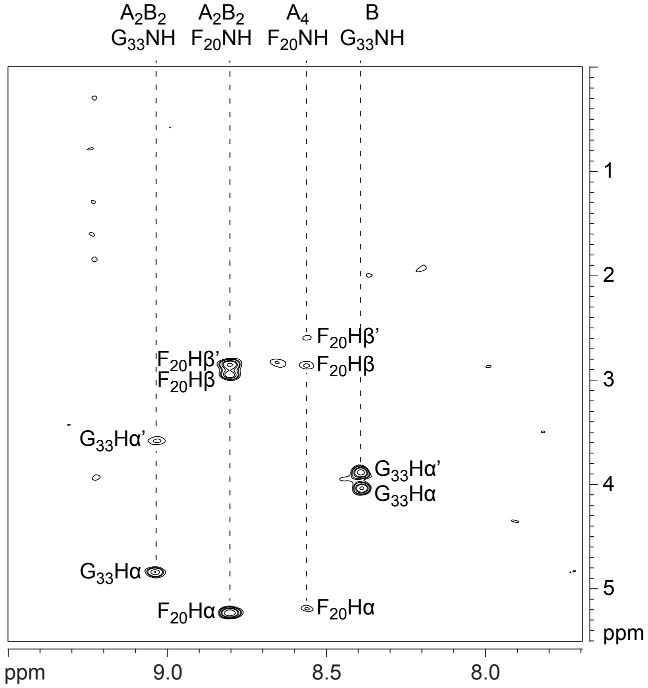


Figure S1. ^{15}N-Edited TOCSY spectrum of the 1:1 mixture of peptides [^{15}N]**1a** and [^{15}N]**1b** at 8.0 mM total concentration in 9:1 H_2O/D_2O at 600 MHz and 293 K.

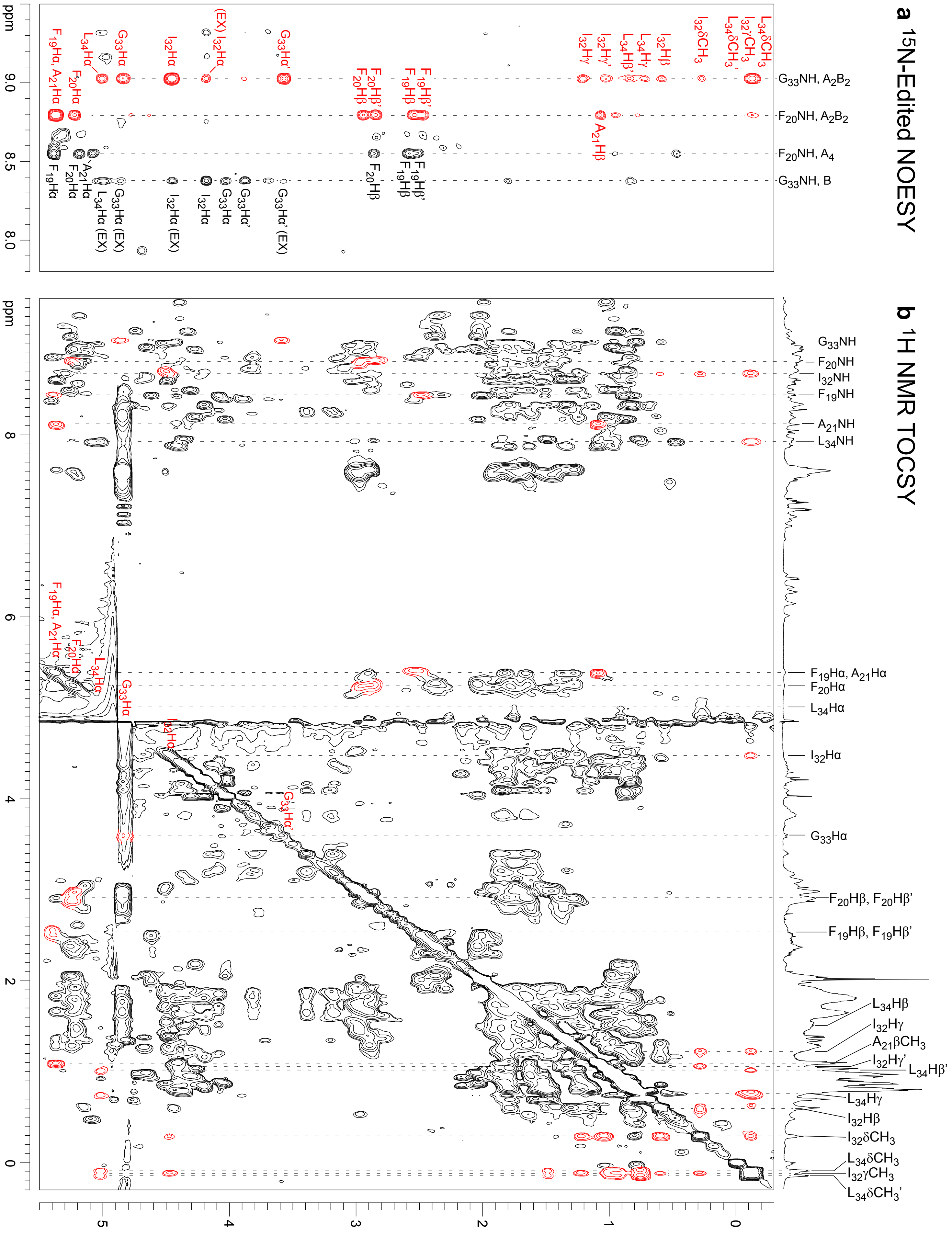


Figure S2. (Continued on next page.)

Figure S2 (preceding page). Comparison of the (a) ^{15}N-Edited NOESY spectrum of the 1:1 mixture of peptides [^{15}N]**1a** and [^{15}N]**1b** with the (b) ^{1}H NMR TOCSY spectrum of the 1:1 mixture of peptides **1a** and **1b** in 9:1 H_2O/D_2O at 600 MHz and 293 K. Key crosspeaks associated with the A_2B_2 heterotetramer are highlighted in red from F_{19}, F_{20}, and A_{21} of peptides [^{15}N]**1a** and **1a**, and also from I_{32}, G_{33}, L_{34} of peptides [^{15}N]**1b** and **1b**.

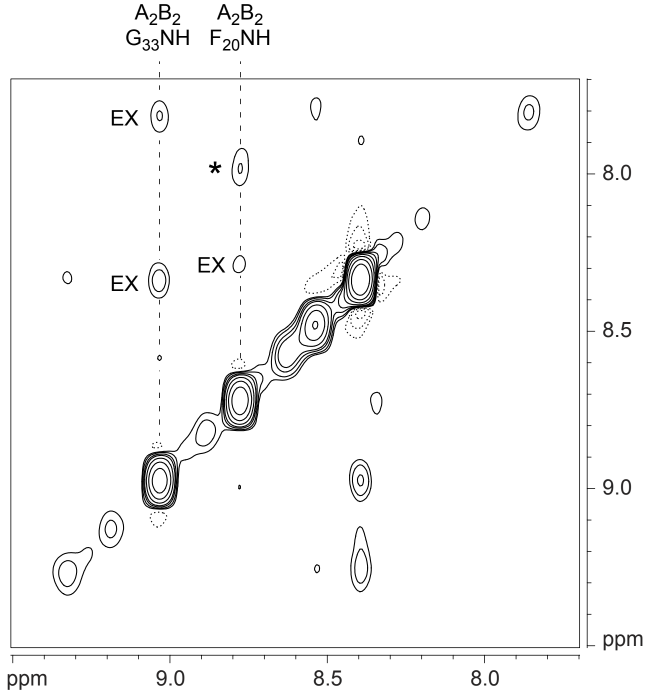


Figure S3. ^{15}N-edited NOESY spectrum of the 1:1 mixture of peptides [^{15}N]**1a** and [^{15}N]**1b** at 8.0 mM total concentration in 9:1 H_2O/D_2O at 600 MHz and 293 K. Crosspeaks associated with chemical exchange between the monomers and tetramers are labeled EX. The asterisk (*) indicates a crosspeak from a minor unidentified species associated with the A_2B_2 heterotetramer.

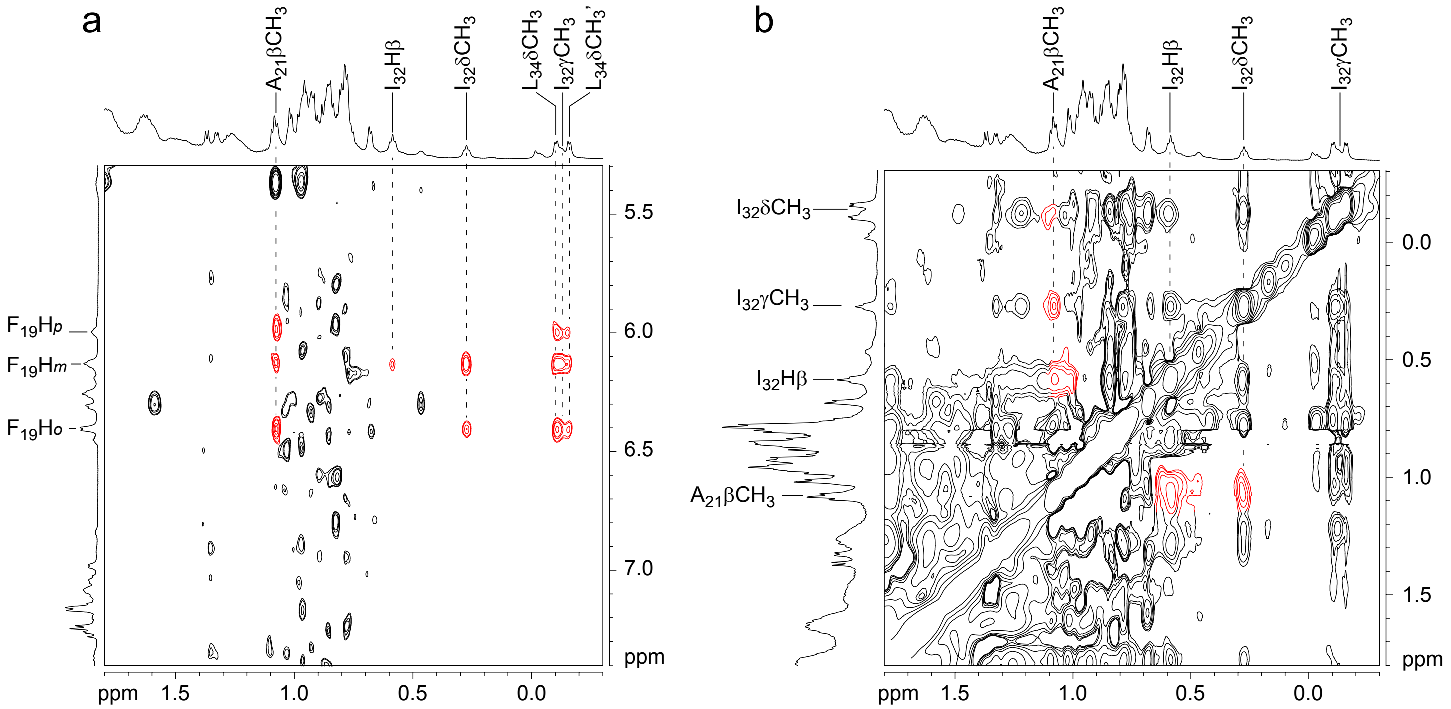
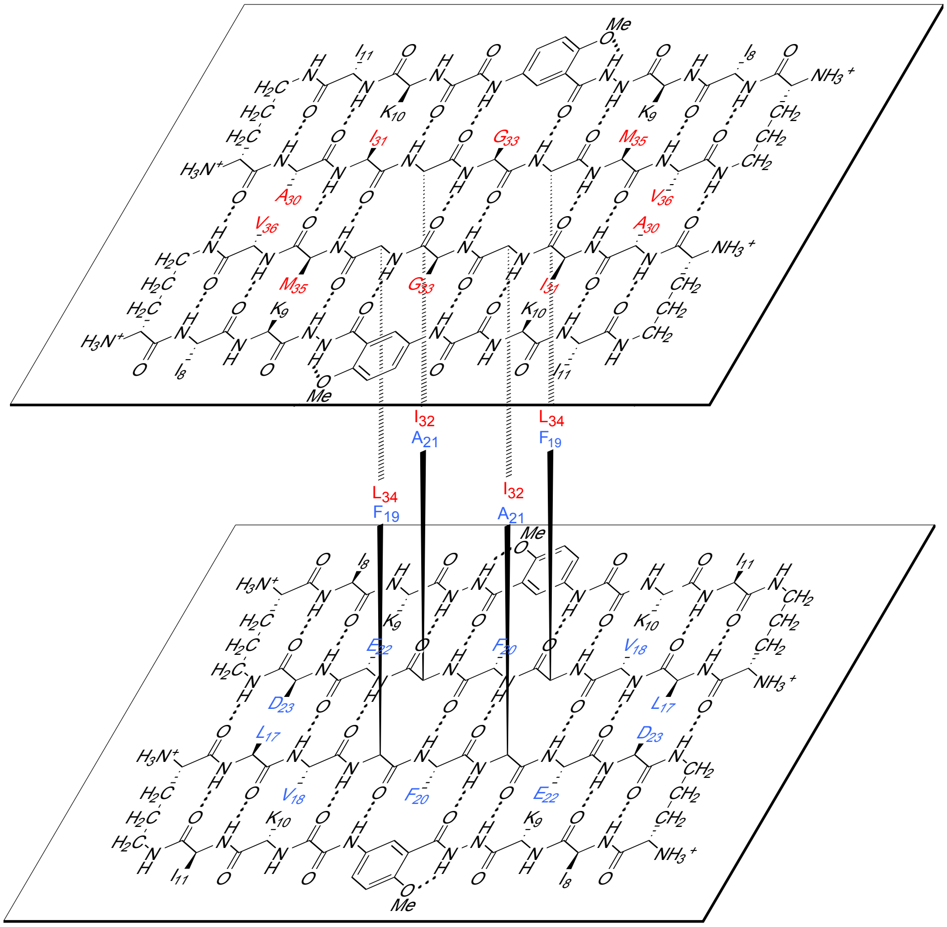


Figure S4. Expansions of the ^{1}H NMR NOESY spectrum of the 1:1 mixture of peptides **1a** and **1b** at 8.0 mM total concentration in 9:1 H_2O/D_2O at 600 MHz and 293 K. Key interlayer NOEs between A·A and B·B homodimers from the A_2B_2 heterotetramer are highlighted in red. F_{19}H*o*, F_{19}H*m*, and F_{19}H*p*, correspond to the ortho, meta, and para protons of F_{19}.



A_2B_2 heterotetramer
A•A/B•B topological isomer

Figure S5. A_2B_2 heterotetramer consisting of two hydrogen-bonded homodimers of peptides **1a** and **1b**. Contacts between the side chains of F_{19} and L_{34} and between the side chains of A_{21} and I_{32} are shown, reflecting observed NOEs.

II. Materials and Methods

Synthesis of Peptides 1

Synthesis and purification of peptides **1a** and **1b**, and [^{15}N]**1a** and [^{15}N]**1b** were performed as described in the preceding paper.[1]

Fmoc-Protection of ^{15}N-Labeled Amino Acids

Fmoc-protection of ^{15}N-labeled glycine and phenylalanine was performed as described in the preceding paper.[1,2]

NMR Spectroscopy of Peptides 1

Sample Preparation. NMR spectroscopy of peptides **1a** and **1b** was performed in D_2O (D, 99.96%; Cambridge Isotope Laboratories, Inc.). The solutions were prepared by dissolving a weighed portion of the peptide in the appropriate volume of solvent. The molecular weights of the peptides were calculated as the TFA salts with all amino groups assumed to be protonated (**1a**, M.W. 2223.85 and **1b**, M.W. 2099.91). The solutions were allowed to stand for 24 h to allow complete hydrogen to deuterium exchange of the amide NH protons.

^{1}H NMR, TOCSY, and NOESY Data Collection. NMR spectra were recorded on a Bruker 600 MHz spectrometer with a TBI probe. Presaturation water suppression was applied as needed. TOCSY spectra were recorded with 2048 points in the f_2 dimension and 512 increments in the f_1 dimension with a 150-ms spin-lock mixing time. NOESY spectra were recorded with 2048 points in the f_2 dimension and 512 increments in the f_1 dimension with a 150-ms mixing time.

^{1}H NMR, TOCSY, and NOESY Data Processing. NMR spectra were processed with Bruker XwinNMR software. Automatic baseline correction was applied in both dimensions after phasing the spectra. TOCSY spectra were Fourier transformed to a final matrix size of 2048 x 2048 real points using a Qsinc weighting function (GB = 0.05) and forward linear prediction. NOESY spectra were Fourier transformed to a final matrix size of 2048 x 2048 real points using a Qsinc weighting function (GB = 0.05) and forward linear prediction.

Diffusion-Ordered Spectroscopy (DOSY) Experiments. DOSY experiments were performed on a Bruker 500 MHz spectrometer equipped with a TCI cryoprobe, with a diffusion delay (Δ) of 75-ms and a diffusion gradient length (δ) of 2.5-ms. Sixteen sets of FIDs were recorded with the gradient strength incremented from 5%–95% using a linear ramp. The combined FIDs were Fourier transformed in Bruker's TopSpin^{TM} software to give a pseudo-2D spectrum. After phasing and performing baseline correction, each pseudo-2D spectrum was processed with logarithmic scaling on the Y-axis. The Y-axis was calibrated to the diffusion coefficient of the residual HOD peak in D_2O (1.9×10^{-9} m^2/s at 298 K).[3] The diffusion coefficients of the peptides were read and converted from logarithmic values to linear values.

NMR Spectroscopy of Peptides [^{15}N]1

Sample Preparation. NMR spectroscopy of peptides [^{15}N]**1a** and [^{15}N]**1b** was performed in 9:1 H_2O/D_2O. The solutions were prepared by dissolving a weighed portion of the peptide in the appropriate volume of solvent. The molecular weights of the peptides were calculated as the TFA salts with all amino groups assumed to be protonated ([^{15}N]**1a**, M.W. 2224.85 and [^{15}N]**1b**, M.W. 2100.91). 4,4-Dimethyl-4-silapentane-1-ammonium trifluoroacetate (DSA) was added as an internal standard for referencing chemical shifts.[4]

^{1}H NMR, ^{1}H,^{15}N HSQC, ^{1}H,^{15}N TOCSY-HSQC (^{15}N-edited TOCSY), and ^{1}H,^{15}N NOESY-HSQC (^{15}N-edited NOESY) Data Collection. NMR spectra were recorded on a Bruker 600 MHz spectrometer with either a TBI probe or a BBFO cryoprobe. Gradient water suppression was applied as needed. ^{1}H,^{15}N HSQC spectra were recorded with 1024 points in the f_2 dimension and 512 increments in the f_1 dimension. ^{1}H,^{15}N TOCSY-HSQC spectra were recorded with a 150-ms spin-lock mixing time, and with 2048 points in the f_3 dimension (^{1}H), one increment in the f_2 dimension (^{15}N), and 512 increments in the f_1 dimension (^{1}H). ^{1}H,^{15}N NOESY-HSQC spectra were recorded with a 150-ms mixing time, and with 2048 points in the f_3 dimension (^{1}H), 1 increment in the f_2 dimension (^{15}N), and 1024 increments in the f_1 dimension (^{1}H).

^{1}H NMR, ^{1}H,^{15}N HSQC, ^{1}H,^{15}N TOCSY-HSQC (^{15}N-edited TOCSY), and ^{1}H,^{15}N NOESY-HSQC (^{15}N-edited NOESY) Data Processing. NMR spectra were Fourier transformed in Bruker XwinNMR software with forward linear prediction and a Qsinc weighting function. Automatic baseline correction was applied in both dimensions after phasing the spectra. The ^{1}H,^{15}N HSQC spectra were processed to a final matrix size of 2048 x 1024 real points and with GB = 0.1 in the f_2 dimension. The ^{1}H,^{15}N TOCSY-HSQC spectra were processed to a final 2D matrix size of 2048 x 1024 real points (f_3, f_1) and with GB = 0.05 in both dimensions. The ^{1}H,^{15}N NOESY-HSQC spectra were processed to a final 2D matrix size of 4096 x 2048 real points (f_3, f_1) and with GB = 0.05 in both dimensions.

Molecular Modeling of Peptides 1a and 1b.

Molecular models of the A_2B_2 heterotetramers were generated using the models and methods from the preceding paper.[1] The A_4 and B_4 homotetramers of peptides **1a** and **1b** were imported into PyMOL: Peptide monomers were selected to construct the A·A and B·B homodimer subunits within the A·A/B·B topological isomer. Peptide monomers were selected to construct the two A·B heterodimer subunits within the A·B/A·B topological isomer. The dimer subunits were oriented so that the side chains of L_{17}, F_{19}, A_{21}, and D_{23} and the side chains of A_{30}, I_{32}, L_{34}, and V_{36} formed the hydrophobic core of the A_2B_2 heterotetramers.

The coordinates were exported from PyMOL. [Note that .pdb was used, but .mol2 file format is actually preferable and is recommended instead of .pdb.] The file was imported into MacroModel with the Maestro user interface. Atom types and bond orders were edited as needed to correct errors in bond type and charge. Distance constraints were applied to reflect the folding and dimerization of the macrocycles. Four interlayer distance constraints between the δ-methyl group of Ile_{11} and the methoxy group of Hao were applied to reflect the observed interlayer contacts. Minimization was performed with the MMFFs force field and GB/SA water solvation. All constraints were removed and minimization was repeated to generate a minimum-energy conformation (local minimum). The coordinates were exported in .pdb file format and imported into PyMOL.

Job's Method of Continuous Variation

Nine samples of peptides [^{15}N]**1a** and [^{15}N]**1b** were prepared at 8.0 mM total concentration with mole fractions of peptide [^{15}N]**1b** = 0.00, 0.125, 0.25, 0.375, 0.50, 0.625, 0.75, 0.875, and 1.00. An ^{1}H,^{15}N HSQC spectrum at 600 MHz and 293 K was recorded for each mixture using the data collection and data processing parameters described above. These spectra are shown on pages S40-48.

The spectra were reprocessed in Bruker's TopSpin^{TM} software using a Qsine weighting function to sharpen the crosspeaks for measuring the intensities. One-dimensional ^{15}N spectra from the two-dimensional ^{1}H,^{15}N HSQC spectra were generated by typing "f1sum" in the command line. A stack plot of the ^{15}N spectra is shown on page S49.

The volume integrals of the crosspeaks in the ^{1}H,^{15}N HSQC spectra were measured and normalized to 1.0. Table S1 summarizes the volume integrals versus the mole fraction of peptide [^{15}N]**1b**, χ_B.

Table S1. Relative integrals of the crosspeaks *1–14* from the ^{1}H,^{15}N HSQC spectra

	A	A_4	B	B_4	A_2B_2		A_3B_1				A_1B_3			
χ_B	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*
0.000	0.0664	0.9336	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.125	0.0449	0.6883	0.0608	0.0000	0.0438	0.0435	0.0363	0.0268	0.0427	0.0125	0.0000	0.0000	0.0000	0.0003
0.250	0.0332	0.4359	0.1089	0.0005	0.1309	0.1345	0.0255	0.0421	0.0318	0.0466	0.0031	0.0030	0.0028	0.0012
0.375	0.0281	0.2986	0.1445	0.0042	0.1854	0.1785	0.0424	0.0393	0.0335	0.0239	0.0039	0.0076	0.0064	0.0037
0.500	0.0177	0.1350	0.1987	0.0209	0.2330	0.2310	0.0205	0.0308	0.0262	0.0281	0.0195	0.0158	0.0087	0.0140
0.625	0.0115	0.0608	0.2699	0.0384	0.2423	0.2325	0.0207	0.0206	0.0148	0.0134	0.0278	0.0224	0.0216	0.0033
0.750	0.0082	0.0096	0.3095	0.1466	0.1782	0.1734	0.0037	0.0097	0.0071	0.0055	0.0435	0.0390	0.0293	0.0368
0.875	0.0044	0.0009	0.3820	0.2741	0.0812	0.0783	0.0035	0.0020	0.0000	0.0006	0.0466	0.0351	0.0475	0.0440
1.000	0.0000	0.0000	0.4796	0.5204	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

To generate the Job plot, the relative integrations of the monomers, homotetramers, and heterotetramers were plotted versus the mole fraction χ_B. The normalized integrals of crosspeaks *1* and *2* were used for the relative integrations of the A monomer and A_4 homotetramer, respectively; the normalized integrals of crosspeaks *3* and *4* were used for the relative integrations of the B monomer and B_4 homotetramer, respectively. The sum of the normalized integrals of crosspeaks *5* and *6* was used for the relative integration of the A_2B_2 heterotetramer; the sum of the normalized integrals of crosspeaks *7–10* was used for the relative integration of the A_3B_1 heterotetramer; and the sum of the normalized integrals of crosspeaks *11–14* was used for the relative integration of the A_1B_3 heterotetramer. Table S2 summarizes the relative integrations. Figure 8 illustrates the resulting Job plot.

Table S2. Relative integrations for the monomers, homotetramers, and heterotetramers

χ_B	A	B	A_4	A_3B_1	A_2B_2	A_1B_3	B_4
0.000	0.0664	0.0000	0.9336	0.0000	0.0000	0.0000	0.0000
0.125	0.0449	0.0608	0.6883	0.1183	0.0873	0.0003	0.0000
0.250	0.0332	0.1089	0.4359	0.1459	0.2654	0.0102	0.0005
0.375	0.0281	0.1445	0.2986	0.1391	0.3639	0.0216	0.0042
0.500	0.0177	0.1987	0.1350	0.1057	0.4640	0.0581	0.0209
0.625	0.0115	0.2699	0.0608	0.0695	0.4748	0.0751	0.0384
0.750	0.0082	0.3095	0.0096	0.0260	0.3516	0.1486	0.1466
0.875	0.0044	0.3820	0.0009	0.0060	0.1595	0.1731	0.2741
1.000	0.0000	0.4796	0.0000	0.0000	0.0000	0.0000	0.5204

III. Mathematical Derivations for the Monomer–Homotetramer–Heterotetramer Equilibrium Model

This section describes the mathematical derivations for the monomer–homotetramer–heterotetramer equilibrium model. This model was used for nonlinear least-squares fitting of the Job plot and for generating simulated Job plots. The mathematical derivations are based on those developed by Collum and co-workers in the supporting information of Liou, L. R; McNeil, A. J.; Ramirez, A.; Toombes, G. E. S.; Gruver, J. M.; Collum, D. B. *J. Am. Chem. Soc.* **2008**, *130*, 4859–4868. The following subsections describe (A) the mathematical derivations with general equations; (B) the implementation for homotetramers and heterotetramers; (C) and our implementation for monomers, homotetramers, and heterotetramers.

A. General Equations

In this subsection, we describe the mathematical derivations with general equations. The general equations calculate the concentrations of homooligomers and heterooligomers (that are the same size) as a function of the mole fraction of compounds "A" and "B":

$$A_N, A_{N-1}B_1, A_{N-2}B_2, A_{N-3}B_3, \ldots B_N$$

The stoichiometry of the oligomers can be generalized using the term "A_nB_{N-n}", where the value of N reflects oligomer size; the value of n reflects the number of "A" subunits; and the value of $N - n$ reflects the number of "B" subunits. For example, $N = 4$ and $n = 1$ for an A_1B_3 heterotetramer.

Three main factors influence the relative concentration of oligomers A_nB_{N-n} at equilibrium: multiplicity, free energy, and chemical potential. The equations developed by Collum and co-workers combine these factors for calculating the concentrations of homooligomers and heterooligomers.

1. **Multiplicity (*M_n*)**: The number of ways the "A" and "B" subunits can be arranged within an oligomer A_nB_{N-n}. Each unique arrangement is called a permutation (ρ). Oligomers that have multiple permutations are present in larger concentrations than oligomers that have only one. The multiplicity or the number of permutations of an oligomer A_nB_{N-n} can be determined with Pascal's triangle or by using binomial theorem, which is shown here:

$$M_n = \frac{N}{(N-n)! \times n!}$$

2. **Free Energy (g_ρ)**: The relative stability of an oligomer permutation ρ. Permutations with the same stoichiometry often have the same relative stability. [In the A_2B_2 heterotetramer of peptides **1a** and **1b**, the A·A/B·B and A·B/A·B topological isomers do not have the same relative stability.] The variable $\phi_{N,n}$ relates the free energy of each permutation to the relative stability.

$$-g_\rho = kT\ln(\phi_{N,n})$$

3. **Chemical Potential (μ_A and μ_B)**: The potential energy associated with the moles of compound "A" and the moles of compound "B" in a mixture. The mole fraction of the compounds reflects the relative chemical potential. The relative chemical potential of μ_A and μ_B shifts as the mole fraction of A and B is varied in a Job's method of continuous variation experiment. For a mixture of tetramers, when the mole fraction of A is greater than the mole fraction of B, the concentration the A_3B_1 heterotetramer is greater than the concentration of the A_1B_3 heterotetramer.

To calculate the concentration of a permutation, the free energy and the chemical potential terms are combined to give the following equation:

$$[\,\rho\,] = C \times \exp\left(\frac{\text{-g}_{\rho} + n_{\rho}\mu_{\mathrm{A}} + (N - n_{\rho})\mu_{\mathrm{B}}}{k\mathrm{T}}\right) \tag{1}$$

The free energy g_{ρ} is the measure of the relative stability of the corresponding permutation ρ; the value of n_{ρ} is the number of the "A" subunits within the permutation ρ; the value of μ_{A} is the chemical potential of compound A; the value of μ_{B} is the chemical potential of compound B. The constant *C* relates oligomerization propensity to the total concentration.

The concentration of an oligomer $\mathrm{A}_n\mathrm{B}_{N-n}$ is the sum of the concentrations of permutations that have the same stoichiometry ($\rho;n_{\rho} = n$). For calculating the concentration of an oligomer $\mathrm{A}_n\mathrm{B}_{N-n}$, the multiplicity term is combined with the free energy term and chemical potential term to give the following equation:

$$[\mathrm{A}_n\mathrm{B}_{N-n}] = \sum_{\rho;n_{\rho} = n} [\,\rho\,] = \; C \times \exp\left(\frac{n_{\rho}\mu_{\mathrm{A}} + (N - n_{\rho})\mu_{\mathrm{B}}}{k\mathrm{T}}\right) \times \sum_{\rho;n_{\rho} = n} \exp\left(\frac{\text{-g}_{\rho}}{k\mathrm{T}}\right) \tag{2}$$

$$= \; C \times \exp\left(\frac{n_{\rho}\mu_{\mathrm{A}} + (N - n_{\rho})\mu_{\mathrm{B}}}{k\mathrm{T}}\right) \times \mathrm{M}_n \times \langle \exp\left(\frac{\text{-g}_{\rho}}{k\mathrm{T}}\right)\rangle_{\rho;n_{\rho} = n} \tag{3}$$

In this equation, the concentrations of permutations ρ that have the same stoichiometry ($\rho;n_{\rho} = n$) are multiplied by the multiplicity M_n to give the oligomer concentration $[\mathrm{A}_n\mathrm{B}_{N-n}]$. In a Job's method of continuous variation experiment, the sum of the concentrations of permutations ρ that have the same stoichiometry ($\rho;n_{\rho} = n$) gives the oligomer concentration $[\mathrm{A}_n\mathrm{B}_{N-n}]$.

To simplify equation (3), the variables a and b were used to represent the effective chemical potentials μ_A and μ_B, and the variable $\phi_{N,n}$ was used to represent the relative stability of an oligomer A_nB_{N-n}.

$$a = \exp\left(\frac{\mu_A}{kT}\right) \qquad b = \exp\left(\frac{\mu_B}{kT}\right) \qquad \phi_{N,n} = \langle \exp\left(\frac{-g_\rho}{kT}\right)\rangle_{\rho;n_\rho = n}$$

Incidentally, the values of a and b are related to each other such that

$$a + b = 1 \text{ and } \frac{a}{b} = \frac{a}{1-a}$$

Incorporation of these variables into equation (3) gives the following equation:

$$[A_nB_{N-n}] = C \times M_n \times \phi_{N,n} \times a^n \times b^{N-n} \tag{4}$$

Equation (4) is the general equation for calculating oligomer concentration. To calculate the relative concentration of an oligomer, the concentration is divided by the sum of the concentrations of all the oligomers A_jB_{N-j} :

$$\frac{[A_nB_{N-n}]}{\sum_{j=0}^{N}[A_jB_{N-j}]} = \frac{C \times M_n \times \phi_{N,n} \times a^n \times b^{N-n}}{\sum_{j=0}^{N} C \times M_j \times \phi_{N,j} \times a^j \times b^{N-j}} \tag{5}$$

B. Equations for Homotetramers and Heterotetramers

In this section, we describe the equations for homotetramers and heterotetramers ($N = 4$). Heterotetramers have multiple permutations ρ, which increases the concentrations of the heterotetramers relative to the concentrations of the homotetramers. Table S3 summarizes the permutations ρ of the homotetramers and heterotetramers.

Table S3. Permutations ρ of the homotetramers and heterotetramers

stoichiometry	multiplicity	permutation
A_nB_{N-n}	M_n	ρ
A_4	1	AAAA
A_3B_1	4	AAAB, AABA, ABAA, AAAB
A_2B_2	6	AABB, ABAB, BAAB, BABA, BBAA, ABBA
A_1B_3	4	ABBB, BABB, BBAB, BBBA
B_4	1	BBBB

The parameters $\phi_{N,n}$ are ascribed to each of the homotetramers and heterotetramers, where the N and n are integers in which the value of N describes the oligomer size and the value of n describes the number of "A" subunits. The value of each $\phi_{N,n}$ reflects the relative stability of each homotetramer or heterotetramer. The parameters $\phi_{4,4}$, $\phi_{4,3}$, $\phi_{4,2}$, $\phi_{4,1}$, and $\phi_{4,0}$ describe the relative stabilities of A_4, A_3B_1, A_2B_2, A_1B_3, and B_4, respectively. The following equations are based on equation (4) and contain these parameters for calculating the concentrations of each homotetramer and heterotetramer:

$$[A_4] = 1 \times C \times \phi_{4,4} \times a^4 \tag{6}$$

$$[A_3B_1] = 4 \times C \times \phi_{4,3} \times a^3b^1 \tag{7}$$

$$[A_2B_2] = 6 \times C \times \phi_{4,2} \times a^2b^2 \tag{8}$$

$$[A_1B_3] = 4 \times C \times \phi_{4,1} \times a^1b^3 \tag{9}$$

$$[B_4] = 1 \times C \times \phi_{4,0} \times b^4 \tag{10}$$

The following equation calculates the relative integration ($I_{N,n}$) by dividing the integration of one tetramer by the sum of the integrations of all tetramers.

$$I_{N,n} = \frac{C \times M_n \times \phi_{N,n} \times a^n \times b^{N-n}}{\sum_{j=0}^{N} C \times M_j \times \phi_{N,j} \times a^j \times b^{N-j}} \tag{11}$$

The following equations calculate the relative integration of each homotetramer and heterotetramer:

$$I_{4,4} = \frac{\phi_{4,4}a^4}{\phi_{4,4}a^4 + 4\phi_{4,3}a^3b^1 + 6\phi_{4,2}a^2b^2 + 4\phi_{4,1}a^1b^3 + \phi_{4,0}b^4} \tag{12}$$

$$I_{4,3} = \frac{4\phi_{4,3}a^3b^1}{\phi_{4,4}a^4 + 4\phi_{4,3}a^3b^1 + 6\phi_{4,2}a^2b^2 + 4\phi_{4,1}a^1b^3 + \phi_{4,0}b^4} \tag{13}$$

$$I_{4,2} = \frac{6\phi_{4,2}a^2b^2}{\phi_{4,4}a^4 + 4\phi_{4,3}a^3b^1 + 6\phi_{4,2}a^2b^2 + 4\phi_{4,1}a^1b^3 + \phi_{4,0}b^4} \tag{14}$$

$$I_{4,1} = \frac{4\phi_{4,1}a^1b^3}{\phi_{4,4}a^4 + 4\phi_{4,3}a^3b^1 + 6\phi_{4,2}a^2b^2 + 4\phi_{4,1}a^1b^3 + \phi_{4,0}b^4} \tag{15}$$

$$I_{4,0} = \frac{\phi_{4,0}b^4}{\phi_{4,4}a^4 + 4\phi_{4,3}a^3b^1 + 6\phi_{4,2}a^2b^2 + 4\phi_{4,1}a^1b^3 + \phi_{4,0}b^4} \tag{16}$$

C. Equations for Monomers, Homotetramers, and Heterotetramers

In this section, we describe how we modified the equations to accommodate the equilibrium of the monomers with the homotetramers and heterotetramers. The result is the equation used for the monomer–homotetramer–heterotetramer equilibrium model for nonlinear least-squares fitting of the Job plot.

$$\mathrm{A},\ \mathrm{B},\ \mathrm{A}_4,\ \mathrm{A}_3\mathrm{B}_1,\ \mathrm{A}_2\mathrm{B}_2,\ \mathrm{A}_1\mathrm{B}_3,\ \mathrm{B}_4$$

We used the following equations to calculate the concentrations of the monomers as a function of their respective relative stabilities $\phi_{N,n}$ and the chemical potentials a and b.

$$[\mathrm{A}] = C \times \phi_{1,1} \times a \tag{17}$$

$$[\mathrm{B}] = C \times \phi_{1,0} \times b \tag{18}$$

We used the mass balance equation to accommodate the total concentration of compounds A and B. The total concentration has little or no effect on the equilibria among homotetramers and heterotetramers. By contrast, the total concentration is critical in the equilibria of the monomers with the homotetramers and heterotetramers. The mass balance equation gives the total concentration of compounds A and B ($[\mathrm{A}]_{\mathrm{total}}$ and $[\mathrm{B}]_{\mathrm{total}}$) as a function of the monomers, homotetramers, and heterotetramers.

$$[\mathrm{A}]_{\mathrm{total}} + [\mathrm{B}]_{\mathrm{total}} = [\mathrm{A}] + [\mathrm{B}] + 4([\mathrm{A}_4] + [\mathrm{A}_3\mathrm{B}_1] + [\mathrm{A}_2\mathrm{B}_2] + [\mathrm{A}_1\mathrm{B}_3] + [\mathrm{B}_4]) \tag{19}$$

Substitution of equations (6), (7), (8), (9), (10), (17), and (18) into the mass balance equation gives the following equation:

$$[\mathrm{A}]_{\mathrm{total}} + [\mathrm{B}]_{\mathrm{total}} = C(\phi_{1,1}\, a + \phi_{1,0}\, b) + 4C(\phi_{4,4}\, a^4 + 4\phi_{4,3}\, a^3b^1 + 6\phi_{4,2}\, a^2b^2 + 4\phi_{4,1}\, a^1b^3 + \phi_{4,0}\, b^4) \tag{20}$$

Equation (20) was simplified using the following identities, which represent a and b in terms of α and 1 − α:

$$\alpha = a / (a + b) \qquad x = a + b \qquad a = \alpha\, x \qquad b = (1 - \alpha)x$$

Substitution of $a = \alpha x$ and $b = (1 - \alpha)x$ into equation (20) gives the following equation:

$$[\mathrm{A}]_{\mathrm{total}} + [\mathrm{B}]_{\mathrm{total}} = xC(\phi_{1,1}\, \alpha + \phi_{1,0}\, (1 - \alpha)) + 4x^4C(\phi_{4,4}\, \alpha^4 + 4\phi_{4,3}\, \alpha^3(1 - \alpha)^1 + 6\phi_{4,2}\, \alpha^2(1 - \alpha)^2 + 4\phi_{4,1}\, \alpha^1(1 - \alpha)^3 + \phi_{4,0}\, (1 - \alpha)^4) \tag{21}$$

Equation (21) was simplified by representing the concentrations of the monomers and tetramers in terms of $\mathrm{M}_{\mathrm{total}}$ and $\mathrm{T}_{\mathrm{total}}$:

$$\mathrm{M}_{\mathrm{total}} = xC(\phi_{1,1}\, \alpha + \phi_{1,0}\, (1 - \alpha))$$

$$\mathrm{T}_{\mathrm{total}} = 4x^4C(\phi_{4,4}\, \alpha^4 + 4\phi_{4,3}\, \alpha^3(1 - \alpha)^1 + 6\phi_{4,2}\, \alpha^2(1 - \alpha)^2 + 4\phi_{4,1}\, \alpha^1(1 - \alpha)^3 + \phi_{4,0}\, (1 - \alpha)^4)$$

Substitution of $\mathrm{M}_{\mathrm{total}}$ and $\mathrm{T}_{\mathrm{total}}$ into the mass balance equation gives the equation for a monomer−tetramer (monomer−homotetramer−heterotetramer) equilibrium model:

$$[\mathrm{A}]_{\mathrm{total}} + [\mathrm{B}]_{\mathrm{total}} = x\, \mathrm{M}_{\mathrm{total}} + 4x^4\, \mathrm{T}_{\mathrm{total}} \tag{22}$$

Setting the equation equal to zero gives the following fourth-order polynomial:

$$x\, \mathrm{M}_{\mathrm{total}} + 4x^4\, \mathrm{T}_{\mathrm{total}} - ([\mathrm{A}]_{\mathrm{total}} + [\mathrm{B}]_{\mathrm{total}}\,) = 0 \tag{23}$$

The fourth-order polynomial was solved for x using Mathematica 10.3 (Wolfram Research, Champaign, IL), which gave a set of four roots (not shown). Each root was evaluated under typical conditions of monomer and tetramer equilibrium (e.g. $\mathrm{M}_{\mathrm{total}} = 1.4$, $\mathrm{T}_{\mathrm{total}} = 1.65$, and $([\mathrm{A}]_{\mathrm{total}} + [\mathrm{B}]_{\mathrm{total}}) = 8$). The root that gave a non-negative value of x was used as the monomer–homotetramer–heterotetramer equilibrium model.

IV. Nonlinear Least-Squares Fitting of the Job Plot

This section describes how we used the monomer–homotetramer–heterotetramer equilibrium model for nonlinear least-squares fitting of the Job Plot. To perform the fit, the model was incorporated into a .m script and executed with a series of scripts in MATLAB 2015b. The scripts are based on those developed by Collum and co-workers in the supporting information of Liou, L. R; McNeil, A. J.; Ramirez, A.; Toombes, G. E. S.; Gruver, J. M.; Collum, D. B. *J. Am. Chem. Soc.* **2008**, *130*, 4859–4868.

The following subsections describe the process of fitting the model to our experimental data (Table S2) from the Job's method of continuous variation experiment. The subsections also contain the code from each script along with annotations that describe how the code is used.

> The code is indicated by bracketing the code text with bars along the left- and right-hand side, as shown here.

Annotations for the code appear in between the bracketed text, as shown here.

A. End User Instructions

1. Copy the code from each subsection into its own text file, but do not transfer the annotations. Save each file into the same folder or directory using the following file names:

   data_Monomer_Tetramer.m

   try_fit.m

   refine_fit.m

   multimers.m

   populations_tetramer.m

   populations_monomer.m

   error_of_model.m

2. Open MATLAB and navigate the “Current Folder” of the program to the directory where the .m scripts were saved.

3. Load the data from the data_Monomer_Tetramer.m file into MATLAB. The data can be loaded in one of two ways: by opening the script with MATLAB and clicking “Run” in the window or by typing the file name into the MATLAB command line and pushing enter.

4. Run the try_fit.m script. This script can be run with or without Expt_Errors. To run the script without Expt_Errors, type the following try_fit function into the command line and push enter:

   ```
   try_fit(Xb, Ctotal, phi_monomer, peak_assignment_monomer, phi_tetramer,
   peak_assignment_tetramer, Expt_Populations)
   ```

   To run the try_fit.m script with Expt_Errors, type the following try_fit function into the command line and push enter:

   ```
   try_fit(Xb, Ctotal, phi_monomer, peak_assignment_monomer, phi_tetramer,
   peak_assignment_tetramer, Expt_Populations, Expt_Errors)
   ```

5. The data points in the figure should resemble the data shown in Figure 8. The curves that overlay the data should resemble the curves from the simulated Job plot in Figure 9e.

6. To run the refine_fit.m script without Expt_Errors, type the following refine_fit function into the command line and push enter:

   ```
   refine_fit(Xb,Ctotal,phi_monomer, peak_assignment_monomer, phi_tetramer,
   peak_assignment_tetramer, Expt_Populations, phi_constant)
   ```

   To run the refine_fit.m script with Expt_Errors, type the following refine_fit function into the command line and push enter:

   ```
   refine_fit(Xb,Ctotal,phi_monomer, peak_assignment_monomer, phi_tetramer,
   peak_assignment_tetramer, Expt_Populations, phi_constant, Expt_Errors)
   ```

   After the fit, the final phi values shown in the MATLAB terminal are the optimized phi values. These values should be comparable the phi values listed in Figure 8.

7. Make a copy of the file data_Monomer_Tetramer.m to a new file called data_Monomer_Tetramer_new.m. Replace the phi values with the optimized values from the refine_fit.m script.

8. Load the new data file data_Monomer_Tetramer_new.m into MATLAB. Type the try_fit function into the command line and push enter to observe the optimized fit.

B. Definitions

1. **Xb(j)** is the mole fraction χ_B.

2. **Ctotal** is the input for the total concentration of each mixture.

3. **Expt_Populations** is the experimental data input for the relative integrations of the monomers, homotetramers, and heterotetramers (Table S2). These data are referred to as the "experimental populations".

4. **Expt_Errors** is the input for the error of the measurements.

5. **peak_assignment_monomer** and **peak_assignment_tetramer** are column identifiers assigned to each monomer and tetramer population.

6. **phi_monomer** and **phi_tetramer** are measures of the relative stabilities of the monomers and tetramers. These values are assigned to each monomer, homotetramer, and heterotetramer population and are used by the monomer–homotetramer–heterotetramer equilibrium model for calculating the relative concentrations of each population.

7. **phi_constant** is the input that dictates whether a phi_monomer or phi_tetramer value remains fixed or is allowed to vary during nonlinear least-squares fitting. A value of 1 allows the corresponding phi to vary; a value of 0 keeps the corresponding phi fixed.

8. **Expt_weights** is the input for the error of each data point and is used for weighting the error of each data point. The data points are weighted equally if nothing is entered.

9. **conc_monomer** and **conc_tetramer** are used for calculating and storing the concentrations for each monomer, homotetramer, and heterotetramer population. [Note that even though the term concentration is used, the scripts are actually calculating the integrations of the monomer and tetramer populations.]

10. **pop_monomer** and **pop_tetramer** are used to temporarily store calculated values for the concentrations (relative integrations) of each monomer or tetramer population. These data are referred to as the calculated (predicted) populations.

11. **Model_Populations** is the final output for the concentrations (relative integrations) of the monomers, homotetramers, and heterotetramers.

12. **mean_error** weighted standard deviation of the residuals over the entire fit.

13. **pop_error(1,j)** is the mean error of experimental populations – calculated populations. The value could be negative, zero, or positive.

14. **pop_error(2,j)** is the root mean square error of experimental populations – calculated populations. The value is always positive.

15. **phi_dimer_new** and **phi_tetramer_new** are the new values of each phi after the fit.

16. **error** is the root mean square error of the new calculated populations.

C. Monomers, Homotetramers, and Tetramers: Data

This script stores the experimental populations and the initial values for performing the fit.

This code clears all stored information in the command line and closes all figures.

```
clear variables;
close all;
clc;
```

This code is the input for the total concentration of the mixtures, which is designated Ctotal. The number of values in Ctotal equal the number of samples studied. In this case, nine samples were studied.

```
Ctotal = [0.008 0.008 0.008 0.008 0.008 0.008 0.008 0.008 0.008];
```

This code is the input the mole fraction χ_B, which is designated Xb.The values entered in Xb equal the mole fraction of each mixture studied. The values are listed from lowest to highest.

```
Xb = [0.00 0.125 0.25 0.375 0.50 0.625 0.75 0.875 1.00];
```

This code is the input for the experimental populations: the relative integrations from the Job's method of continuous variation experiment (Table S2). Each column lists the relative integrations of the monomer and tetramer populations in the following order: A, B, A_4, A_3B_1, A_2B_2, A_1B_3, B_4. The columns are separated by a space. Each row lists the relative integrations of the mole fractions Xb listed in the following order: 0.00, 0.125, 0.25, 0.375, 0.50, 0.625, 0.75, 0.875, and 1.00. The rows are separated by a semicolon.

```
Expt_Populations = [
0.0664  0.0000  0.9336  0.0000  0.0000  0.0000  0.0000;
0.0449  0.0608  0.6883  0.1183  0.0873  0.0003  0.0000;
0.0332  0.1089  0.4359  0.1459  0.2654  0.0102  0.0005;
0.0281  0.1445  0.2986  0.1391  0.3639  0.0216  0.0042;
0.0177  0.1987  0.1350  0.1057  0.4640  0.0581  0.0209;
0.0115  0.2699  0.0608  0.0695  0.4748  0.0751  0.0384;
0.0082  0.3095  0.0096  0.0260  0.3516  0.1486  0.1466;
0.0044  0.3820  0.0009  0.0060  0.1595  0.1731  0.2741;
0.0000  0.4796  0.0000  0.0000  0.0000  0.0000  0.5204;];
```

The Expt_Errors input is optional for the fit. These values should be listed for the monomer and tetramer populations in the following order: A, B, A_4, A_3B_1, A_2B_2, A_1B_3, B_4, which is the same order used for the populations in the Expt_Populations input.

```
% Expt_Errors = [;];
```

The peak_assignment_monomer and peak_assignment_tetramer are inputs that designate the column for each monomer or tetramer. For a given monomer or tetramer, the assignment value specifies which column the data should be read from or stored in. The peak_assignment_monomer values are listed in the following order: A, B; the peak_assignment_tetramer values are listed in the following order: A_4, A_3B_1, A_2B_2, A_1B_3, B_4.

```
peak_assignment_monomer = [1 5];
peak_assignment_tetramer = [ 1 2 3 4 5 ];
```

The initial phi values are all set to one.

The phi_monomer values are listed in the following order: A, B; the phi_tetramer values are listed in the following order: A_4, A_3B_1, A_2B_2, A_1B_3, B_4.

```
phi_monomer = [1 1];
phi_tetramer = [1 1 1 1 1];
```

The phi_constants that are set to 1 allow the phi value to be refined with the refine_fit.m script; the phi_constants that are set to 0 keep the value fixed during the refine_fit.m script.

The phi_constant values are listed in the following order: A, B, A_4, A_3B_1, A_2B_2, A_1B_3, B_4. The phi_constant for the A_4 homotetramer was fixed so that the refine_fit.m script gives a unique solution.

```
phi_constant = [1 1 0 1 1 1 1];
```

D. Monomers, Homotetramers, and Tetramers: Try Fit

This script plots the experimental populations, and also plots the populations calculated from the phi values. The two plots are overlayed in a new window. The experimental populations are plotted versus the mole fraction Xb as open circles; the calculated populations are plotted versus the mole fraction Xb as smooth lines. The script also determines the error between the experimental populations and the calculated populations then prints these values in the MATLAB terminal.

```
function try_fit(Xb, Ctotal,...
        phi_monomer, peak_assignment_monomer,...
        phi_tetramer, peak_assignment_tetramer,...
        Expt_Populations, Expt_Errors)
```

This code determines whether Expt_Errors are entered.

```
        if(nargin<8)
           Expt_weights=ones(size(Expt_Populations));
        else
           Expt_weights=1./(Expt_Errors+mean(mean(Expt_Errors)));
        end
```

This code plots the experimental populations of the monomers and tetramers from the Expt_Populations input.

```
        hold on ; cscheme= 'kybmgcrkybmgcr'; axis([0 1 0 1]); xlabel('X_B');
        ylabel('Relative Integration');
        for j=1:size(Expt_Populations,2)
           if (nargin<8)
              plot(Xb, Expt_Populations(:,j),sprintf('%so',cscheme(j)));
           else
              errorbar(Xb, Expt_Populations(:,j), Expt_Errors(:,j),sprintf('%so',cscheme(j)));
           end
        end
```

This code calculates the monomer and tetramer populations using the initial values of the phi's entered.

```
XBc = (0:0.01:1);
Ctotalac = Ctotal(1)*ones(size(XBc));
[conc_monomers, conc_tetramers] = multimers(XBc, Ctotalac,...
                                phi_monomer, phi_tetramer);
```

This code stores the calculated monomer and tetramer populations and stores them in a matrix.

```
pop_tetramer = populations_tetramer(conc_monomers,...
   peak_assignment_monomer, conc_tetramers, peak_assignment_tetramer);

pop_monomer = populations_monomer(conc_monomers,...
   peak_assignment_monomer, conc_tetramers, peak_assignment_tetramer);

pop_tetramer_corrected = pop_tetramer-pop_monomer;
pop_combined = horzcat(pop_monomer(:,1),...
   pop_monomer(:,5),pop_tetramer_corrected);
```

This code plots the calculated populations.

```
for j=1:size(pop_combined,2)
   plot(XBc,pop_combined(:,j),sprintf('%c',cscheme(j)) );
end
```

This code compares the experimental and the calculated populations, then calculates and displays the error.

```
[mean_error, pop_error] = error_of_model(Xb, Ctotal,...
   phi_monomer, peak_assignment_monomer,...
   phi_tetramer, peak_assignment_tetramer,...
   Expt_Populations, Expt_weights);

N = length(horzcat(phi_monomer, phi_tetramer)) - 1;
fprintf(1,'\nThe Mean mismatch is %f percent.\n\n', mean_error*100);

for j=1:size(pop_error, 2)
   fprintf(1,'Predicted value of Population %d exceeds measurement by %f percent\n
and mean square error of %f percent.\n\n',j, pop_error(1,j)*100,pop_error(2,j)*100);
end
```

E. Monomers, Homotetramers, and Tetramers: Refine Fit

This script performs the nonlinear least-squares fitting. The script optimizes the phi values to match the calculated populations to the experimental populations. The script reports the new phi values and the root mean square difference between the calculated and the experimental populations.

```
function [phi_dimer_new, phi_tetramer_new, error] = refine_fit(Xb,Ctotal,...
        phi_monomer, peak_assignment_monomer,...
        phi_tetramer, peak_assignment_tetramer,...
        Expt_Populations, phi_constant, Expt_Errors)
```

This code determines whether Expt_errors are entered.

```
    if (nargin<9)
       Expt_weights = ones(size(Expt_Populations));
    else
       Expt_weights = 1./( Expt_Errors + mean(mean(Expt_Errors)));
    end
```

This code merges the monomer Expt_Populations and the tetramer Expt_Populations into a single input.

```
    phimerge = [phi_monomer, phi_tetramer];
    idx_monomer = [1 2]; idx_tetramer = [3 4 5 6 7];
    param = [1:length(phimerge)];
```

This code sets the initial step size used to optimize the phi values; the initial step size is 10%.

```
    step_size = 0.1*phi_constant.*phimerge(param);
    N_no_progress = 0;
    N_max_trials = 30;
```

This code compares the calculated and experimental populations of the monomers and tetramers, then calculates and displays the error of the model.

```
[error_best, temp] = error_of_model(Xb,Ctotal,...
         phimerge(idx_monomer), peak_assignment_monomer,...
         phimerge(idx_tetramer), peak_assignment_tetramer,...
         Expt_Populations, Expt_weights);

      fprintf(1,'\n Initial Error of Fit = %f percent.\n', error_best * 100);
```

This code is a "for while" loop that reduces the error of the model by optimizing the phi values.

```
      while (N_no_progress < N_max_trials)
         flag = 0;
         for k=1:length(param)
```

This code adjusts the value of phi to the "right" and to the "left".

```
            phi_testr = phimerge;
            phi_testr(param(k))=abs(phimerge(param(k)) + step_size(k));
            [error_testr, temp] = error_of_model(Xb,Ctotal,...
               phi_testr(idx_monomer), peak_assignment_monomer,...
               phi_testr(idx_tetramer), peak_assignment_tetramer,...
               Expt_Populations, Expt_weights);

            phi_testl = phimerge;
            phi_testl(param(k))=abs(phimerge(param(k)) - step_size(k));
            [error_testl, temp] = error_of_model(Xb,Ctotal,...
               phi_testl(idx_monomer), peak_assignment_monomer,...
               phi_testl(idx_tetramer), peak_assignment_tetramer,...
               Expt_Populations, Expt_weights);
```

This code determines which adjustment of phi decreases the error of the model. If either the right or the left value decreases the error, then that value is stored and the loop repeats again. If neither the right or the left value decreases the error, then the step is flagged and the size of the step is reduced.

```
  if (error_testr < error_best)
    error_best = error_testr;
    phimerge = phi_testr;
    step_size(k) = step_size(k) * 1.5;
    N_no_progress = 0;

  elseif (error_testl < error_best)
    error_best = error_testl;
    phimerge = phi_testl;
    step_size(k) = step_size(k) * 1.5;
    N_no_progress = 0;

  else
    flag = flag + 1;
  end
end

if (flag >= length(param))
  step_size = step_size * (0.75 + 0.25 * rand);
  N_no_progress = N_no_progress + 1;
end
```

This code displays the new fit after each phi has been adjusted

```
fprintf(1,'\n\n Error - %f , Last Good Step - %d , Mean Step Size - %f'...
,error_best, N_no_progress, 100*mean(step_size) );
fprintf('\n     Phi Monomer - '); fprintf(1,'%f ', phimerge(idx_monomer));
fprintf(1,'\n     Phi Tetramer - '); fprintf(1,'%f ',phimerge(idx_tetramer));

end

error = error_best;
phi_monomer_new = phimerge(idx_monomer);
phi_tetramer_new = phimerge(idx_tetramer);
```

F. Monomers, Homotetramers, and Tetramers: Multimers

For each mole fraction Xb, this script calculates the concentrations (relative integrations) of the monomer and tetramer populations using the inputs: Xb, Ctotal, phi_monomer, and phi_tetramer.

```
function [conc_monomer, conc_tetramer] = multimers(Xb, Ctotal,...
       phi_monomer, phi_tetramer)

   for j=1:length(Xb)
      [conc_monomer(j,:), conc_tetramer(j,:)] = bisect(Xb(j), Ctotal(j),...
        phi_monomer, phi_tetramer);
   end
```

This code is the bisection function, which optimizes the "relative chemical potential" until the value reflects the mole fraction of the experimental mole fraction.

```
function [conc_monomer, conc_tetramer] = bisect(Xb, Ctotal,...
        phi_monomer, phi_tetramer)

   tolerance = 1e-6;
   bmax = 1; bmin = 0;
   [Xmin, conc_monomer, conc_tetramer] = Cparametric(bmin,...
        phi_monomer, phi_tetramer, Ctotal);
   [Xmax, conc_monomer, conc_tetramer] = Cparametric(bmax,...
        phi_monomer, phi_tetramer, Ctotal);

   while ((Xmax - Xb) > tolerance)
        btest = (bmin + bmax) / 2;
   [Xtest, conc_monomer, conc_tetramer] = Cparametric(btest, phi_monomer,...
        phi_tetramer, Ctotal);

      if (Xtest > Xb)
         bmax = btest; Xmax = Xtest;
      else
         bmin = btest; Xmin = Xtest;
      end
end
```

This code is the mathematical model for the monomer and tetramer equilibrium.

```
function [Xb, conc_monomer, conc_tetramer] = Cparametric(b, phi_monomer,...
        phi_tetramer, Ctotal)

        a = 1 - b;

        Tscale = 1e9;

        Mtotal = (phi_monomer(1)*a + phi_monomer(2)*b);
        Ttotal = Tscale * (phi_tetramer(1) * a^4 + 4 * phi_tetramer(2) * a^3 * b +...
           6 * phi_tetramer(3) * a * a * b * b + 4 * phi_tetramer(4) * a * b^3 +...
           phi_tetramer(5)*b^4);


        Chi = ((1/2)*sqrt(-((9*Ttotal*Mtotal^2+sqrt(3)*sqrt(27*Ttotal^2*Mtotal^4+...
           1024*Ttotal^3*Ctotal^3))^(1/3)/(2*6^(2/3)*Ttotal))+Mtotal/(sqrt(2)*...
           Ttotal*sqrt((9*Ttotal*Mtotal^2+sqrt(3)*sqrt(27*Ttotal^2*Mtotal^4+1024*...
           Ttotal^3*Ctotal^3))^(1/3)/(6^(2/3)*Ttotal)-(4*2^(2/3)*Ctotal)/(3^(1/3)*...
           (9*Ttotal*Mtotal^2+sqrt(3)*sqrt(27*Ttotal^2*Mtotal^4+1024*Ttotal^3*...
           Ctotal^3))^(1/3))))+(2*2^(2/3)*Ctotal)/(3^(1/3)*(9*Ttotal*Mtotal^2+...
           sqrt(3)*sqrt(27*Ttotal^2*Mtotal^4+1024*Ttotal^3*Ctotal^3))^(1/3)))-...
           sqrt((9*Ttotal*Mtotal^2+sqrt(3)*sqrt(27*Ttotal^2*Mtotal^4+1024*Ttotal^3*...
           Ctotal^3))^(1/3)/(6^(2/3)*Ttotal)-(4*2^(2/3)*Ctotal)/(3^(1/3)*(9*Ttotal*...
           Mtotal^2+sqrt(3)*sqrt(27*Ttotal^2*Mtotal^4+1024*Ttotal^3*Ctotal^3))^(1/3)))/...
           (2*sqrt(2)));

        conc_monomer = Chi/Ctotal*[phi_monomer(1)*a, phi_monomer(2)*b];

        conc_tetramer = Tscale * 4 * Chi^4/Ctotal*[phi_tetramer(1)*a*a*a*a,...
           4*phi_tetramer(2)*a*a*a*b, 6*phi_tetramer(3)*a*a*b*b,...
           4*phi_tetramer(4)*a*b*b*b, phi_tetramer(5)*b*b*b*b];


        Xb = sum(conc_monomer.*[0 1])+sum(conc_tetramer.*[0 0.25 0.5 0.75 1]);
```

G. Monomers, Homotetramers, and Tetramers: Tetramer Populations

For all mole fractions Xb in the calculation, the populations_tetramer script stores all of the calculated populations for the tetramers.

```
function result = populations_tetramer(conc_monomer, peak_assignment_monomer,...
   conc_tetramer, peak_assignment_tetramer)

   result = zeros(size(conc_monomer,1), max(max(peak_assignment_monomer),...
   max(peak_assignment_tetramer)));
     N = size(conc_monomer,2);
      for j=1:N
         idx = peak_assignment_monomer(j);
         result(:,idx) = result(:,idx) + conc_monomer(:,j);
      end

      N = size(conc_tetramer,2);

      for j=1:N
         idx = peak_assignment_tetramer(j);
         result(:,idx) = result(:,idx) + conc_tetramer(:,j);
      end
```

H. Monomers, Homotetramers, and Tetramers: Monomer Populations

For all mole fractions Xb in the calculation, the Populations_monomer script stores all of the calculated populations for the tetramers.

```
function pop_monomer = populations_monomer(conc_monomer,...
   peak_assignment_monomer, conc_tetramer, peak_assignment_tetramer)

   result = zeros(size(conc_monomer,1), max(max(peak_assignment_monomer),...
      max(peak_assignment_tetramer)));
   N = size(conc_monomer,2);
   for j=1:N
      idx = peak_assignment_monomer(j);
      result(:,idx) = result(:,idx) + conc_monomer(:,j);
   end

   pop_monomer = result;
   N = size(conc_tetramer,2);

   for j=1:N
      idx = peak_assignment_tetramer(j);
      result(:,idx) = result(:,idx) + conc_tetramer(:,j);
   end
```

I. Monomers, Homotetramers, and Tetramers: Error of Model

This script is called within the try_fit.m and in the refine_fit.m scripts. The script reports the weighted mean error and population error of the fit.

```
function [mean_error, pop_error] = error_of_model(Xb, Ctotal,...
        phi_monomer, peak_assignment_monomer,...
        phi_tetramer, peak_assignment_tetramer,...
        Expt_Populations, Expt_Errors)

          if (nargin<8)
             Expt_weights=ones(size(Expt_Populations));
          else
             Expt_weights = 1./(Expt_Errors + mean(mean(Expt_Errors)));
          end

          [conc_monomers, conc_tetramers] = multimers(Xb,...
             Ctotal, phi_monomer, phi_tetramer);

          pop_tetramer = populations_tetramer(conc_monomers,...
             peak_assignment_monomer, conc_tetramers, peak_assignment_tetramer);

          pop_monomer = populations_monomer(conc_monomers,...
             peak_assignment_monomer, conc_tetramers, peak_assignment_tetramer);

          pop_tetramer_corrected = pop_tetramer-pop_monomer;

          Model_Populations = horzcat(pop_monomer(:,1),...
             pop_monomer(:,5),pop_tetramer_corrected);

          sizeof_Model = size(Model_Populations);
          sizeof_Expt = size(Expt_Populations);

          Model_Populations_Combined = horzcat(conc_monomers, conc_tetramers);

          sizeof_Model_Combined = size(Model_Populations_Combined);

          diff = Model_Populations_Combined-Expt_Populations;

          mean_error = sqrt(sum(sum(diff.*diff.*Expt_weights)) / sum(sum(Expt_weights)));

          pop_error = sum(diff.*Expt_weights,1) ./ sum(Expt_weights,1);
          pop_error(2,:) = sqrt(sum(diff.*diff.*Expt_weights,1) ./ sum(Expt_weights,1));
```

V. References

1. Truex, N. L.; Wang, Y.; Nowick, J. S. *J. Am. Chem. Soc.* **2016**. DOI: 10.1021/jacs.6b06000.

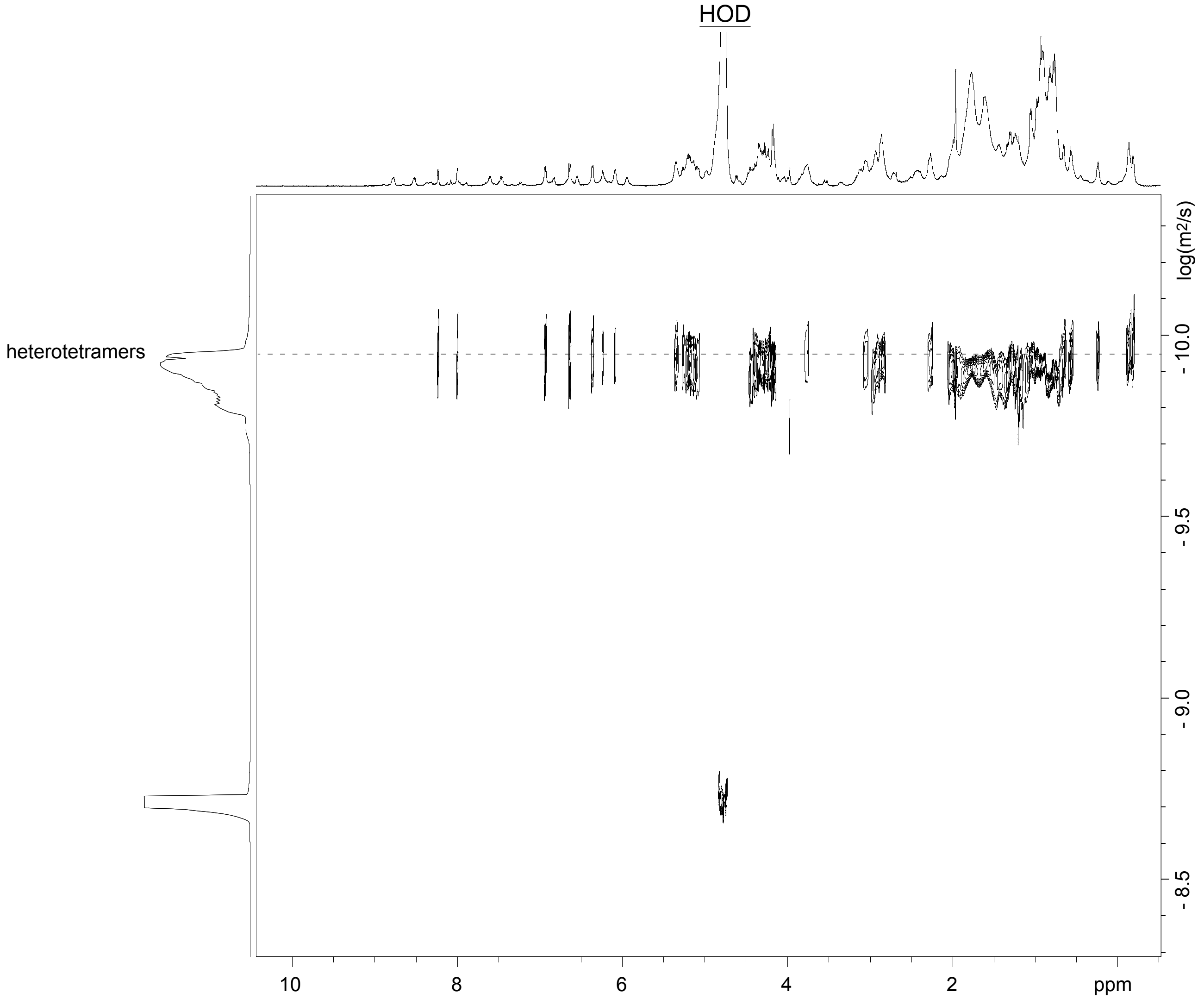
2. Podlech, J.; Gurrath, M.; Müller, G. 9-Fluorenylmethoxycarbonyl Group. In *Houben-Weyl Methods of Organic Chemistry*, Goodman, M., Ed. Thieme: Stuttgart, 2003; Vol. E22a, p 61.

3. Longsworth, L. G. *J. Phys. Chem.* **1960**, *64*, 1914–1917.

4. Nowick, J. S.; Khakshoor, O.; Hashemzadeh, M.; Brower, J. O. *Org. Lett.* **2003**, *5*, 3511–3513.

VI. Characterization Data

^1H NMR DOSY of a 1:1 mixture of peptides **1a** and **1b**
8.0 mM total concentration in D_2O at 500 MHz and 298 K



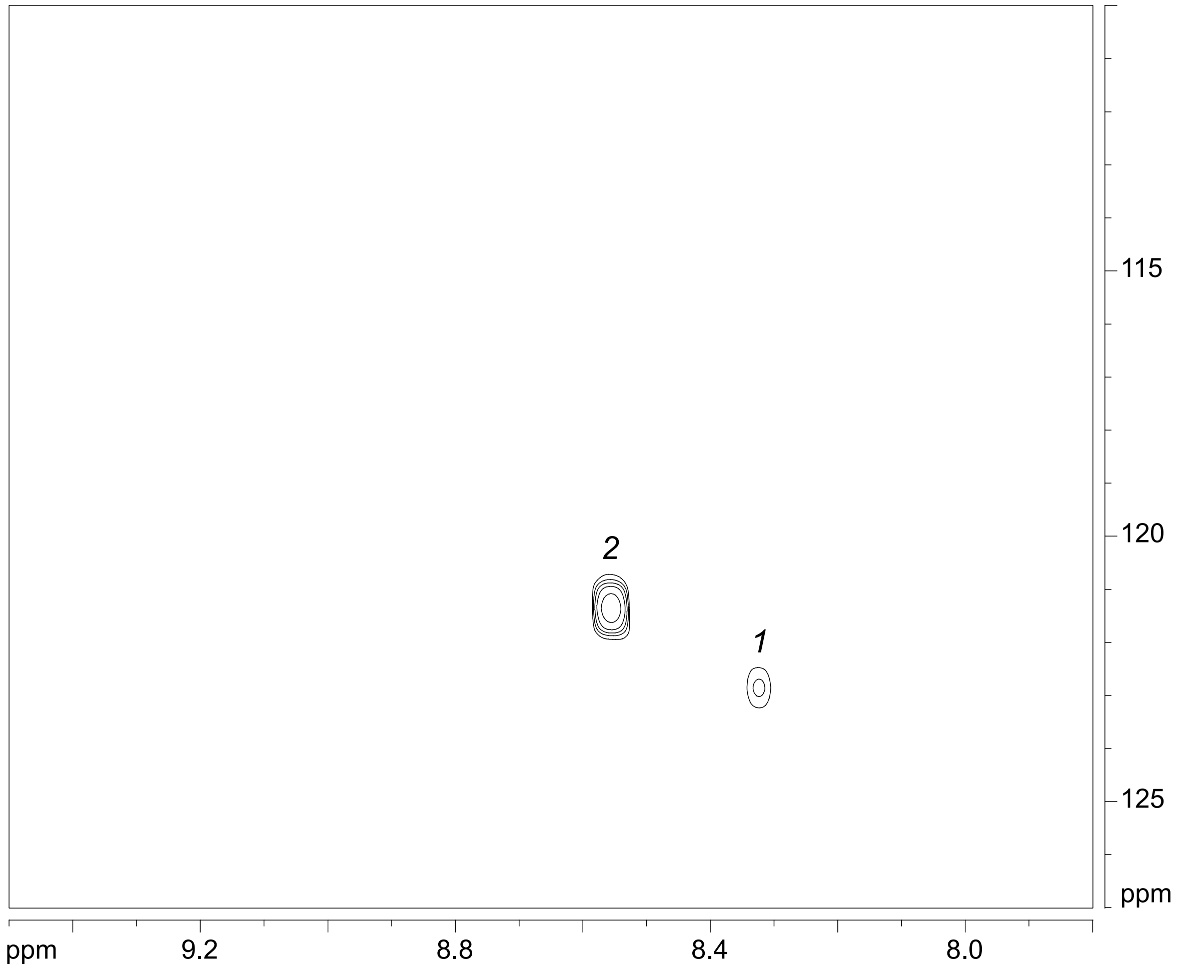
Calculations for the 1:1 mixture of peptides **1a** and **1b** at 8.0 mM total concentration

D_{HOD} = 19.0 x 10^{-10} m^2/s [a]
log (D_{HOD}) = -8.721

$D_{\mathrm{heterotetramers}}$: log($D$) = -9.943; D = $10^{-9.943}$ = 11.4 ± 1.1 x 10^{-11} m^2/s

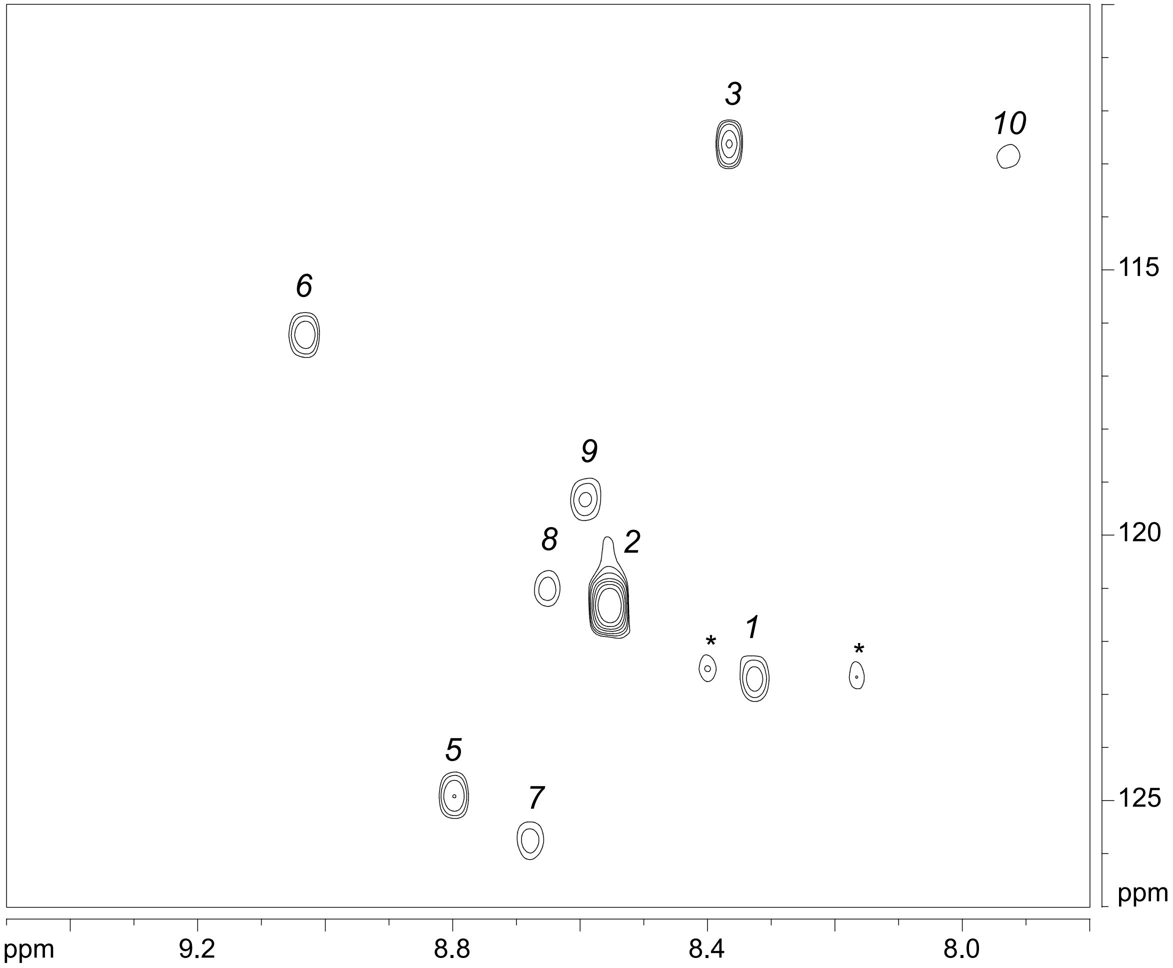
[a] Longsworth, L. G. *J. Phys. Chem.* **1960**, *64*, 1914−1917.

$^1H,^{15}N$ HSQC of peptides [^{15}N]**1a** and [^{15}N]**1b** at 600 MHz and 293 K
$\chi_B{}^a$ = 0.00; 8.0 mM total concentration in 9:1 H_2O/D_2O



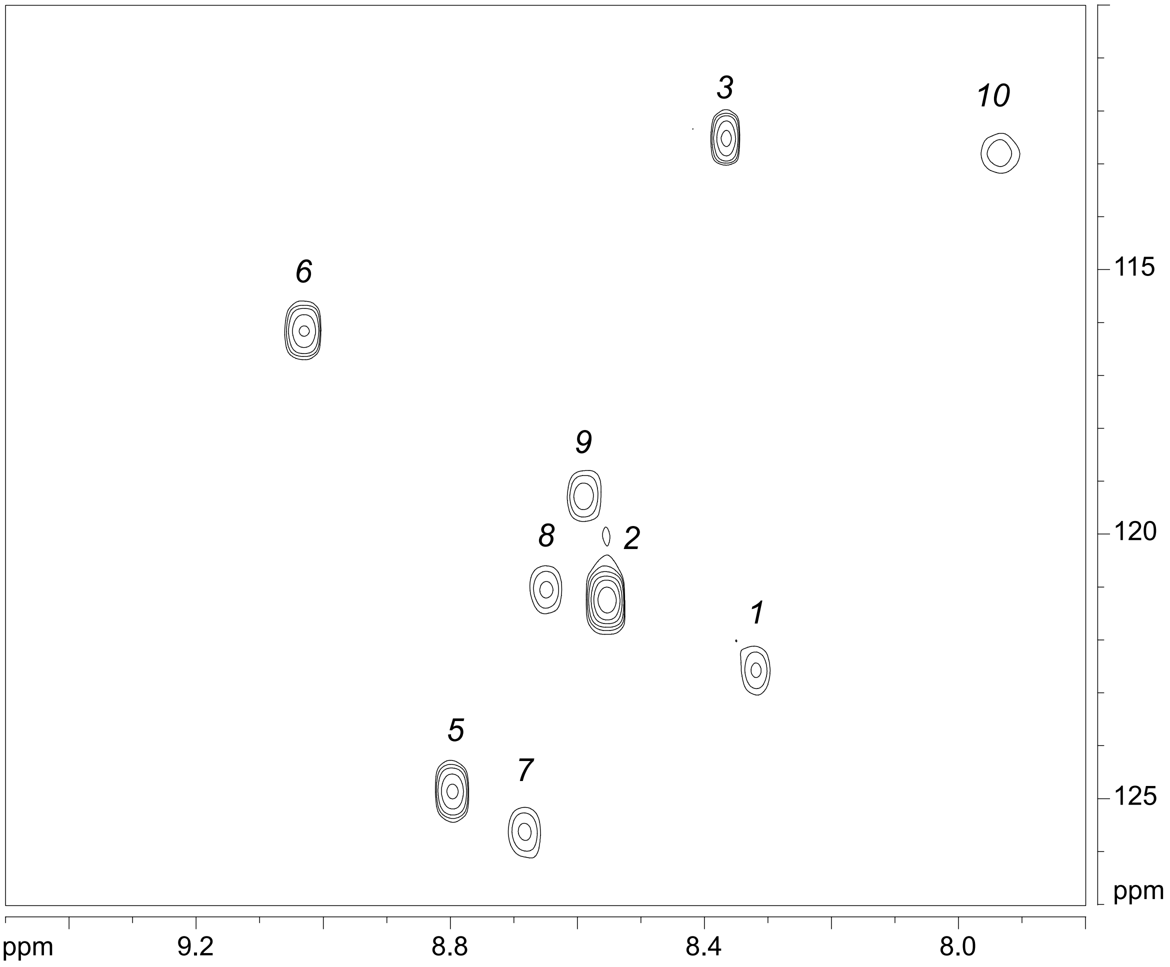
^{a} χ_B designates the mole fraction of peptide [^{15}N]**1b**.

^{1}H,^{15}N HSQC of peptides [^{15}N]**1a** and [^{15}N]**1b** at 600 MHz and 293 K
$\chi_B{}^a$ = 0.125; 8.0 mM total concentration in 9:1 H_2O/D_2O



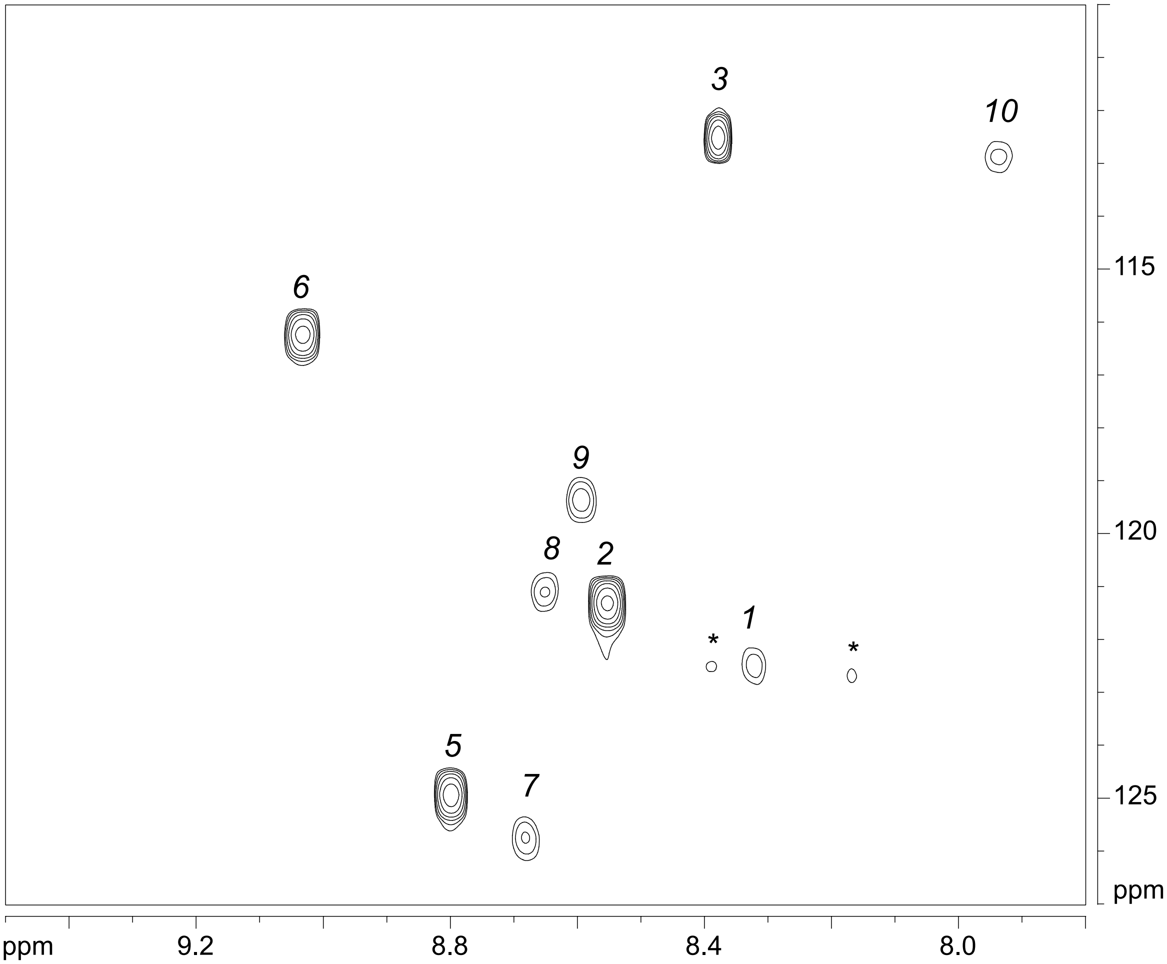
^{a} χ_B designates the mole fraction of peptide [^{15}N]**1b**.
The asterisks (*) indicate crosspeaks associated with minor unidentified species.

^{1}H,^{15}N HSQC of peptides [^{15}N]**1a** and [^{15}N]**1b** at 600 MHz and 293 K
$\chi_B{}^a$ = 0.25; 8.0 mM total concentration in 9:1 H_2O/D_2O



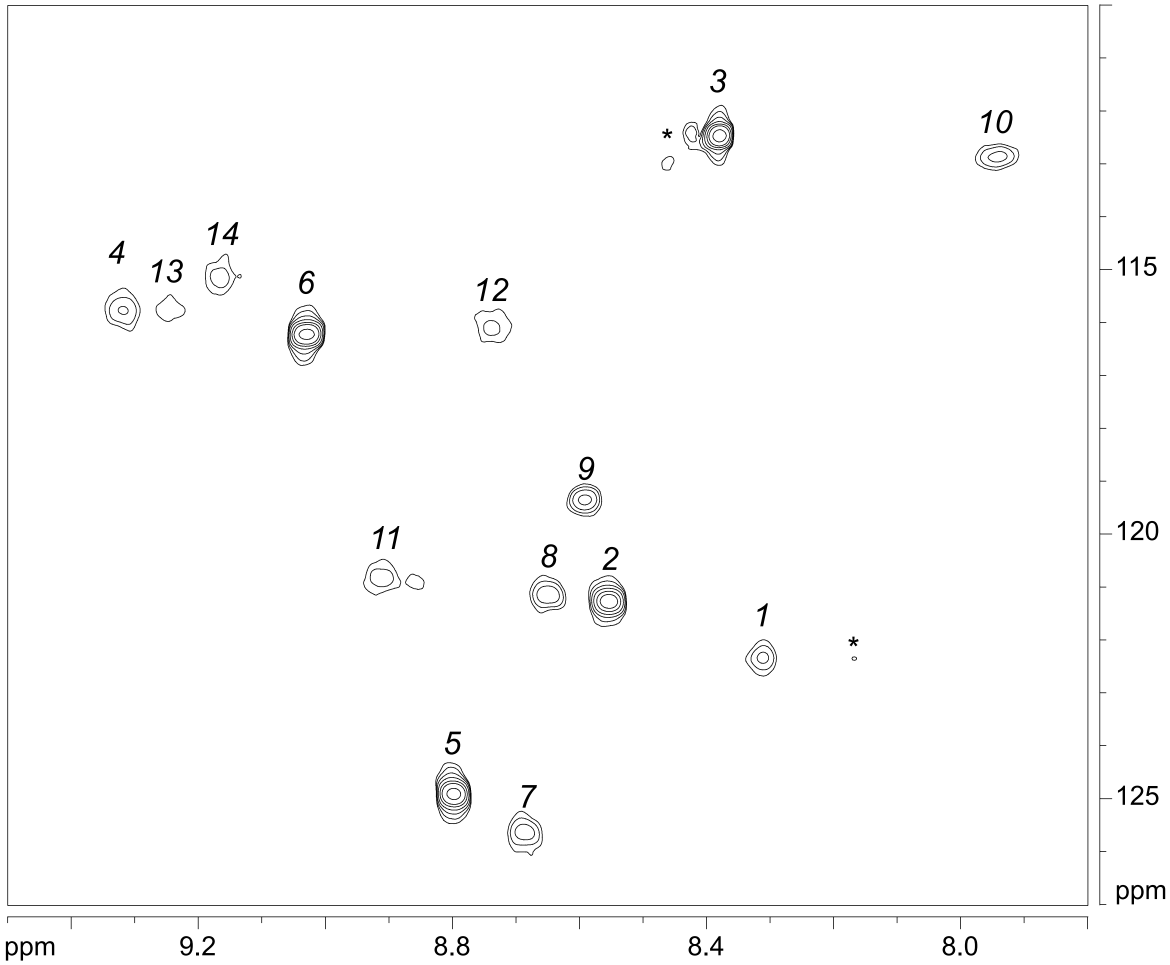
[a] χ_B designates the mole fraction of peptide [^{15}N]**1b**.

$^1H,^{15}N$ HSQC of peptides [^{15}N]**1a** and [^{15}N]**1b** at 600 MHz and 293 K
$\chi_B{}^a$ = 0.375; 8.0 mM total concentration in 9:1 H_2O/D_2O



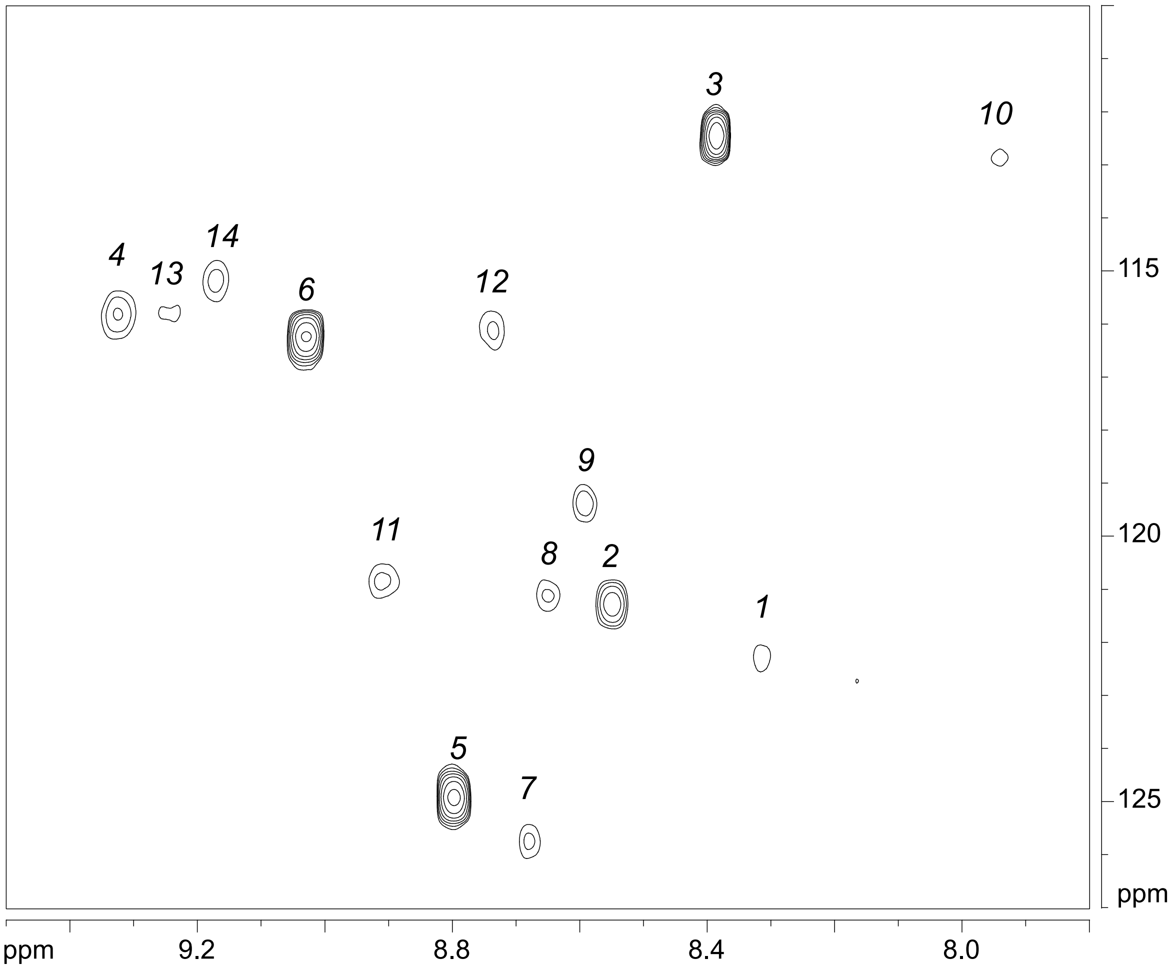
[a] χ_B designates the mole fraction of peptide [^{15}N]**1b**.
The asterisks (*) indicate crosspeaks associated with minor unidentified species.

$^{1}H,^{15}N$ HSQC of peptides [^{15}N]**1a** and [^{15}N]**1b** at 600 MHz and 293 K
$\chi_B{}^{a}$ = 0.50; 8.0 mM total concentration in 9:1 H_2O/D_2O



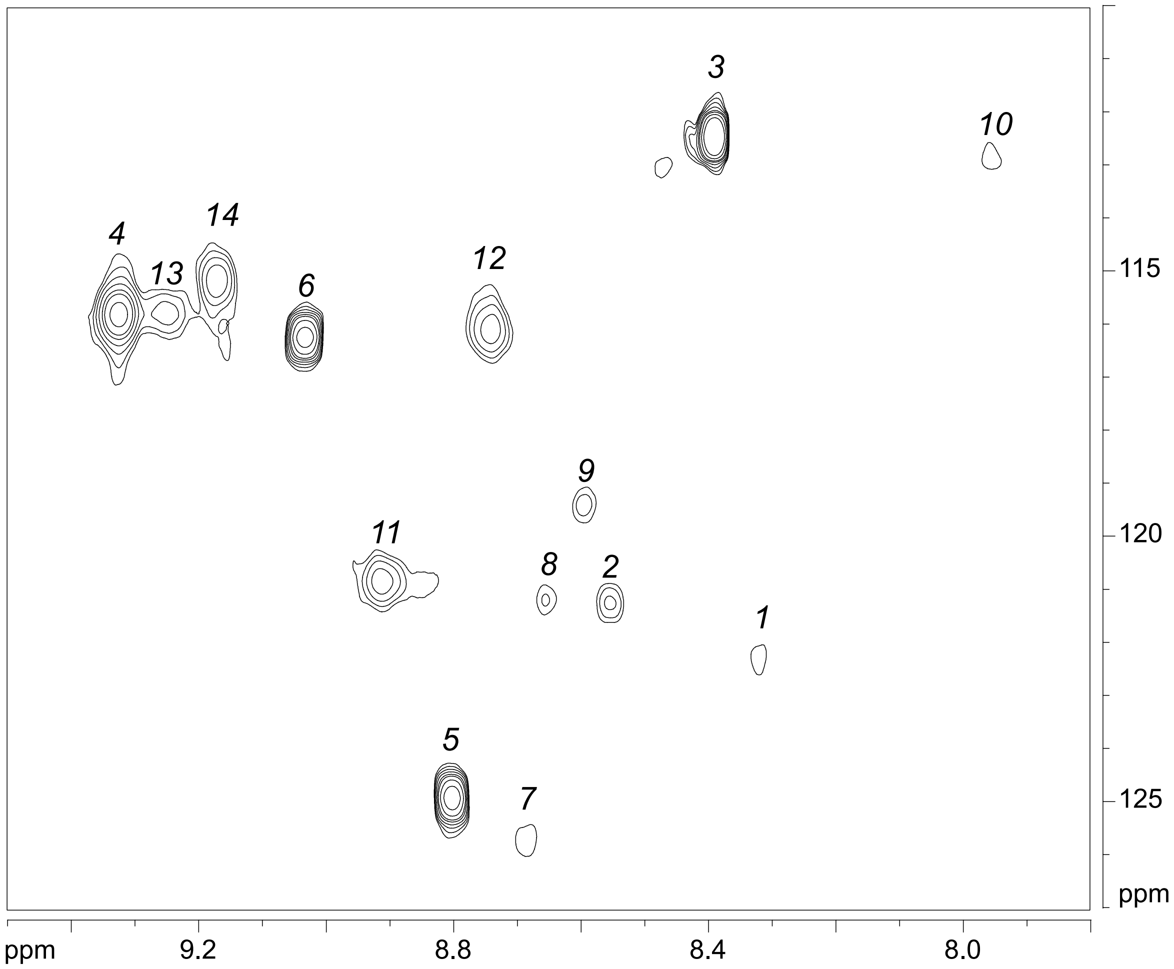
[a] χ_B designates the mole fraction of peptide [^{15}N]**1b**.
The asterisks (*) indicate crosspeaks associated with minor unidentified species.

^{1}H,^{15}N HSQC of peptides [^{15}N]**1a** and [^{15}N]**1b** at 600 MHz and 293 K
$\chi_B{}^a$ = 0.625; 8.0 mM total concentration in 9:1 H_2O/D_2O



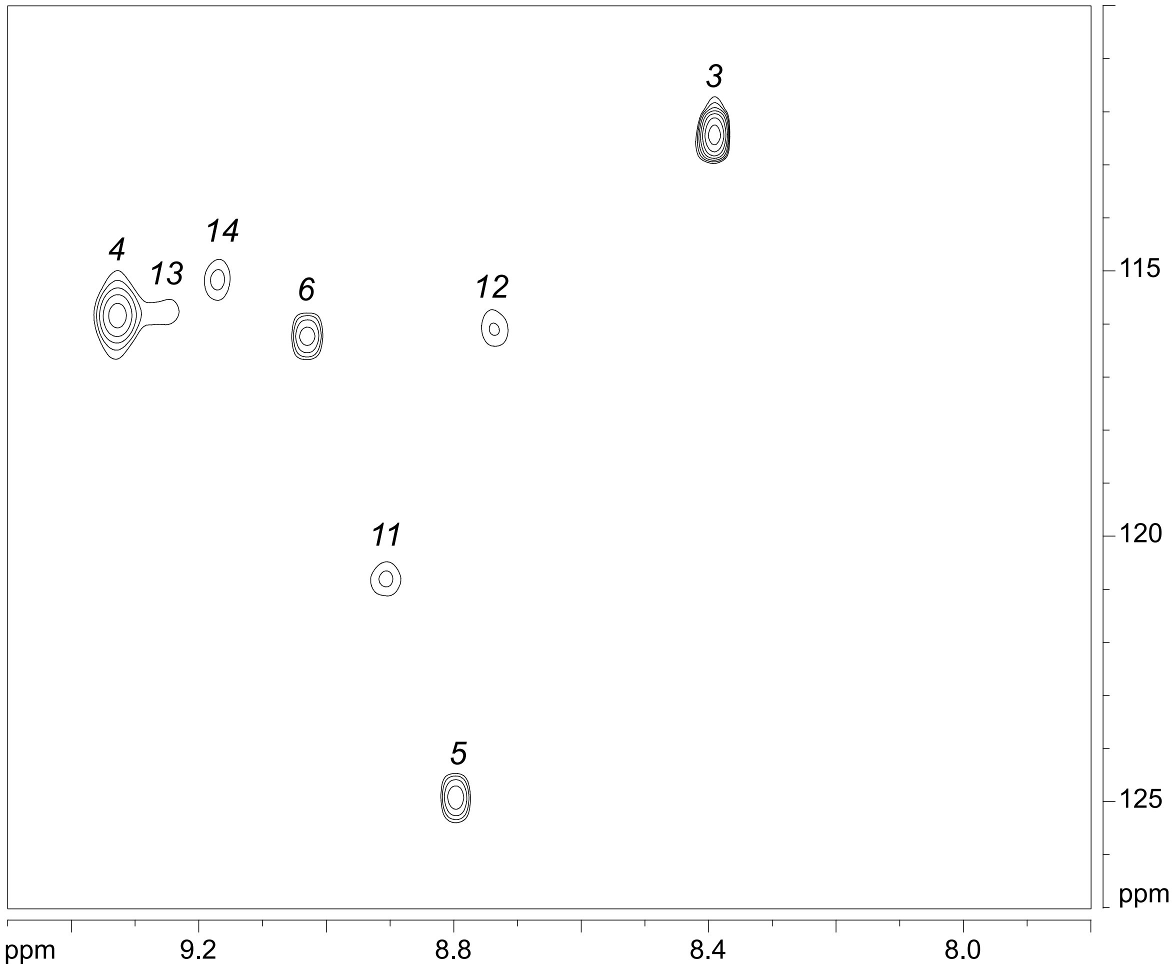
[a] χ_B designates the mole fraction of peptide [^{15}N]**1b**.

^{1}H,^{15}N HSQC of peptides [^{15}N]**1a** and [^{15}N]**1b** at 600 MHz and 293 K
$\chi_B{}^a$ = 0.75; 8.0 mM total concentration in 9:1 H_2O/D_2O



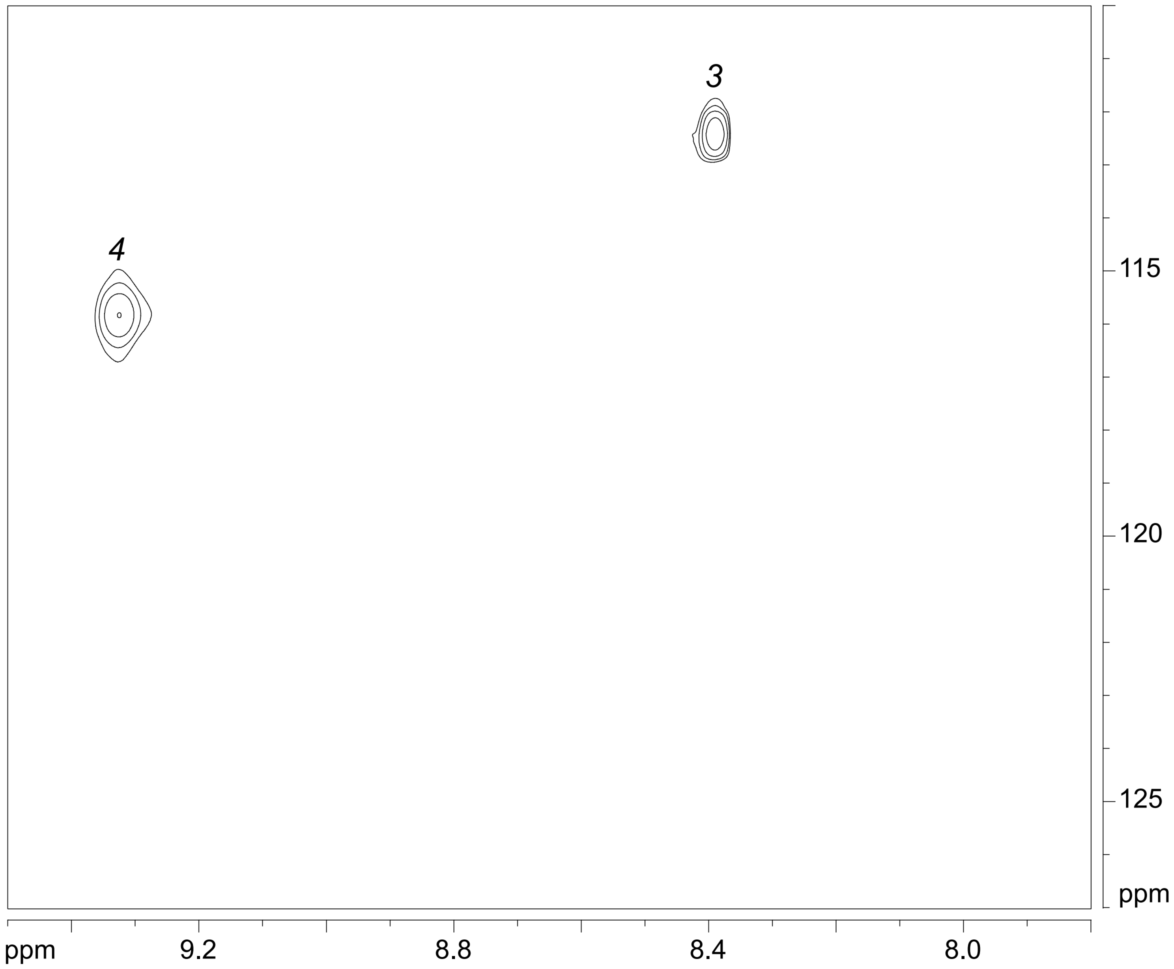
[a] χ_B designates the mole fraction of peptide [^{15}N]**1b**.

$^1H,^{15}N$ HSQC of peptides [^{15}N]**1a** and [^{15}N]**1b** at 600 MHz and 293 K
χ_B[a] = 0.875; 8.0 mM total concentration in 9:1 H_2O/D_2O



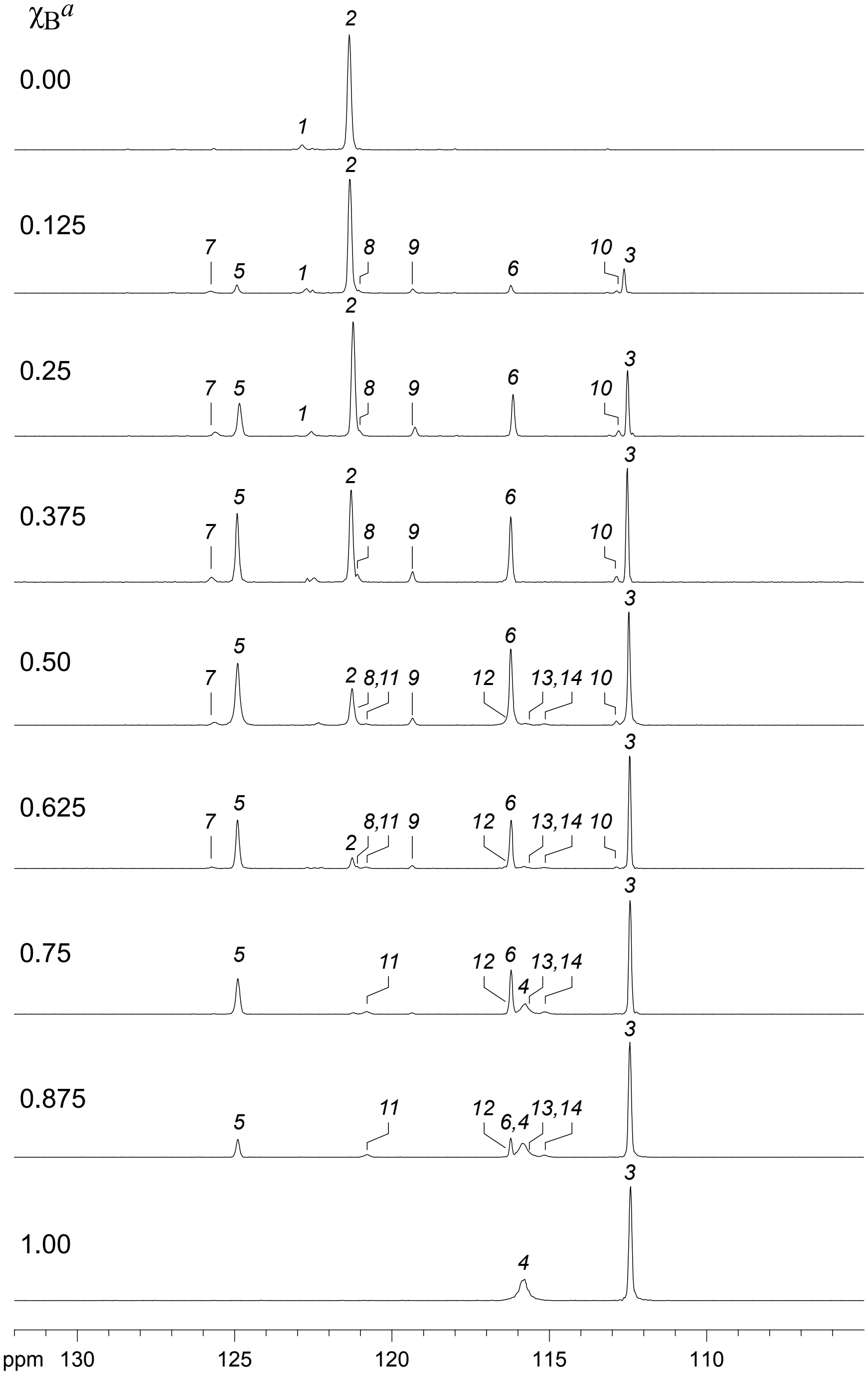
[a] χ_B designates the mole fraction of peptide [^{15}N]**1b**.

$^1H,^{15}N$ HSQC of peptides [^{15}N]**1a** and [^{15}N]**1b** at 600 MHz and 293 K
$\chi_B{}^a$ = 1.00; 8.0 mM total concentration in 9:1 H_2O/D_2O



[a] χ_B designates the mole fraction of peptide [^{15}N]**1b**.

^{1}H,^{15}N HSQC of peptides [^{15}N]**1a** and [^{15}N]**1b** in 9:1 H_2O/D_2O at 600 MHz and 293 K
Stack of ^{15}N spectra from the *f*1 projections of the ^{1}H,^{15}N HSQC spectra



[a] χB designates the mole fraction of peptide [^{15}N]**1b**.