Electronic Supplementary Material (ESI) for Soft Matter.

Electronic Supporting Information

Effects of Solvent Conditions on the Self-Assembly of Heterotrimeric Collagen-Like Peptide (CLP) Triple Helices: A Coarse-Grained Simulation Study

Phillip A. Taylor,[1,#] Stephen Kronenberger,[1,#] April M. Kloxin,[1,2] Arthi Jayaraman*[1,2]

1. Department of Chemical and Biomolecular Engineering, University of Delaware
2. Department of Materials Science and Engineering, University of Delaware

# Equal contributions

*Corresponding author arthij@udel.edu

Section S.1. Details of the CLP model and Simulations

The simulations presented in the main manuscript are run in an NVT ensemble with a constant reduced temperature T* = 3.0, with the following CG model parameters.

Table S1. Bond potential parameters for the CG CLP Model

Type	**Force constant, k (ε/σ^2)**	**Equilibrium bond length, $r_0(\sigma)$**
BB-BB	1000	0.5
BB-HB	1000	0.37

Table S2. Angle potential parameters for the CG CLP Model

Type	**Force constant, k (ε/rad^2)**	**Equilibrium angle, $\theta_0(rad)$**
BB-BB-BB	20	π
HB-BB-BB	300	π/2

Table S3. Dihedral angle potential parameters for the CG CLP Model

Type	**Force constant, k (ε)**	**Equilibrium dihedral angle, $\varphi_0(rad)$**
GH-GB-PB-PH	15	-2π/3
PH-PB-GB-GH	15	2π/3

Table S4. Nonbonded potential parameters for the CG CLP Model under good solvent conditions

Type	**Potential Type**	ϵ_{ij} **(ε)**	$\sigma_{ij}(\sigma)$
BB-BB	WCA	1.0	1.00
BB-HB	WCA	1.0	0.65
HB-HB (D-A)	LJ	50.4	0.30
HB-HB(D-D) or (A-A)	WCA	1.0	0.70

Table S5. Nonbonded Potential Parameters for the CG CLP Model under poor solvent conditions

Type	**Potential Type**	ϵ_{ij} **(ε)**	$\sigma_{ij}(\sigma)$
BB-BB	LJ	ε_{BB}	1.00
BB-HB	WCA	1.0	0.65
HB-HB (D-A)	LJ	50.4	0.30
HB-HB(D-D) or (A-A)	WCA	1.0	0.70

Section S.2: Details for Analysis Implementation

a. Fractal Dimension

The box-counting method of approximating fractal dimension computes the number of cubes with side length ε that are needed to cover the entirety of the voxelated shape. The fractal dimension is then computed as:

$$D_f = -\lim_{\varepsilon \to 0} \frac{\log\left(N(\varepsilon)\right)}{\log\left(\varepsilon\right)} \qquad \text{(S1)}$$

where D_f is the fractal dimension, ε is the size of the side length of the spanning cubes (in number of voxels), and $N(\varepsilon)$ is the number of cubes of side length ε needed to cover the voxelated imaged. We show instances of this calculation for one trial each for several simulation conditions in **Figure S1**. This box counting was performed using the box-counting method in PoresPy, with 15 bins used for the calculation.[1]

As the cube size increases to lengths near the simulation box size, frequently $N(\varepsilon)$ remains constant over a range of ε (that is, the number of cubes required to cover our voxelated image of the network remains the same for different cube sizes over that range). To avoid biasing our calculated fractal dimension as a result of these plateaus, we use only the first half of the data to calculate the fractal dimension (the smallest cube sizes), and note that the linear fit over this half of the data generally describes the trend of the data at larger cube sizes moderately well.

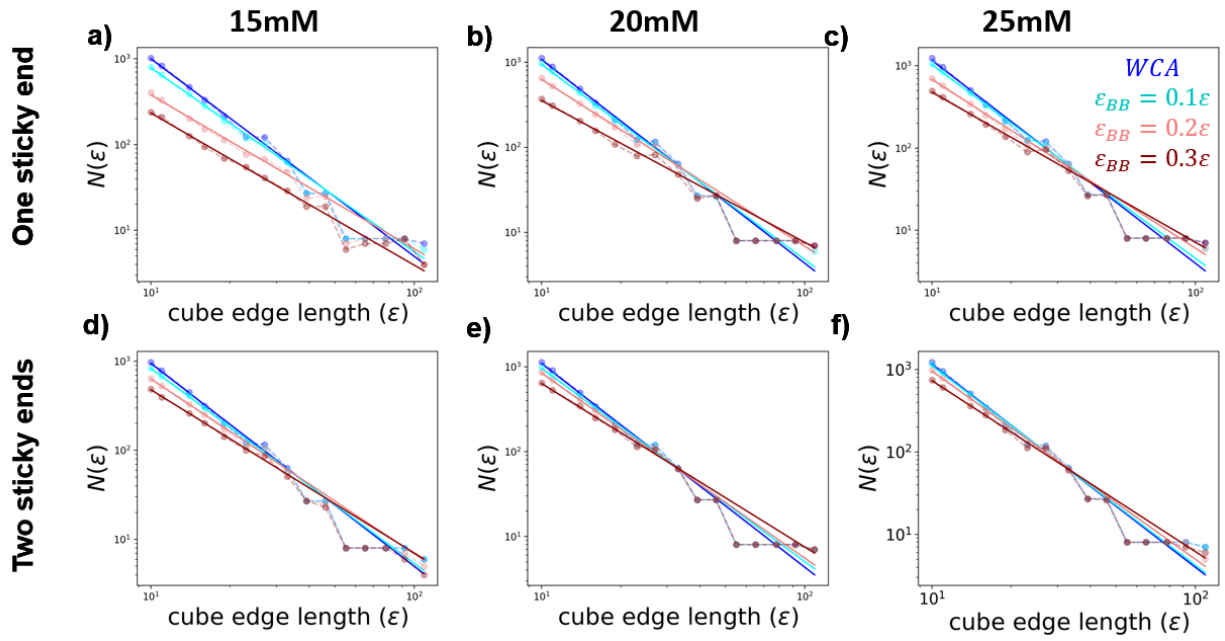

Figure S1: Results of the box counting method used to calculate the fractal dimension for one trial of each simulation condition. Filled circles with dotted lines show the box counting data, as described above, while solid lines show the linear fit to the data. The top row plots correspond to results from one sticky-ended CLP helices while the bottom row plots correspond to two sticky-ended CLP helices; each column corresponds to the value of CLP concentration as indicated. In each plot, the results for each solvent quality are depicted with a dark red curve ($\varepsilon_{BB} = 0.3\ \varepsilon$; worst solvent), light red curve ($\varepsilon_{BB} = 0.2\ \varepsilon$), cyan curve ($\varepsilon_{BB} = 0.1\ \varepsilon$), dark blue (WCA interaction; good solvent).

b. Lacunarity

This method computes the lacunarity for cubes containing different numbers of voxels. The voxel representation of the simulation is partitioned evenly into cubes of the same size, and the number of voxels valued 1 in each cube is counted. These counts are then represented as a probability distribution over all cubes. The mean and variance of this distribution are calculated, and the lacunarity at that cube size is computed using:

$$\lambda(\varepsilon) = \frac{\sigma_\varepsilon^2}{\mu_\varepsilon^2} \qquad \text{(S2)}$$

where λ is the lacunarity, ε is the number of voxels along one edge of the cube, and μ_ε and σ_ε are the mean and standard deviation of the distribution of counts across all cubes.

To create a voxel representation that can be partitioned into cubes of many different sizes, we truncate the voxel representation to be of size 100x100x100 (from 110x110x110), discarding extra voxels. We then compute the lacunarity for all cube sizes that are even divisors of 100 (1, 2, 4, 5, 10, 20, 25, 50, 100). We choose to truncate the data rather than change the voxel size to ensure our analysis is focused on the overall structure, and we do not get additional artifacts from the spherical shape of the simulation beads (if we choose a smaller voxel size) or a poor approximation of the actual structure (if we choose a larger voxel size)

We note that the lacunarity values at $log(\varepsilon) = 0$ (a single voxel) vary as we change the solvent quality at a given concentration, while in theory, the values should depend only on the number of occupied voxels that have a value of 1, which would imply dependence only on the concentration and not on how the CLP helices interact. This fact would be true if the voxels are infinitely small. However, using finite sized voxels, the approximation of structure with the voxel representation leads to discrepancies. Since we value a voxel at 1 it contains any part of a bead, we are consistently overestimating the volume fraction when doing this calculation. This overestimation will generally be higher for networks with smaller diameters and higher branching, since finite-sized voxels will resolve these finer features worse than they would coarser features. We thus linearly scale all lacunarity curves for a given concentration to the range between 1 and the maximum $\lambda(1)$ (across that concentration) for a more appropriate comparison. We note that the choice of the maximum $\lambda(1)$ is somewhat arbitrary, and the most important consideration is that the curves are scaled to the same value at $\varepsilon = 1$ to ensure the lacunarity curves represent the

physical reality of the system rather than artifacts due to voxelization. We show the unscaled version of the lacunarity curves in **Figure 8** of the main text in **Figure S2.**

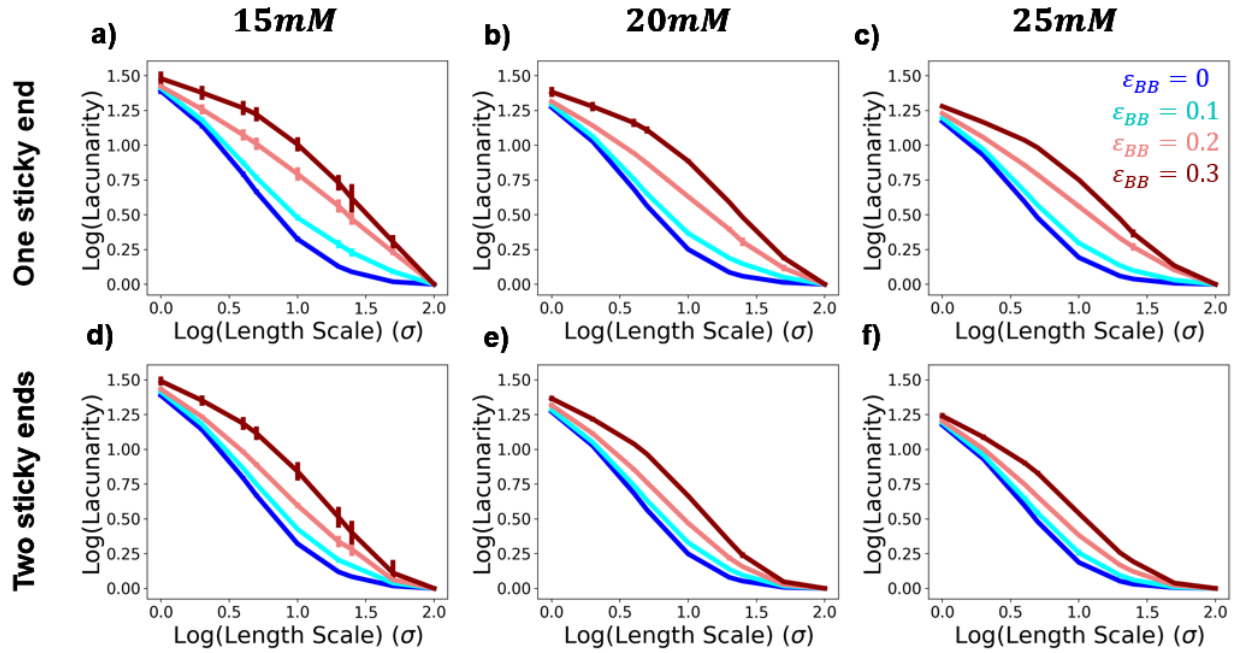

Figure S2: Unnormalized lacunarity curves for the same data shown in Figure 8 of the main body.

c. Graph-based Analysis of Strand Lengths and Diameters

Implementation Details

Here, we include the specific details for the implementation our graph-based analysis.

The NetworkX Python package allows the user to assign attributes to nodes and edges, which we use to track the positions of nodes.[2] We use this feature to keep track of the voxels and simulation beads associated with each node and edge in the voxel graph as we modify the graph. The steps we take to generate the graph for analysis, and attributes we store in each node and edge at each step are as follows:

(i) Initial Voxel Graph

Node and Edge Meanings:

Each node represents a voxel in the skeleton, and each edge represents connects adjacent voxels.

Node Attributes:

- x,y,z position of the node
- A list of the nearest original voxels (before the skeletonization)

Edge Attributes:

- A weight of 1, to be used for simplifying the graph over the next two steps

(ii) Simplified Voxel Graph

In the previous step, we have a node at every voxel along a network strand. However, we would like our graph representation to have node representing network junctions and edges representing network strands. To accomplish this, we iteratively remove all nodes with degree 2 (nodes with 2 attached edges are along a strand in the network), connecting their neighbors with an edge. For each node removal, we sum the edge attribute weight of the two connected edges to keep track a rough estimate of how far apart the nodes are. We also add another attribute to each edge to keep track of the original voxels attributed to each edge, and add the list of original voxels of each deleted node to its associated edge.

Node and Edge Meanings:

Each node represents a junction in the skeleton (voxels from the initial voxel graph with 1 or 3+ neighbors), and each edge represents the strand along the skeleton connecting the nodes in the initial voxel graph.

Node Attributes:

- x,y,z position of the node
- A list of the nearest original voxels (before the skeletonization)

Edge Attributes:

- A weight greater than or equal to 1, to be used for simplifying the during the next step
- A list of the nearest original voxels (before the skeletonization) of all nodes deleted along this edge.

(iii) Simplified Voxel Graph with Removal of "Junk" Nodes

After simplifying the graph in **step (ii),** the skeletonization process can result in many junctions clustered together around the true junction. We show some examples of these "junk" nodes in **Figure S3.** To remove these "junk" nodes, we identify clusters of adjacent nodes from **step (ii)** that have edge weights of 1, corresponding to adjacent voxels with 3 or more neighbors. For each cluster, we remove all nodes in the cluster, and add a new node located at the average position of all the nodes that were removed in that cluster. The original voxel list of each of the removed nodes is added to the new node. After resolving all the "junk" nodes, we once again simplify the graph as described in **step (ii)**

Node and Edge Meanings:

Same as step (ii)

Node Attributes:

- x,y,z position of the node
- A list of the nearest original voxels (before the skeletonization)

Edge Attributes:

- A weight greater than or equal to 1, which will not be used any further
- A list of the nearest original voxels (before the skeletonization) of all nodes deleted along this edge

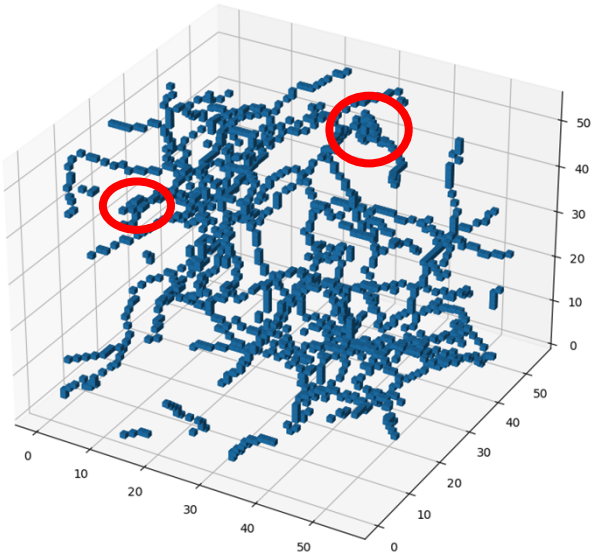

Figure S3: Example of the skeletonized voxel representation of the simulation, showing some instances of "junk" nodes (circled in red), where the skeletonization process results in clumps of junctions surrounding the true junction.

(iv) Removal of Extraneous CLP Helices and periodic boundary artifacts

While it may be possible to include periodic boundaries into the graph representation of the simulation, it is not included in this implementation. This can result in small, isolated artifacts near the edge of the box that we would like to avoid performing analysis on. To accomplish this, we first remove any node with a position within two voxels of the edge of the voxelated image. We then finalize the graph for analysis by taking the component of the resulting graph with the most nodes.

Node and Edge Meanings:

Same as step (ii) or (iii)

Node Attributes:

Same as step (iii)

Edge Attributes:

Same as step (iii)

(v) Bead Graph

We construct another graph based on the simulation beads themselves, where each simulation bead is represented as a node, and beads within a cutoff distance of 2σ are connected by edges. We use only OB beads for the construction of both graphs to reduce the computational intensity of the analysis. Each node represents a simulation bead, and each edge connects beads within the cutoff distance. In each edge, we store the distance between the connected nodes, to be used for the length calculation.

(vi) Length Calculation

We compute the length of each edge in the analysis graph first, since we also use the length calculation to verify that the edge is not an artifact of the voxelization and skeletonization process. To compute the length of each edge in the graph from Step 4, we identify all simulation beads associated with the edge and its two nodes using the list of original voxels stored in their respective attributes. We use Dijkstra's algorithm to compute the shortest distance (weighted by the edge weights of the distances between simulation beads) along the graph from the beads closest in position to each node in the voxel graph. Some instances occur where the edge is an artifact of the voxelization and skeletonization, and no path exists between the two nodes. We show two such

examples in **Figure S4**. In this case, we remove the edge from the graph so that it is not used in future analysis.

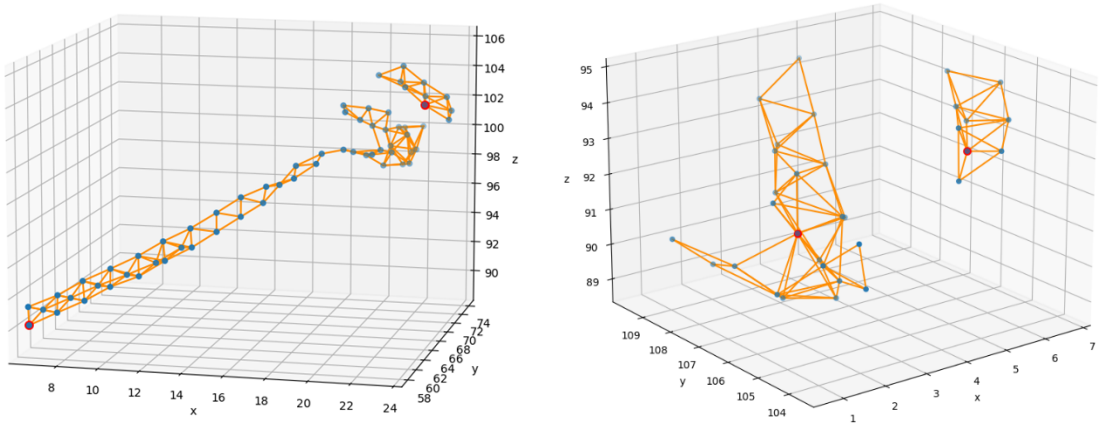

Figure S4: Examples of edges in the graph that have no connecting path. Bead locations nearest to the identified node are highlighted in red. The blue points show all simulation beads associated with the strand, and orange edges connect all beads that are within a distance of 2σ of one another.

(vii) Diameter Calculation

For each edge on the voxel graph, we get the associated voxels from the graph in Step 1. We only consider edges with more than five of these voxels for the diameter calculation, with smaller edges having a higher tendency to give unreliable results. For the diameter calculation, we then take a five-voxel segment of the edge and calculate the direction vector from the positions of the first and last voxels in this segment. We compute the distance from each simulation bead associated with this segment to that vector, and report the diameter of that segment as double the maximum distance plus one (to account for the radii of the simulation beads themselves). This process is repeated for each five-voxel-long segment along the edge, and the diameter for the edges is reported as the mean of all diameter segments.

d. Pore Size Distribution

For our analysis, we sample 1000 points per simulation snapshot analyzed to compute the pore size distribution. We have sampled also tried sampling different numbers of coordinates for each snapshot for a few different systems and found larger variations when we sample fewer coordinates (**Figure S5, right**). However, even with sufficiently fewer coordinates (100) sampled per configuration, the distributions mean distributions are very similar (**Figure S5, left**). Sampling 1000 coordinates shows only minor differences to sampling 1500 coordinates, so we proceeded with using 1000 as the number of sampled coordinates.

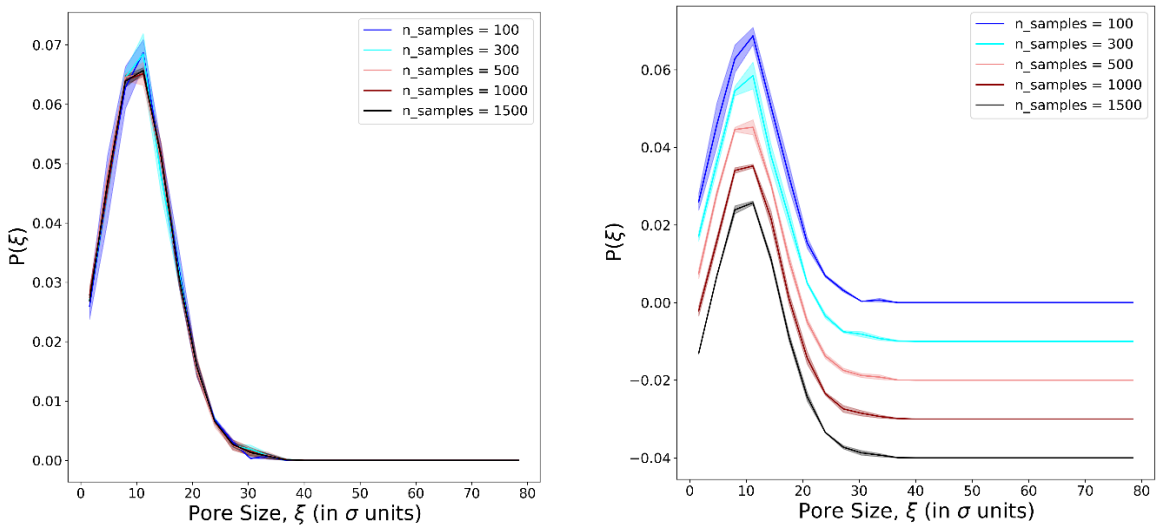

Figure S5: Plots of the pore size distribution for networks of CLP triple helices using a different number of sampled coordinates to compute the pore size distribution (all other details are identical to those described in **Figure 8** of the main paper). The lines indicate the mean of three simulation trials, while the shaded region represents the standard deviation across the three simulation trials. The data shown is for 20mM 1 sticky-ended CLP triple helix networks with ε_{BB}=0 (WCA interactions). Left: data is shown as computed to show a direct comparison of the means, Right: Data for 300, 500, 1000, and 1500 are shifted downward by 0.01, 0.02, 0.03, and 0.04 respectively for visual clarity of the error bars.

Section S.3: Additional Visualization of Simulations

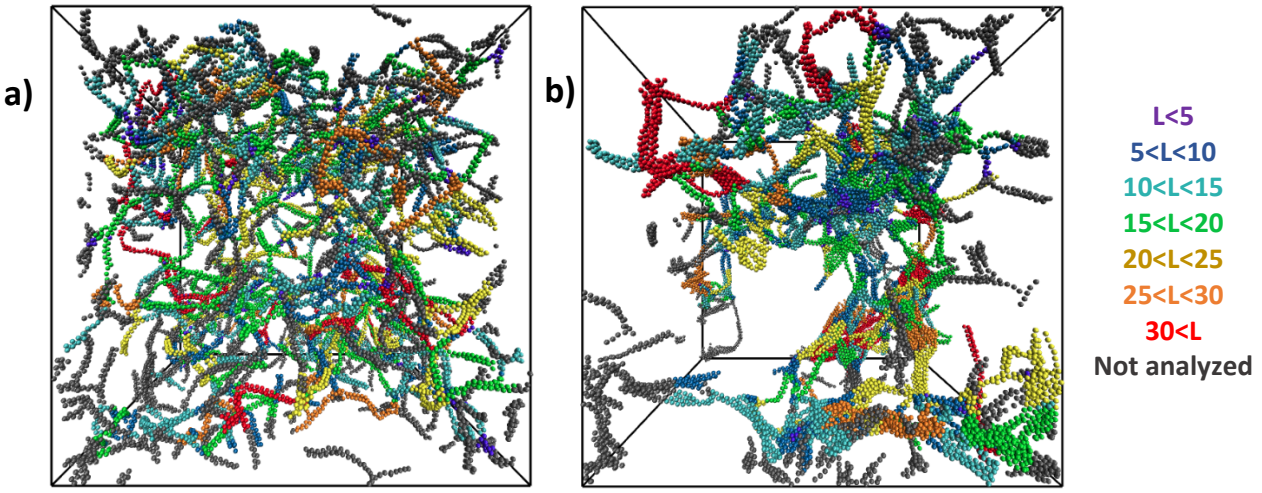

Figure S6: Examples of the strand length calculation. Here, we show the entire simulation box colored by the values of strand length (L) calculated by the graph analysis method. Images are from a snapshot of a network assembly in a 20mM simulation of one sticky-ended CLP helices with $\varepsilon_{BB} = 0$ (a) and $\varepsilon_{BB} = 0.2$ (b) at the end of the 10 million timestep production run. Gray beads are not included in the analysis as a result of removing nodes near the boundary of the box and removing edges during the strand length calculation as described in the implementation details in Section 1.

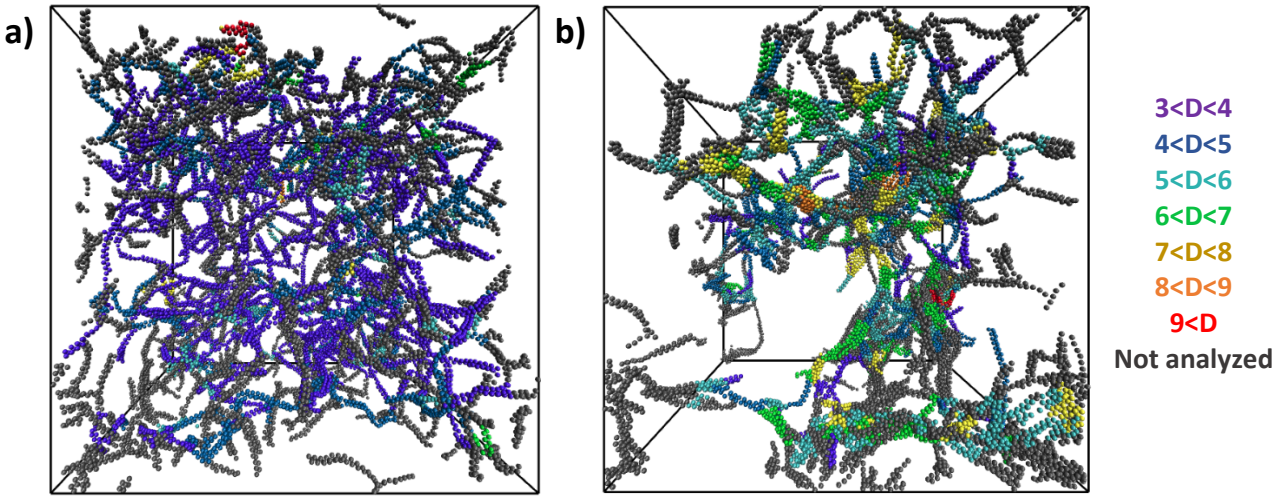

Figure S7: Examples of the strand diameter calculation. Here, we show the entire simulation box colored by the values of strand diameter calculated by the graph analysis method. Images are from a snapshot of a network assembly in a 20mM simulation of one sticky-ended CLP helices with $\varepsilon_{BB} = 0$ (a) and $\varepsilon_{BB} = 0.2$ (b) at the end of the 10 million timestep production run. Gray beads are not included in the analysis as a result of removing nodes near the boundary of the box and removing edges during the strand length calculation as described in the implementation details in Section 1.

Section S.4: Fibril Assembly at Dilute CLP helix concentrations

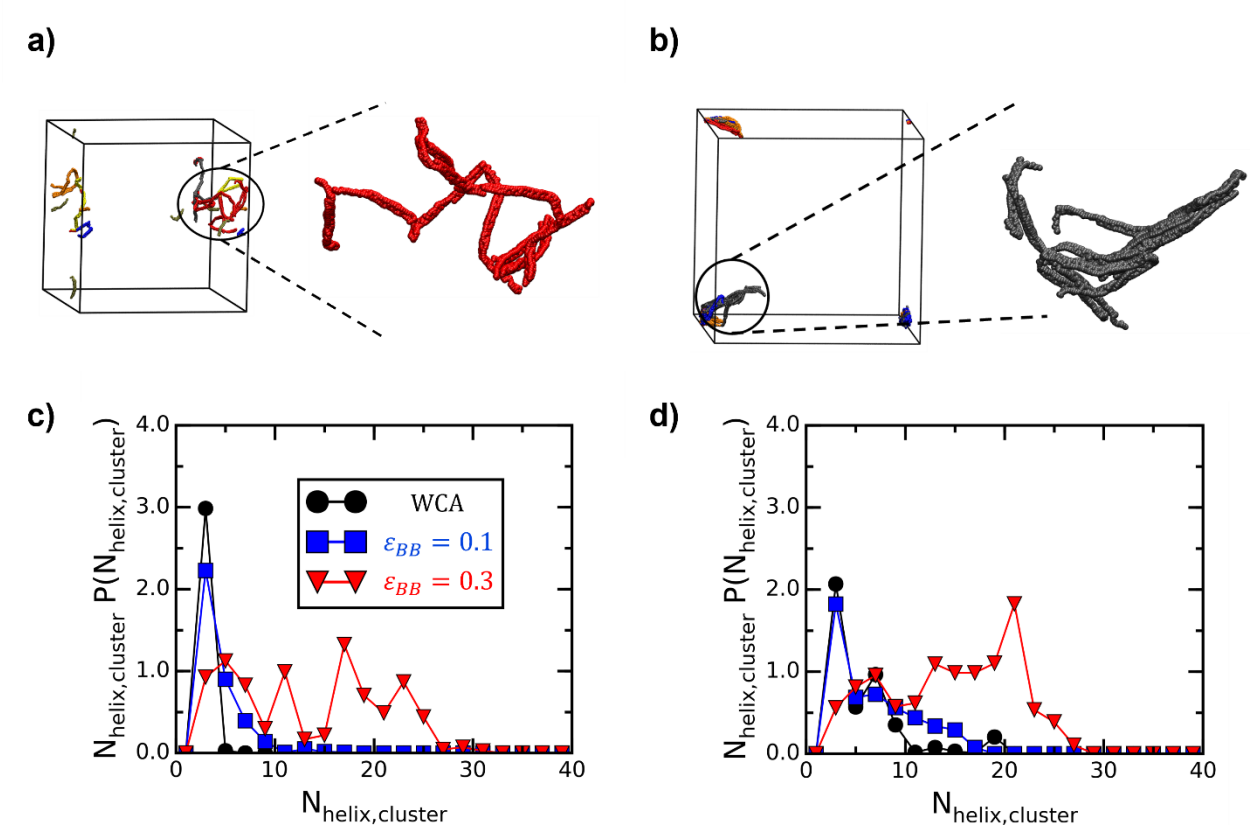

Figure S8. Simulation snapshots of two sticky-ended coarse-grained CLP triple helices and weighted probability distributions of sizes of clusters of one and two sticky-ended heterotrimeric CLP triple helices measured in terms of the number of helices per cluster, $N_{helix,cluster}$ with decreasing solvent quality at a CLP concentration of 1 mM. Snapshots for each solvent quality are depicted with a red curve (ε_{BB} = 0.3 ε; worst solvent), blue curve (ε_{BB} = 0.1 ε), and black curve (WCA interaction; good solvent). Weighted distributions of number of helices per cluster are shown for c) one sticky-ended and d) two sticky-ended CLP triple helices.

Section S.5: Figures Contrasting Results from One vs Two sticky-ended CLPs

In this section, we replot the data from the main body of the text for easier comparison between the one and two sticky-ended CLP data. The curves and error bars have the same meaning as those described in the main text, and we have only changed the colors and order in which the data is plotted.

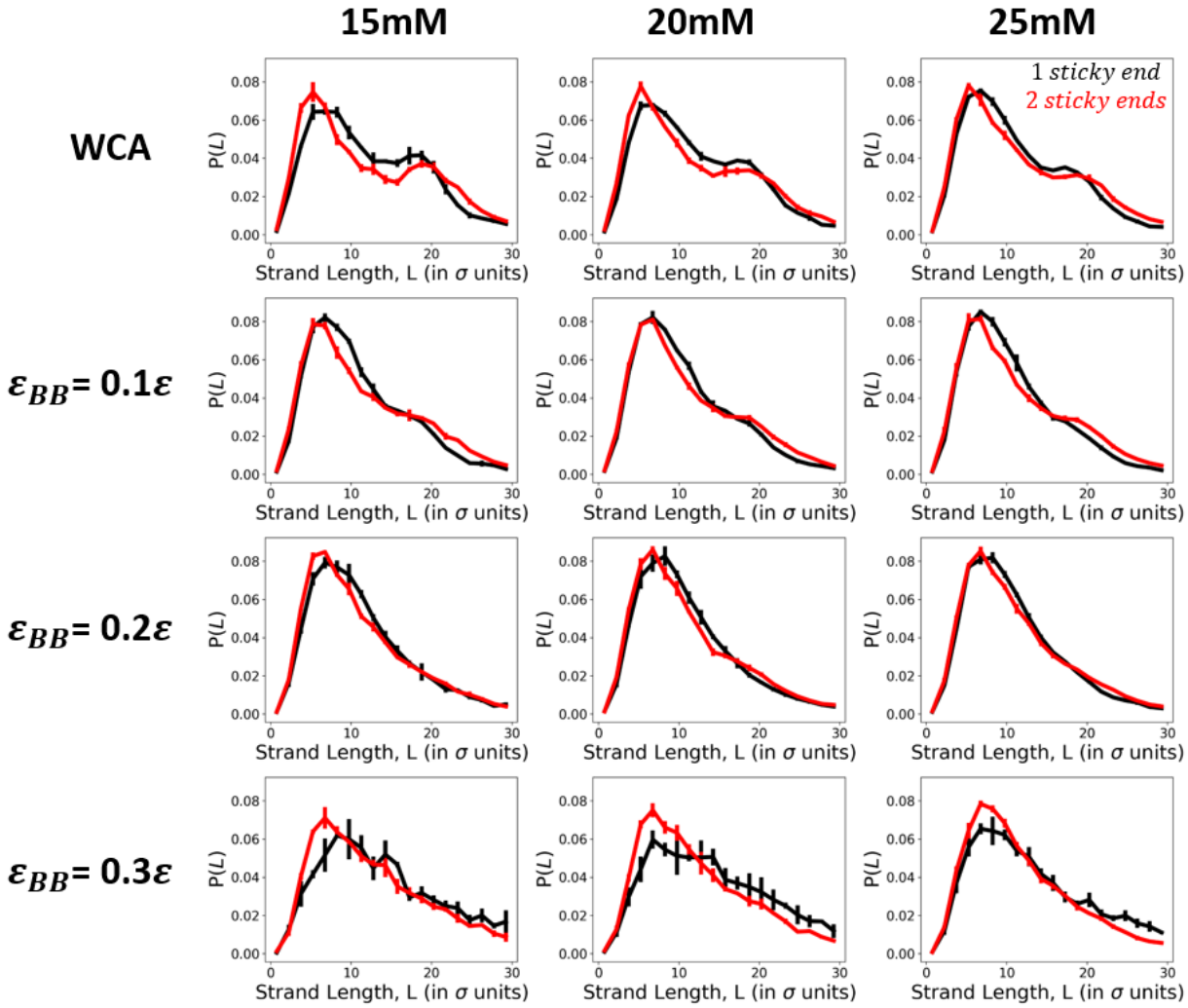

Figure S9: Distribution of strand lengths, L, observed for the networks formed from CLP triple helices at 15, 20, and 25*mM*. Each row of plots corresponds to a solvent quality, with the best solvent quality in the top row and worst solvent quality in the bottom row; each column corresponds to the value of CLP concentration as indicated. In each plot, the results for one and two sticky-ended CLP helices are depicted with a black and red curve, respectively. For each simulation, we compute the mean length distribution over ten snapshots collected every one million timesteps for the final 10 million timesteps of the simulation. Error bars represent the

standard deviation of these means across three independent trials as described in the methods section.

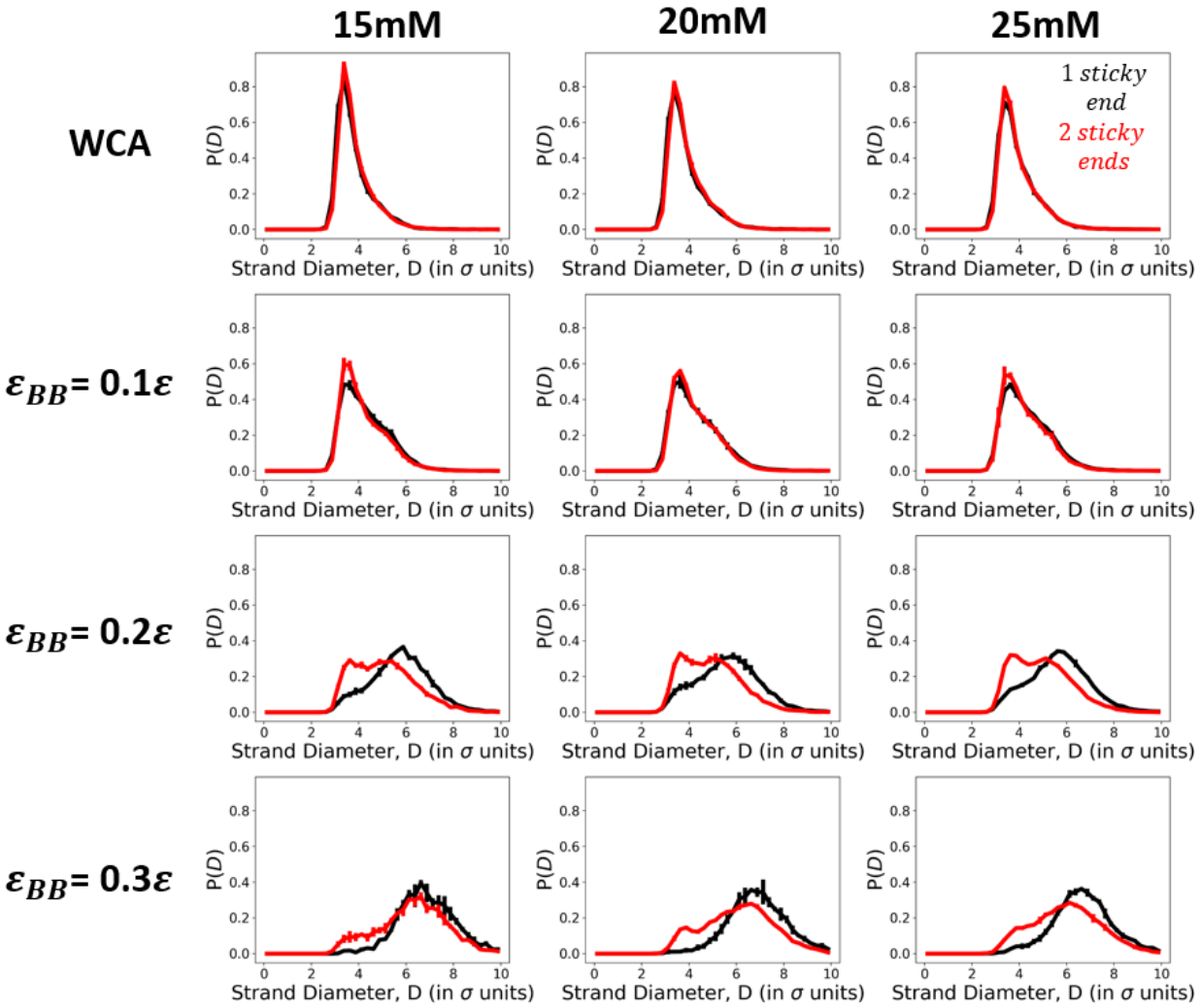

Figure S10: Distribution of strand diameters, D, observed for the networks formed from CLP triple helices at 15, 20, and 25*mM*. Each row of plots corresponds to a solvent quality, with the best solvent quality in the top row and worst solvent quality in the bottom row; each column corresponds to the value of CLP concentration as indicated. In each plot, the results for one and two sticky-ended CLP helices are depicted with a black and red curve, respectively. For each simulation, we compute the mean diameter distribution over ten snapshots collected every one million timesteps for the final 10 million timesteps of the simulation. Error bars represent the standard deviation of these means across three independent trials as described in the methods section.

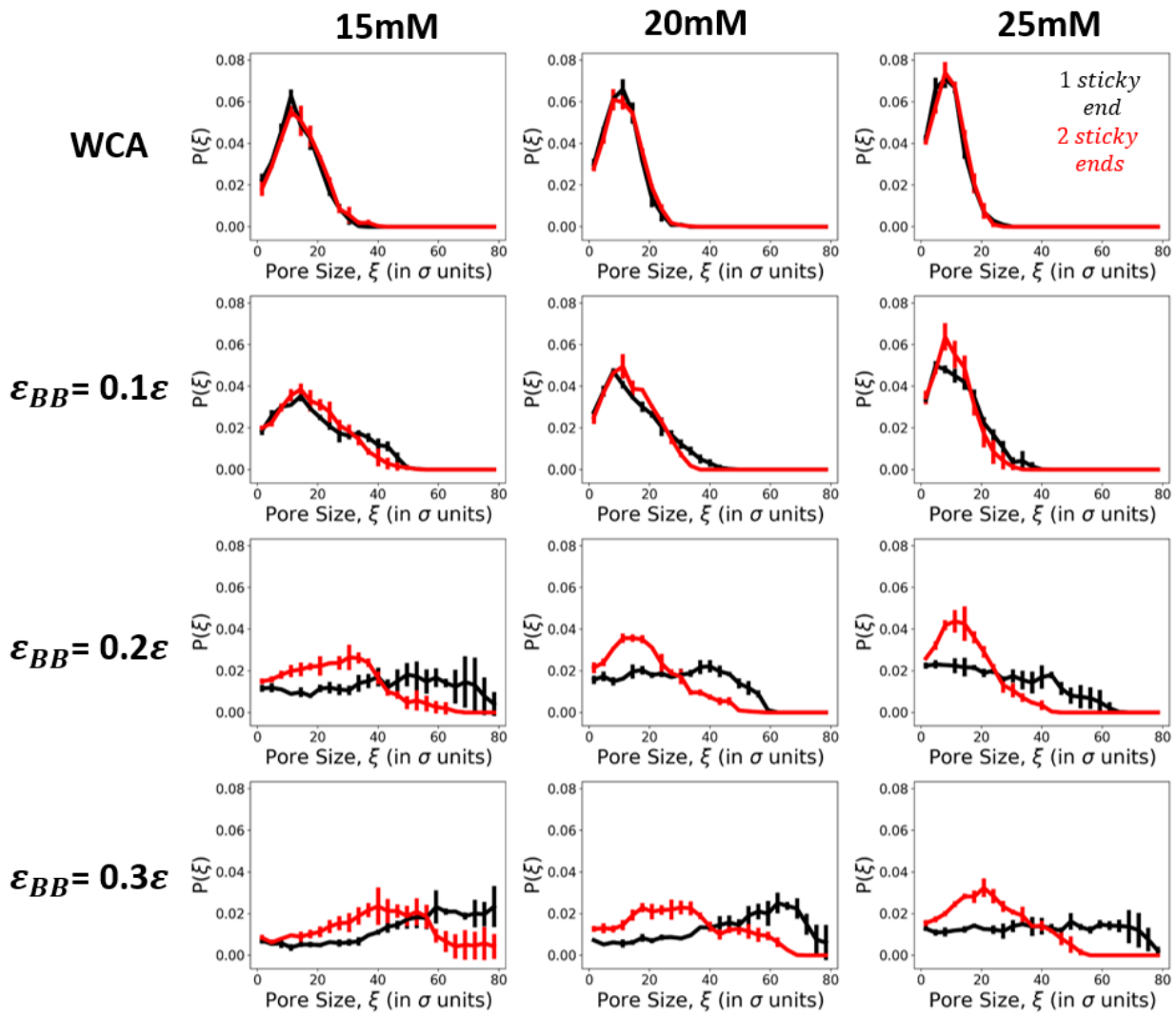

Figure S11: Distribution of pore sizes, ξ, observed for the networks formed from CLP triple helices at 15, 20, and 25*mM*. Each row of plots corresponds to a solvent quality, with the best solvent quality in the top row and worst solvent quality in the bottom row; each column corresponds to the value of CLP concentration as indicated. In each plot, the results for one and two sticky-ended CLP helices are depicted with a black and red curve, respectively. For each simulation, we compute the mean pore size distribution over ten snapshots collected every one million timesteps for the final 10 million timesteps of the simulation. Error bars represent the standard deviation of these means across three independent trials as described in the methods section.

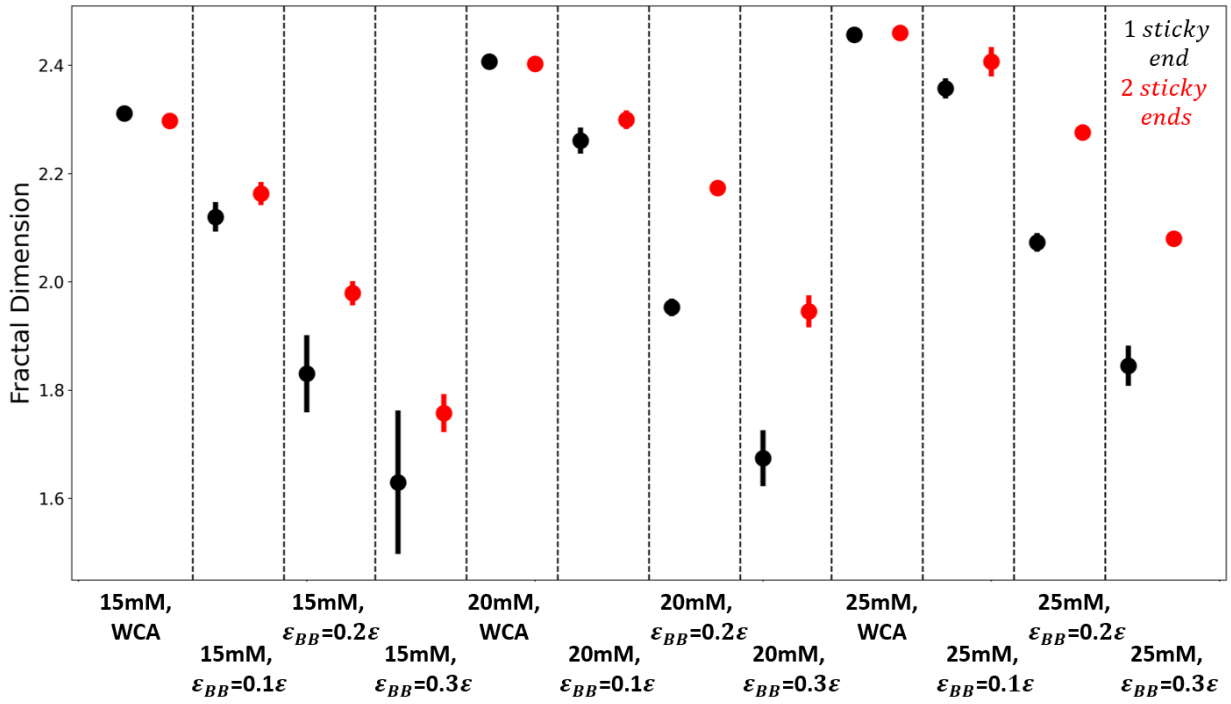

Figure S12: The **fractal dimension** for each CLP concentration and solvent quality. For each simulation, we compute the mean fractal dimension over ten snapshots collected every one million timesteps for the final 10 million timesteps of the simulation. Error bars represent the standard deviation of these means across three independent trials as described in the methods section. Black data points correspond to networks assembled from one sticky-ended CLP helices, and red data points correspond to networks assembled from two sticky-ended CLP helices

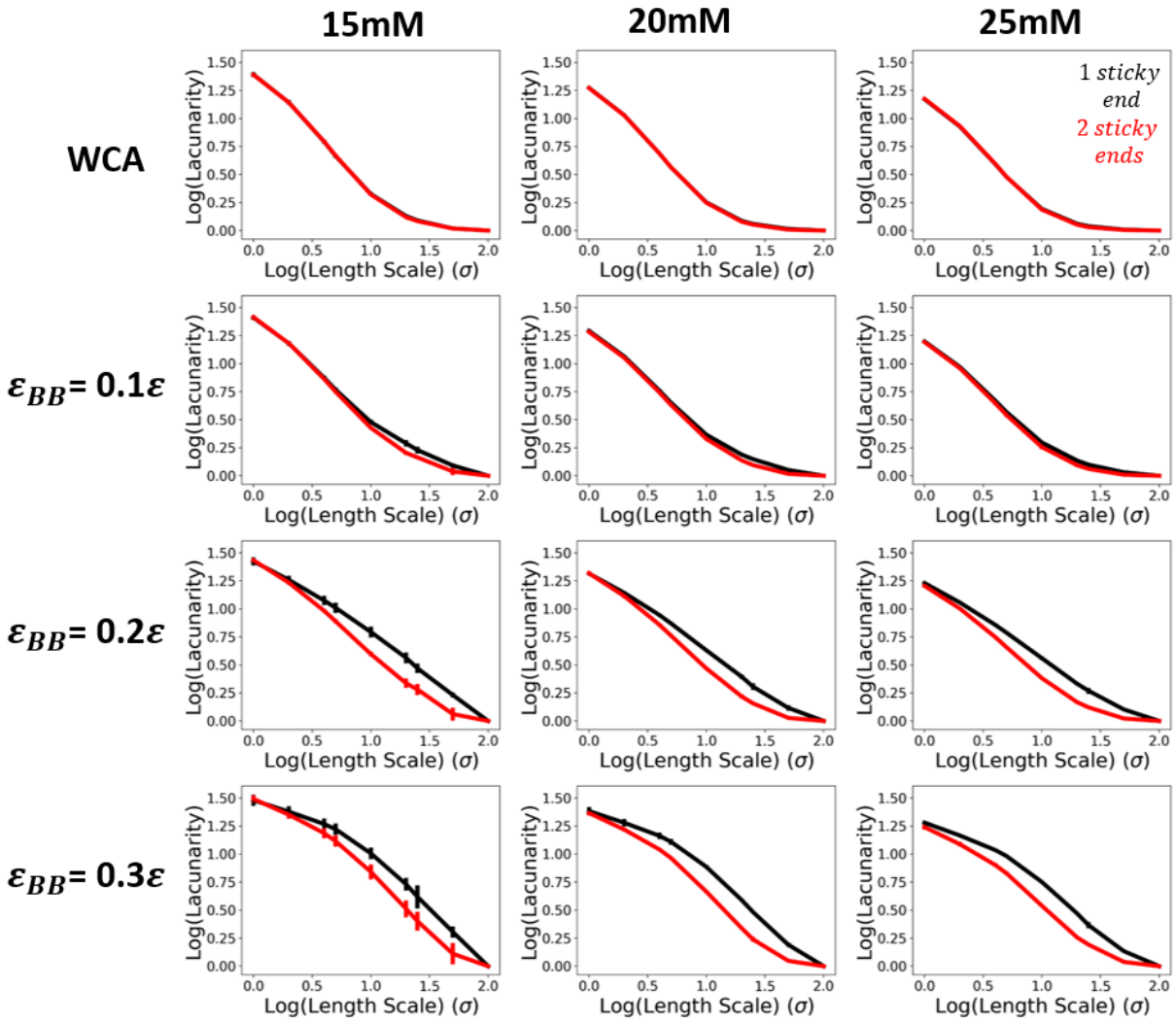

Figure S13: Lacunarity curves (normalized as described in SI Section S.1) observed for the networks formed from CLP triple helices at 15, 20, and 25mM. Each row of plots corresponds to a solvent quality, with the best solvent quality in the top row and worst solvent quality in the bottom row; each column corresponds to the value of CLP concentration as indicated. In each plot, the results for one and two sticky-ended CLP helices are depicted with a black and red curve, respectively. For each simulation, we compute the lacunarity curve over ten snapshots collected every one million timesteps for the final 10 million timesteps of the simulation. Error bars represent the standard deviation of these means across three independent trials as described in the methods section.

Maximum Diameter

Competing enthalpic and entropic contributions for self-assembly can lead to size-limiting mechanisms in the form of a maximum fibril diameter that occurs as the loss in entropy of adding additional chains to a

fibril balances the gain in enthalpy that occurs. This phenomenon has been shown in the study of Douglas and coworkers for semiflexible chains.[3] To test if that happens in our system of CLP triple helices, in **Figure S14**, we plot the mean diameter of the network strands in our simulations as a function of solvent quality. We do not see the mean diameter plateauing within the range of ε_{BB} explored in this study. Therefore, we do not see any evidence of a size-limiting mechanism in CLP network strands irrespective of the backbone interaction strength.

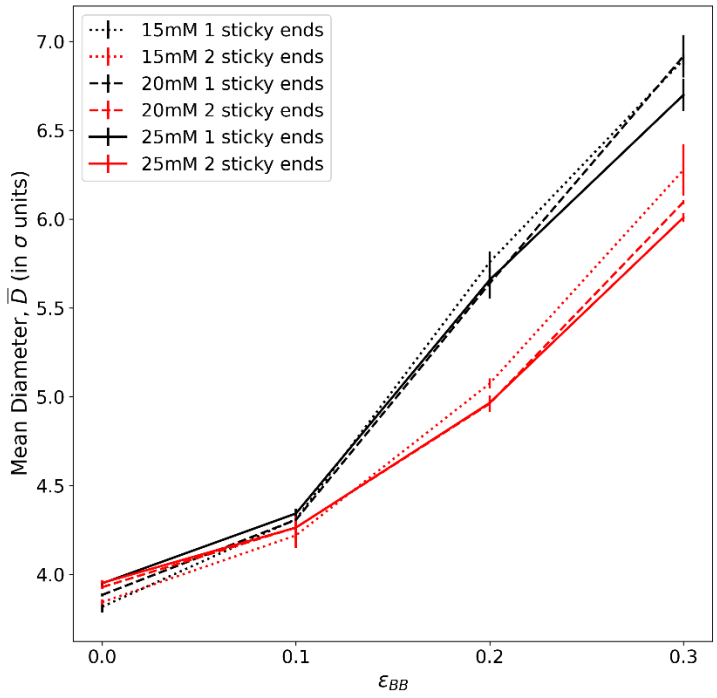

Figure S14: Plots of the mean diameter of CLP network strands as a function of solvent quality for our CLP networks. The lines indicate the mean, and the error bars indicate the standard deviation, of three simulation trials**.** Data is plotted in black for one sticky-ended CLP triple helices, and red for two sticky-ended CLP triple helices. The line style indicates the CLP concentration: dotted is 15mM, dashed is 20mM, and solid is 25mM. All other details are identical to those described in **Figure 7** of the main paper.

References

1. J. T. Gostick, Z. A. Khan, T. G. Tranter, M. D. r. Kok, M. Agnaou, M. Sadeghi and R. Jervis, *Journal of Open Source Software*, 2019, **4**, 1296.

2. A. A. Hagberg, D. A. Schult and P. J. Swart, 2008, 5.
3. F. Vargas-Lara and J. F. Douglas, *Gels*, 2018, **4**, 27.