

Supplemental Information

Design Rules for Encapsulating Proteins into Complex Coacervates

Whitney C. Blocher McTigue and Sarah L. Perry*
Department of Chemical Engineering and the Institute for Applied Life Sciences,
University of Massachusetts Amherst
*Correspondence: perrys@engin.umass.edu

Table S1. Table of counter-ions, molecular weights and polydispersity index (PDI) of polypeptides. PDI was found using gel permeation chromatography (GPC). Polypeptides of chain length 50 were made in house, while N = 100, 400, and 800 were purchased from Alamanda Polymers.

		Counter Ion	M_w (g/mol)	PDI
50	K ₅₀	TFA ^{-a}	12,000	-
	E ₅₀ ^b	Na ⁺	7,500	-
100	K ₁₀₀	Br ⁺	21,000	1.06
	E ₁₀₀ ^c	Na ⁺	15,000	1.02
400	K ₄₀₀	TFA ^{-a}	97,000	1.08
	E ₄₀₀ ^c	Na ⁺	60,000	1.01
800	K ₈₀₀	TFA ^{-a}	194,000	1.06
	E ₈₀₀ ^c	Na ⁺	120,000	1.06

^a TFA is defined as trifluoroacetate.

^b E₅₀ was made in house and is composed of alternating D and L monomers.

^c E_n was purchased from Alamanda Polymers and is racemic, but without sequence control.

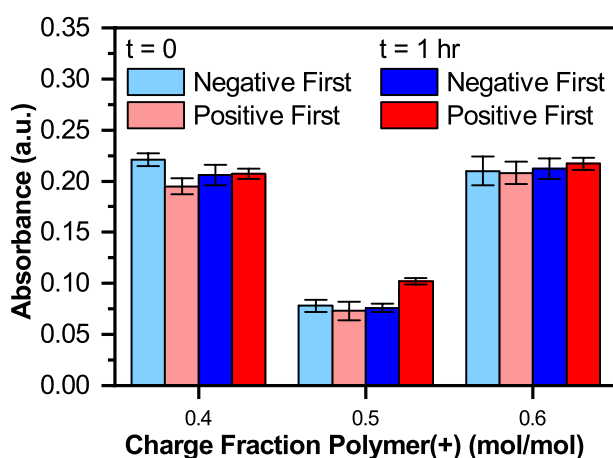


Figure S1. Plot of Coomassie dye absorbance (corresponding to protein concentration) for HEWL-containing samples prepared using different orders of polymer addition (HEWL/E₅₀/K₅₀ vs. HEWL/K₅₀/E₅₀) for different compositions of K₅₀ and E₅₀ in 10 mM HEPES, pH 7.0. Error bars are the standard deviation of the reported average.

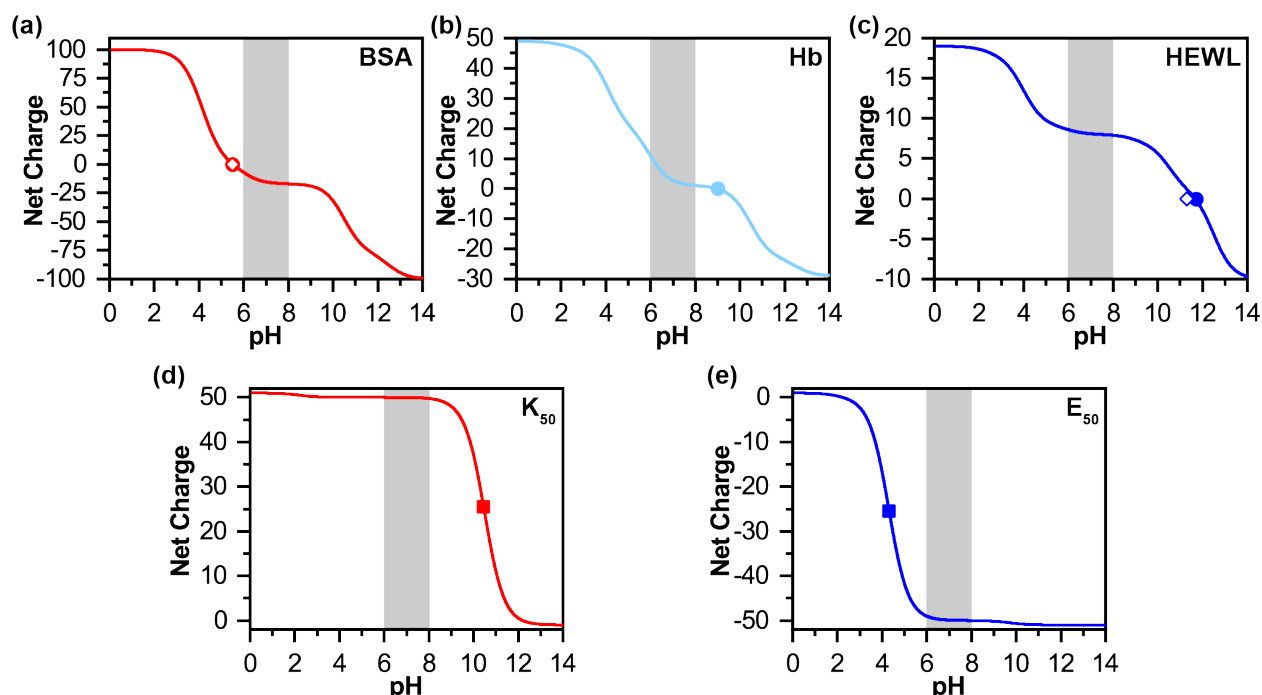


Figure S2. Plots of net protein charge as a function of pH for (a) BSA (pI = 5.5), (b) Hb (pI = 9.0), and (c) HEWL (pI = 11.7). The calculated value for the pI is indicated with a circle. Experimental values reported in the literature are shown with a diamond (pI = 5.5 for BSA,¹ and pI = 11.3 for HEWL²). Plots of net polymer charge as a function of pH for (d) K₅₀ (pKa = 10.45) and (e) E₅₀ (pKa = 4.32). The calculated value for the pKa is indicated with a square. K₅₀ has a calculated charge of 49.998 to 49.751 from pH 6-8 and E₅₀ has a calculated charge of -49.022 to -50.010 from pH 6-8. The grey boxes indicate the pH range that we used for experiments.

Table S2. Table of calculated net charge for the various proteins and peptides used in this study as a function of pH.

	6.0	7.0	8.0
BSA	-7.01	-15.3	-17.0
Hb	10.9	2.8	1.1
HEWL	8.6	8.1	8.0
E ^a	-0.98	-1.0	-1.02
K ^a	1.0	0.99	0.91

^a Charge is listed for a single residue. Net charge can be calculated by multiplying the listed charge by the peptide degree of polymerization.

Table S3. Table of the calculated number of positive and negative charges for the various proteins used in this study as a function of pH.

	6.0			7.0			8.0		
	+	-	+/-	+	-	+/-	+	-	+/-
BSA	90.5	97.5	0.93	83.5	98.9	0.84	82.0	99.0	0.83
Hb	37.5	26.7	1.4	2.8	29.7	1.1	28.1	27.0	1.0
HEWL	17.5	8.9	2.0	17.1	9.0	1.9	17.0	9.0	1.9

Table S4. Table of physical parameters for the model proteins bovine serum albumin (BSA), human hemoglobin (Hb), and hen egg white lysozyme (HEWL) used in this study. These results match closely to those calculated using other computational tools such as ProtParam.

	BSA	Hb	HEWL
Theoretical pI	5.5	9.0	11.7
Optimal pH	8.0	6.0	6.0
pI-pH	2.5	3.0	5.7
EE (%)	54.9 ± 0.1	47.1 ± 0.1	93.6 ± 0.4
MW (kDa)	66.4	64.5	14.3
Total # Residues	583	574	129
# Charged Residues ^a	253	166	38

^a Potentially charged amino acids were defined as aspartate (D), glutamate (E), lysine (K), arginine (R), and histidine (H).

The values for the pI of all three proteins were calculated using a MATLAB script. The calculated values for BSA and HEWL match closely to those reported in the literature. However, Hb has a wide range of reported pIs due to the presence of multiple chains, as well as the wide variety of mutations. The pI for Hb was calculated by running both chains simultaneously in the MATLAB script.

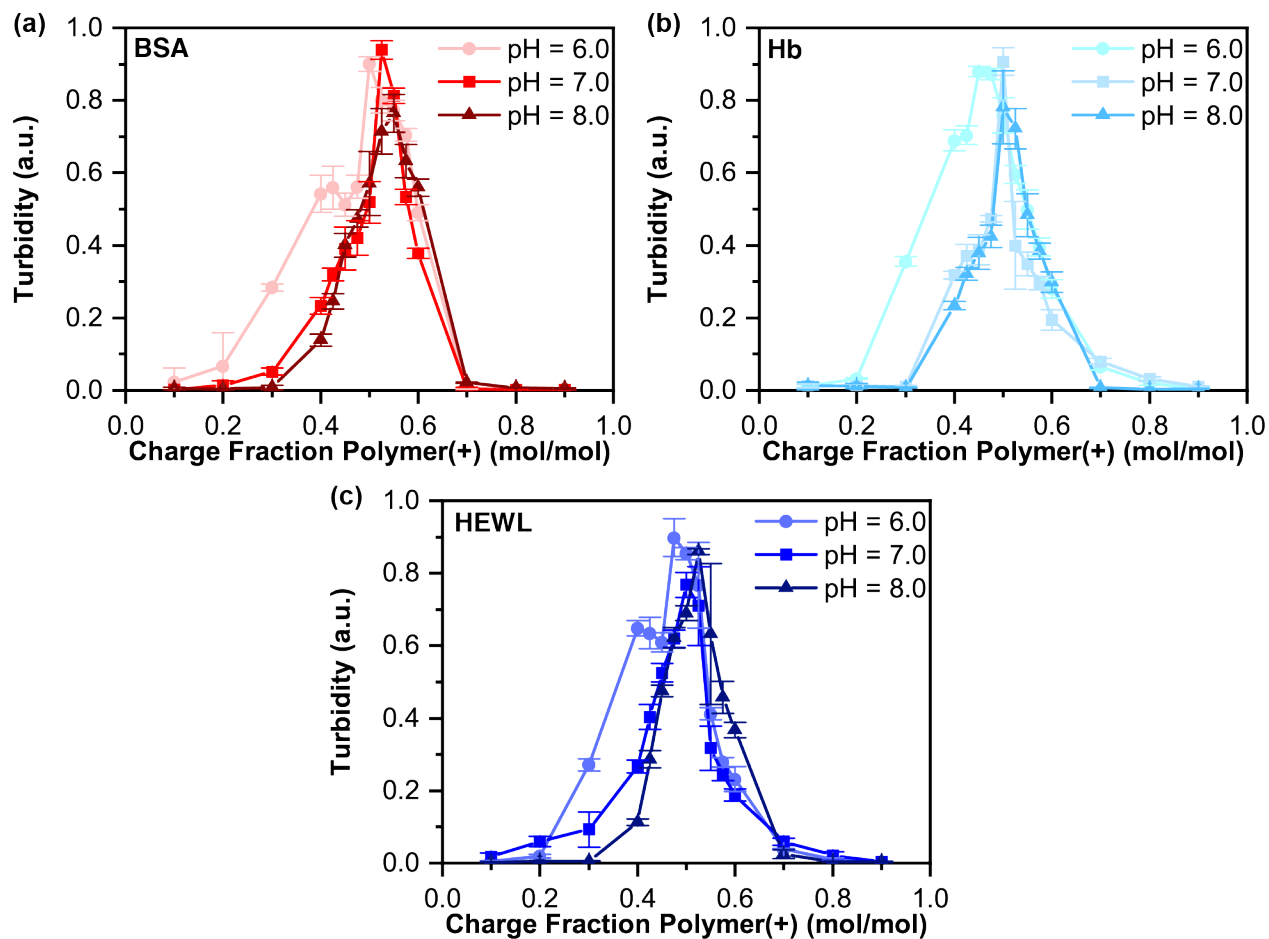


Figure S3. Plots of turbidity as a function of pH for (a) BSA, (b) Hb, and (c) HEWL at pH 6.0, 7.0, and 8.0. Increasing pH is shown by a change in color from light to dark. In each case, we observed a decrease in the range of charge stoichiometries over which phase separation, as well as a shift towards higher cationic charge fractions with increasing pH. Lines connecting data points are a guide to the eye. Error bars are the standard deviation of the reported average, including propagated error.

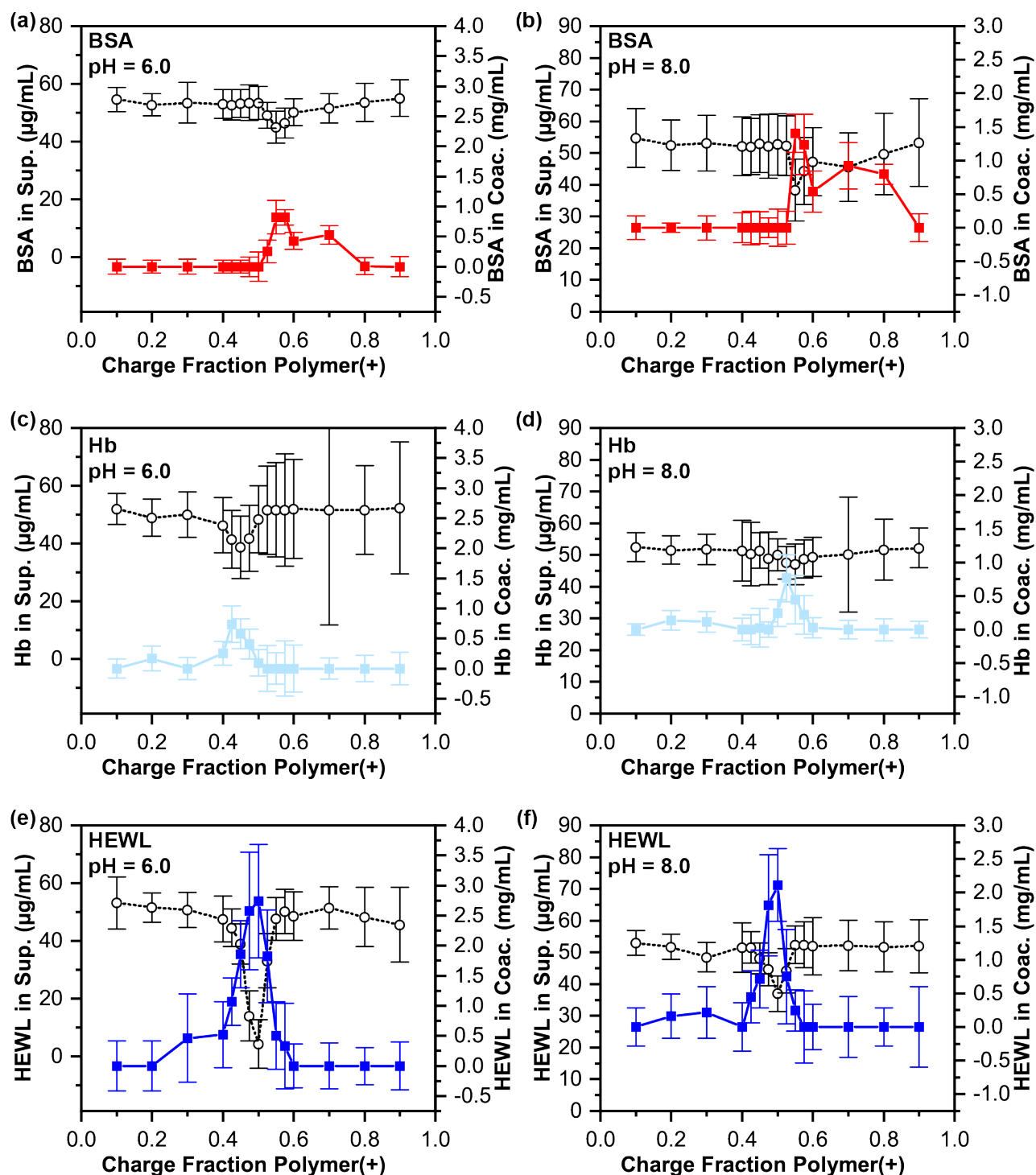


Figure S4. Plots of concentration vs. cationic charge fraction for both the supernatant (open circles) and the coacervate phases (colored squares) at pH 6.0 (a, c, e) and pH 8.0 (b, d, f) for BSA (a, b), Hb (c, d), and HEWL (e, f). We observed a decrease in the amount of protein taken up into the coacervate phase going from pH 6.0 to 8.0 for Hb and HEWL, as the solution pH shifts closer to the pI of the protein, while the opposite trend is observed for BSA, which has an acidic pI. Lines connecting data points are a guide to the eye. Error bars are the standard deviation of the reported average, including propagated error. Quantification of the maximum partitioning and encapsulation efficiency is shown in Figure 3.

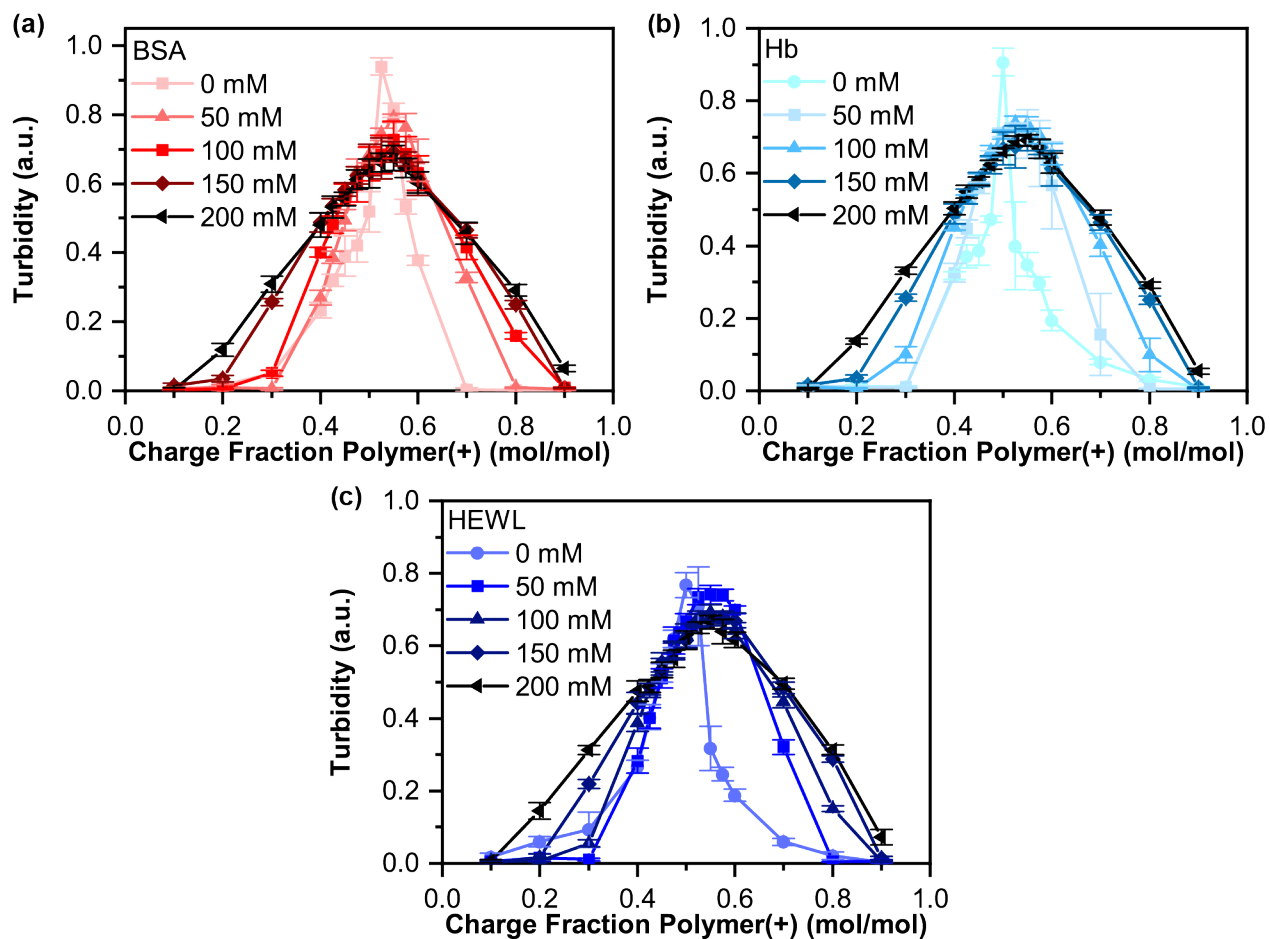


Figure S5. Plots of turbidity as a function of increasing NaCl concentration for (a) BSA, (b) Hb, and (c) HEWL. Increasing salt concentration is shown by a change in color from light to dark. We observed an increase in the range of charge stoichiometries over which phase separation was observed with increasing salt concentration, as salt ions can facilitate charge neutralization of polymer complexes over a wider range of polymer ratios than is possible in the absence of added salt. Lines connecting data points are a guide to the eye. Error bars are the standard deviation of the reported average, including propagated error.

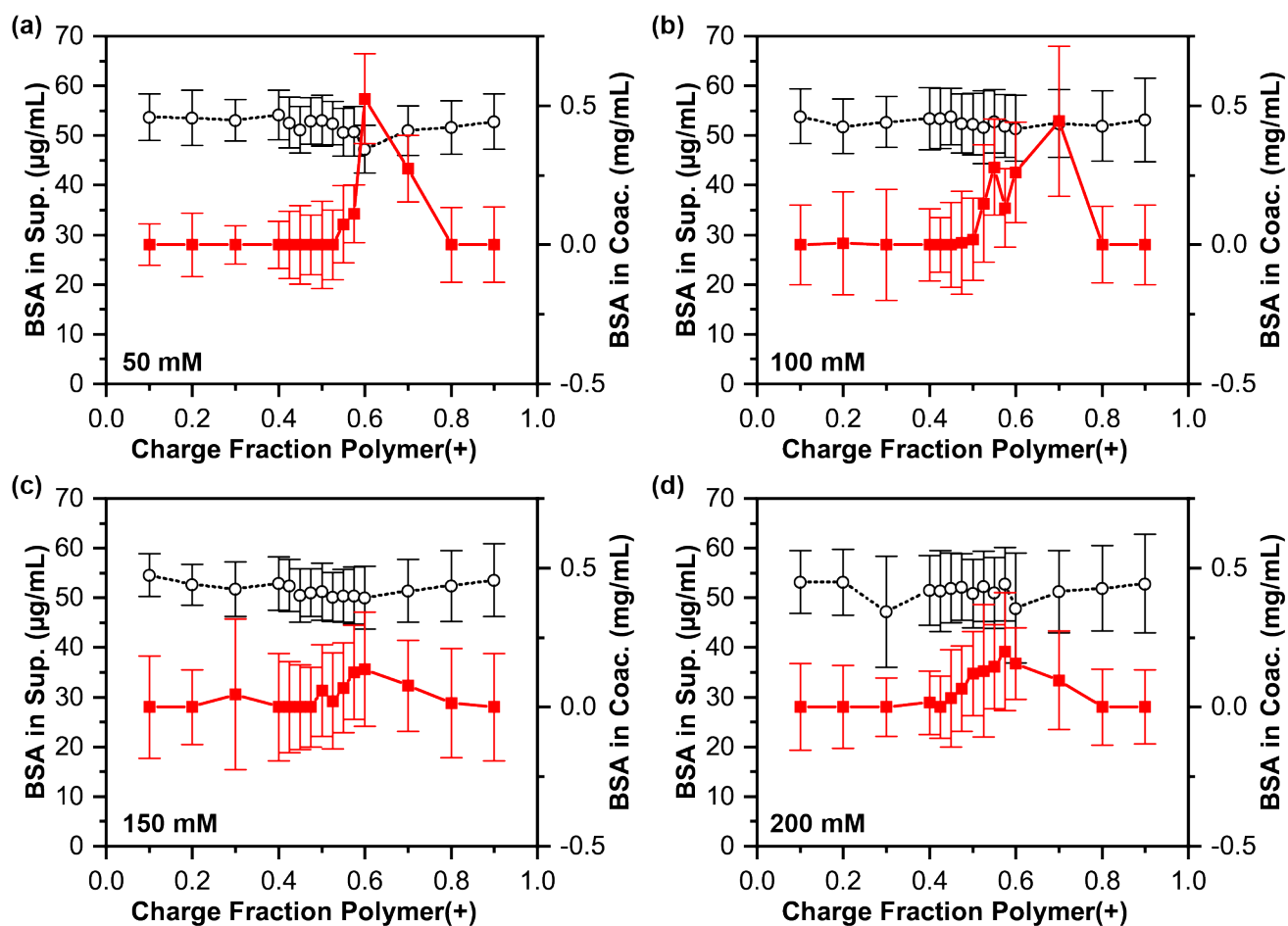


Figure S6. Plots of the concentration of BSA for both the supernatant (open circles) and the coacervate phases (colored squares) as a function of the cationic charge fraction at **(a)** 50 mM, **(b)** 100 mM, **(c)** 150 mM, and **(d)** 200 mM NaCl. We observed a decrease in the amount of protein taken up into the coacervate phase with increasing salt concentration, as seen with HEWL in Figure S7. Lines connecting data points are a guide to the eye. Error bars are the standard deviation of the reported average, including propagated error. Quantification of the maximum partitioning and encapsulation efficiency is shown in Figure 4.

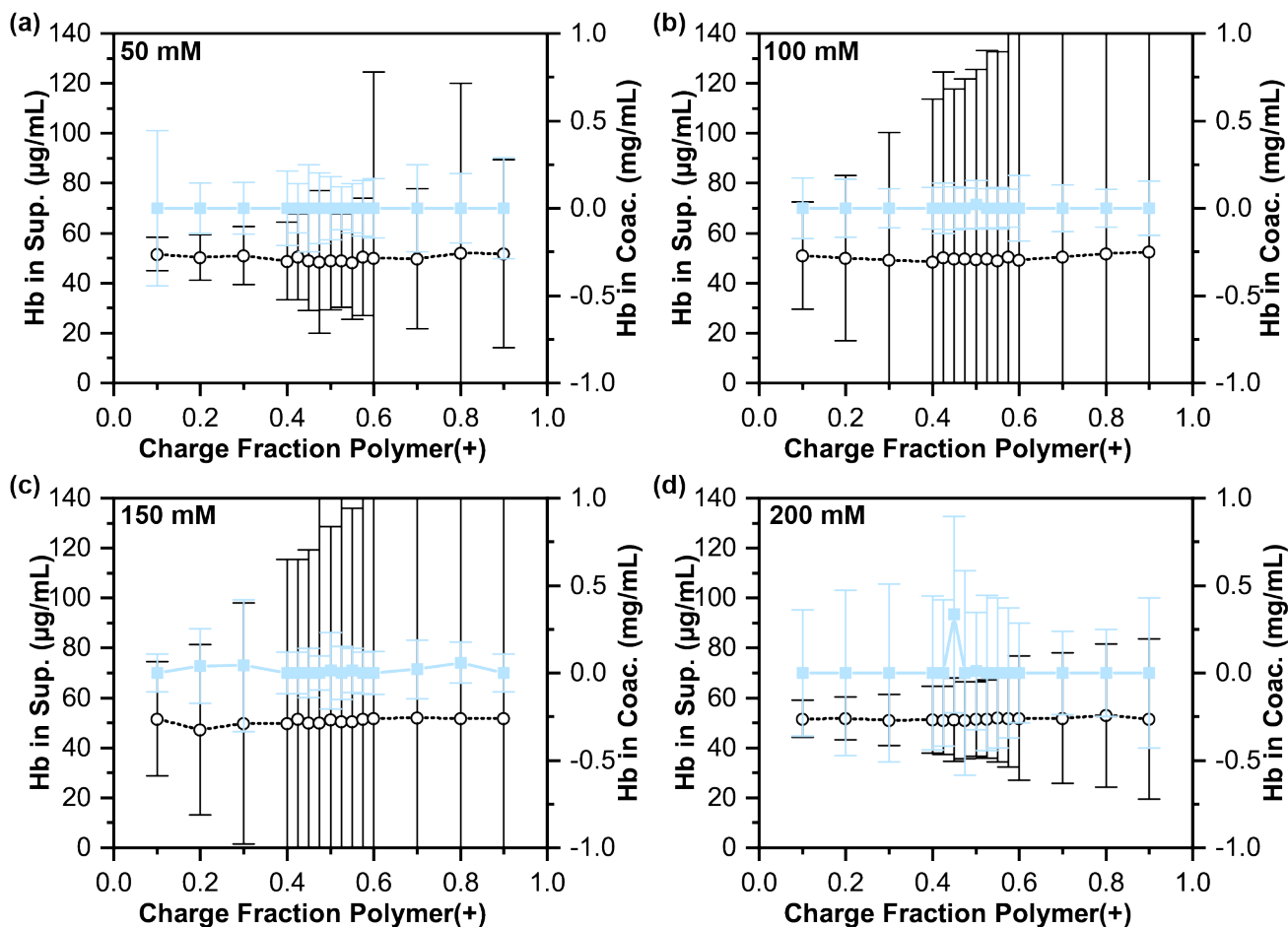


Figure S7. Plots of the concentration of Hb for both the supernatant (open circles) and the coacervate phases (colored squares) as a function of the cationic charge fraction at **(a)** 50 mM, **(b)** 100 mM, **(c)** 150 mM, and **(d)** 200 mM NaCl. We observed very little protein uptake into the coacervate phase in the presence of added salt, resulting in the appearance of significant error due to propagation of uncertainty through our calculations. Lines connecting data points are a guide to the eye. Error bars are the standard deviation of the reported average, including propagated error. Quantification of the maximum partitioning and encapsulation efficiency is shown in Figure 4.

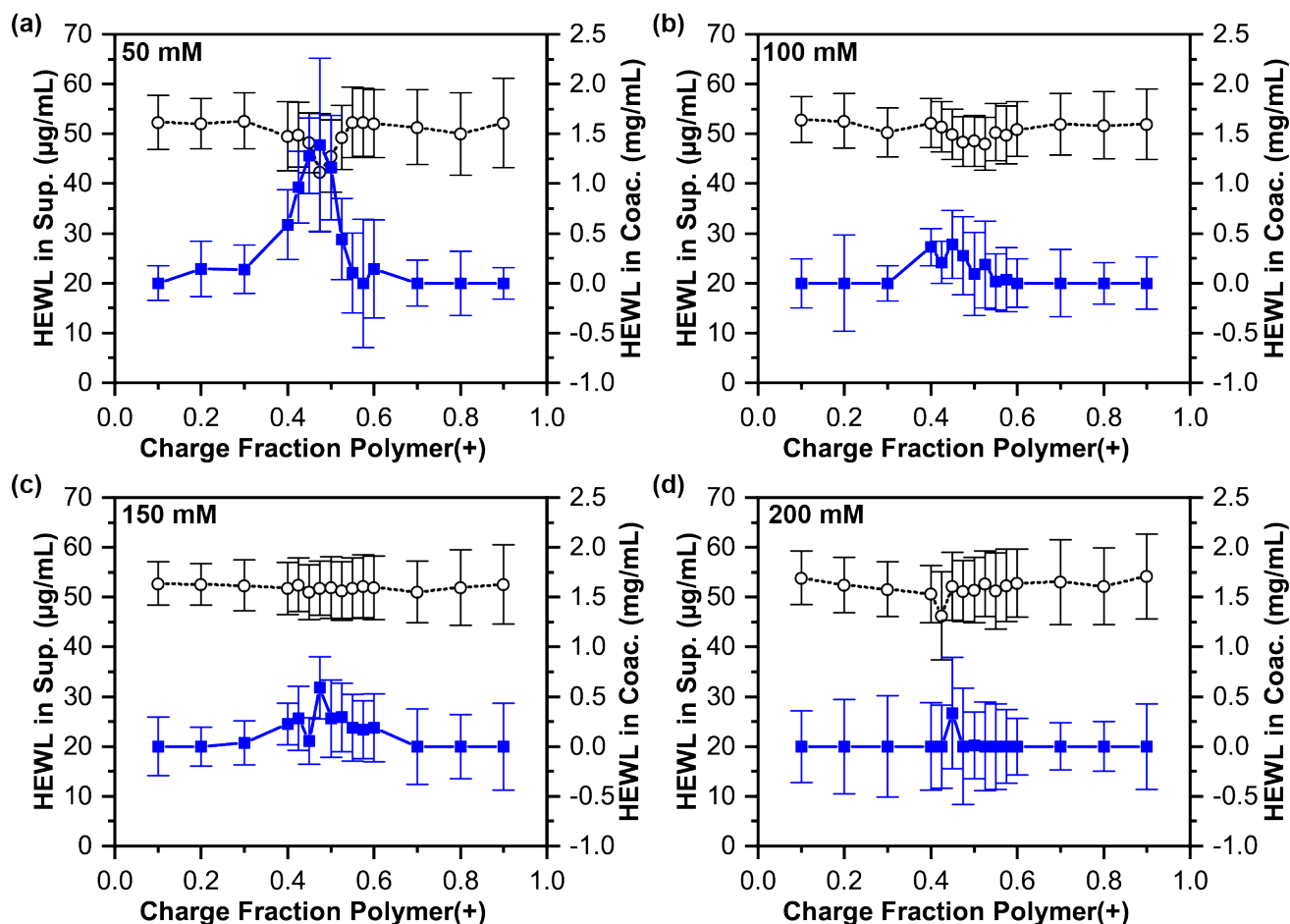


Figure S8. Plots of the concentration of HEWL for both the supernatant (open circles) and the coacervate phases (colored squares) as a function of the cationic charge fraction at (a) 50 mM, (b) 100 mM, (c) 150 mM, and (d) 200 mM NaCl. We observed a decrease in the amount of protein taken up into the coacervate phase with increasing salt concentration, as seen for BSA in Figure S5. Lines connecting data points are a guide to the eye. Error bars are the standard deviation of the reported average, including propagated error. Quantification of the maximum partitioning and encapsulation efficiency is shown in Figure 4.

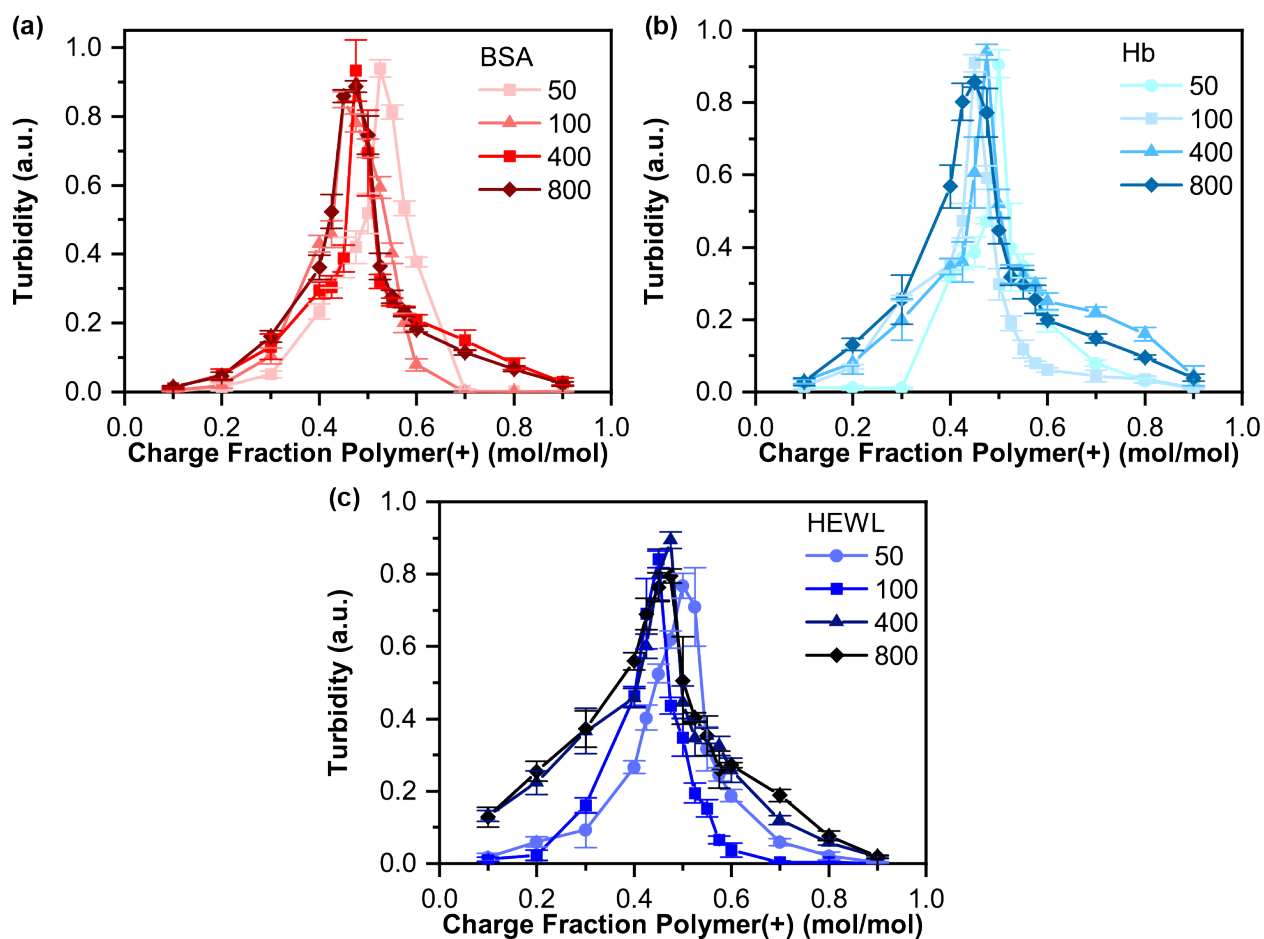


Figure S9. Plots of turbidity as a function of increasing polypeptide chain length for (a) BSA, (b) Hb, and (c) HEWL. Increasing chain length is shown by a change in color from light to dark. We observed an increase in the range of charge stoichiometries over which phase separation was observed with increasing chain length. Lines connecting data points are a guide to the eye. Error bars are the standard deviation of the reported average, including propagated error.

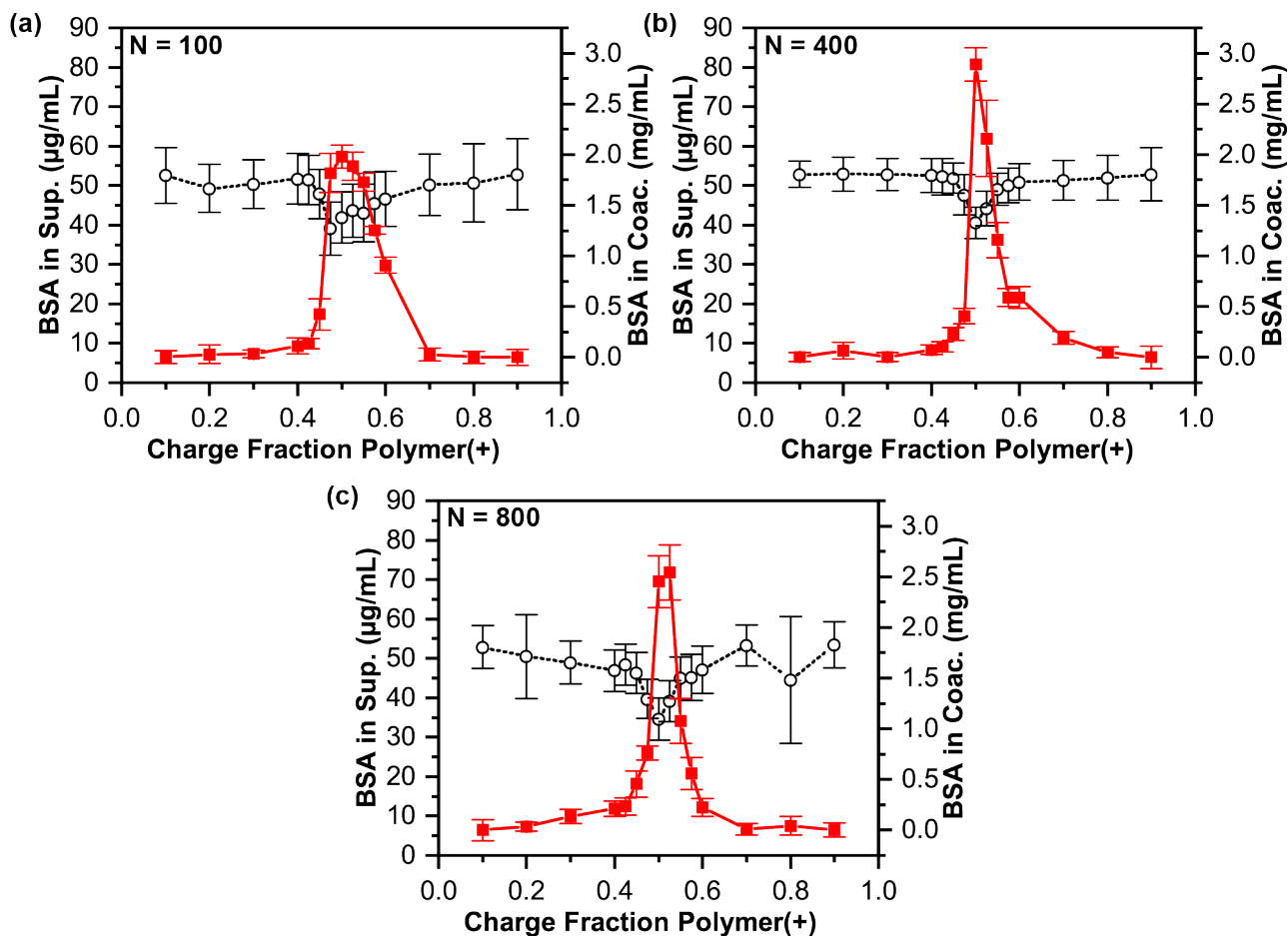


Figure S10. Plots of the concentration of BSA for both the supernatant (open circles) and the coacervate phases (colored squares) as a function of the cationic charge fraction with increasing polypeptide chain length **(a)** $N = 100$, **(b)** $N = 400$, and **(c)** $N = 800$. Lines connecting data points are a guide to the eye. Error bars are the standard deviation of the reported average, including propagated error. Quantification of the maximum partitioning and encapsulation efficiency is shown in Figure 5.

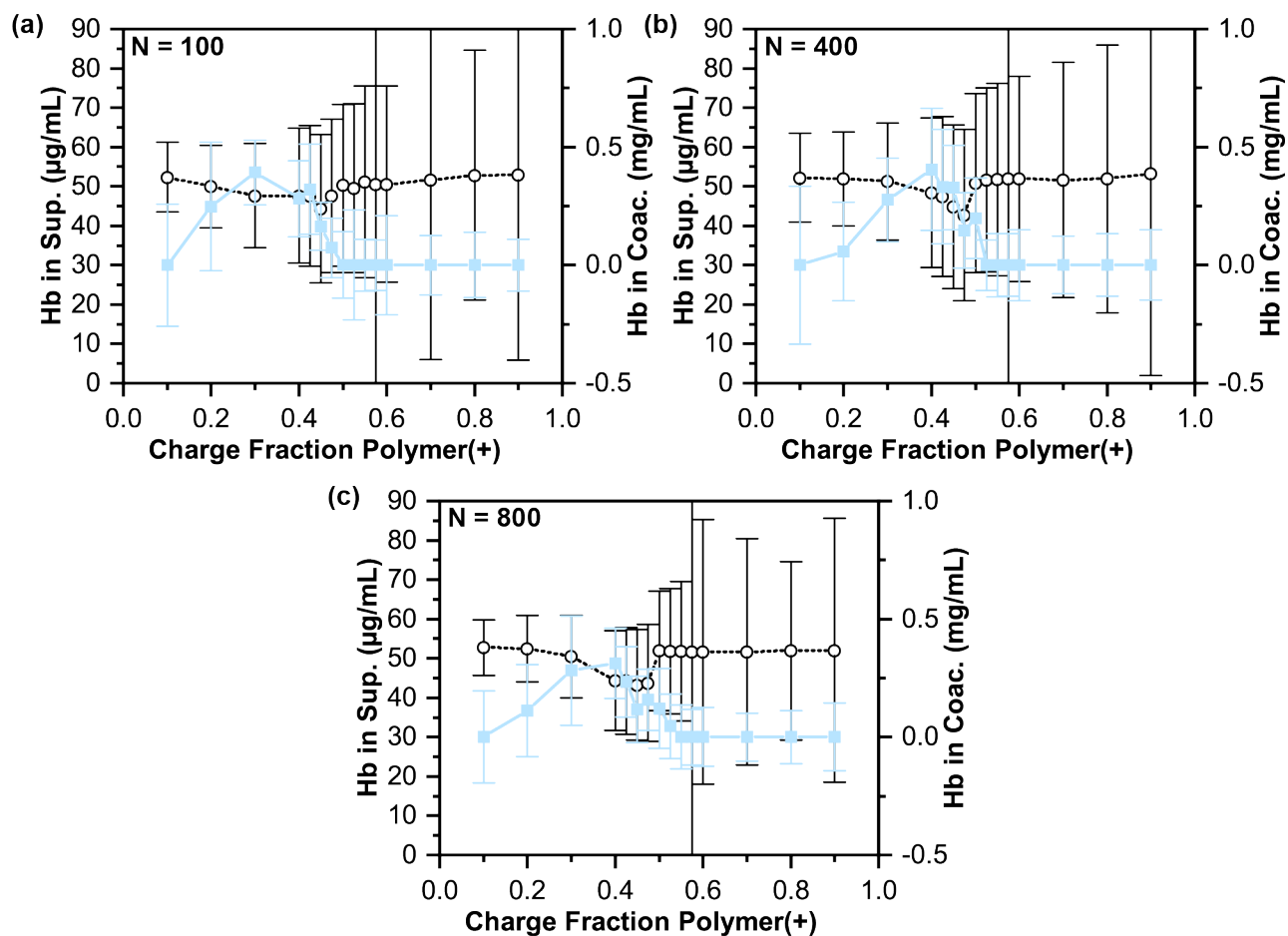


Figure S11. Plots of the concentration of Hb for both the supernatant (open circles) and the coacervate phases (colored squares) as a function of the cationic charge fraction with increasing polypeptide chain length **(a)** $N = 100$, **(b)** $N = 400$, and **(c)** $N = 800$. Lines connecting data points are a guide to the eye. Error bars are the standard deviation of the reported average, including propagated error. Quantification of the maximum partitioning and encapsulation efficiency is shown in Figure 5.

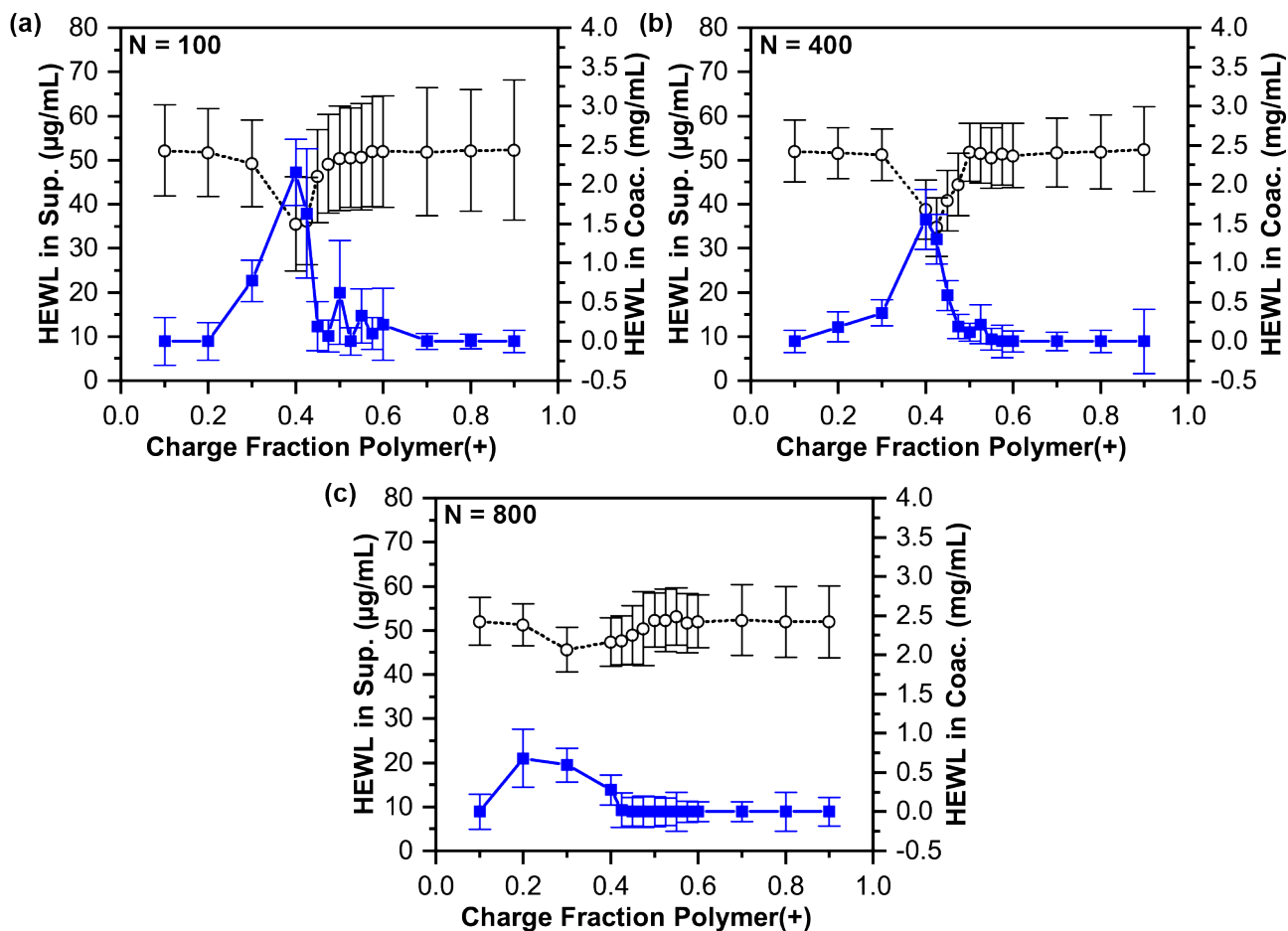


Figure S12. Plots of the concentration of HEWL for both the supernatant (open circles) and the coacervate phases (colored squares) as a function of the cationic charge fraction with increasing polypeptide chain length **(a)** $N = 100$, **(b)** $N = 400$, and **(c)** $N = 800$. We observed a decrease in the amount of protein taken up into the coacervate phase compared with $N = 50$ (Figure 2f) with increasing chain length. Lines connecting data points are a guide to the eye. Error bars are the standard deviation of the reported average, including propagated error. Quantification of the maximum partitioning and encapsulation efficiency is shown in Figure 5.

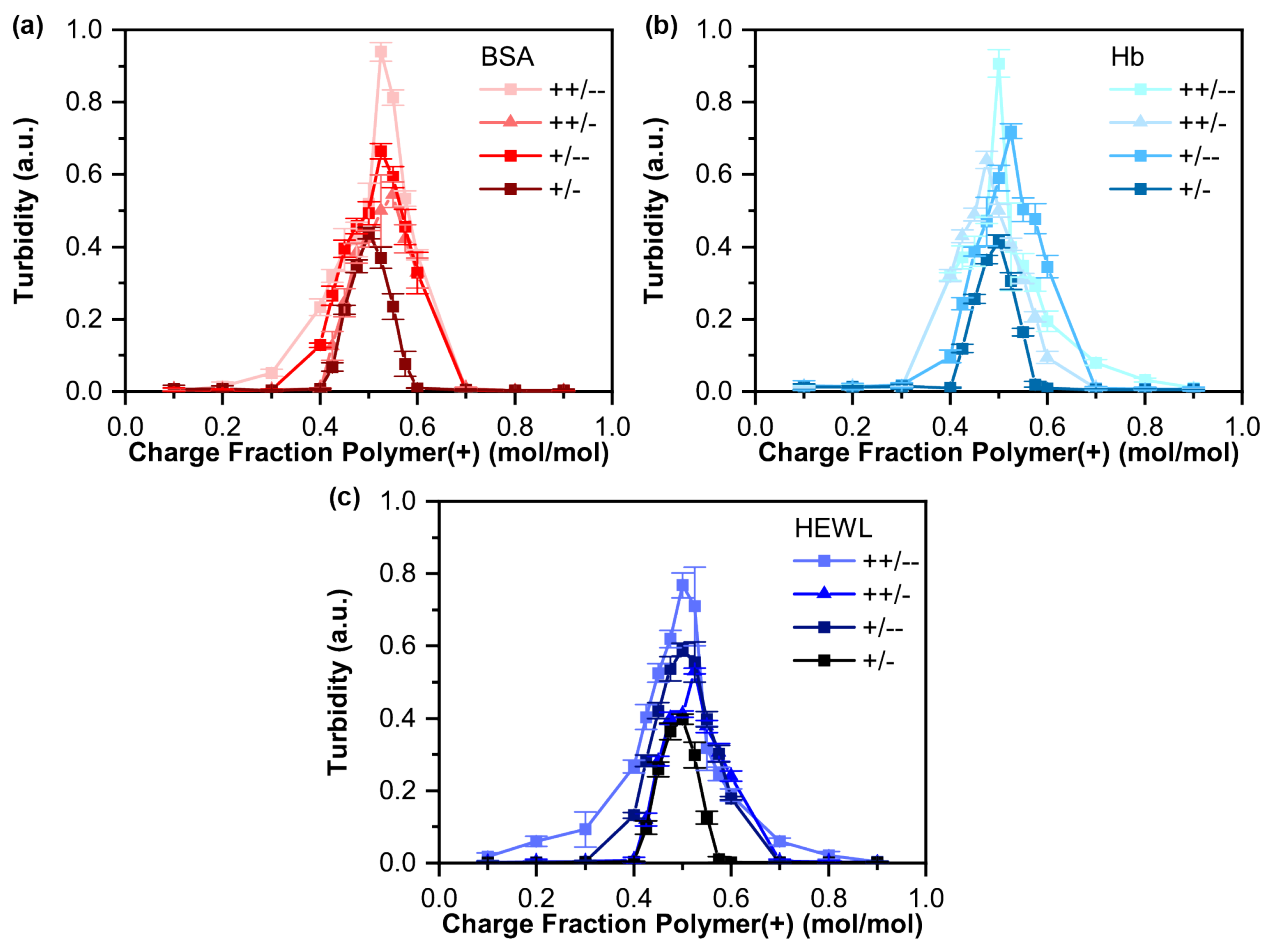


Figure S13. Plots of turbidity as a function of increasing polypeptide charge content for **(a)** BSA, **(b)** Hb, and **(c)** HEWL. The polypeptide systems are denoted: K_{50}/E_{50} as $++/--$, $K_{50}/(EG)_{25}$ as $++/-$, $(KG)_{25}/E_{50}$ as $+/--$, and $(KG)_{25}/(EG)_{25}$ as $+/-$. We observed a decrease in the turbidity signal as the overall charge on the polypeptides decreased from K_{50}/E_{50} (fully charged) to $(KG)_{25}/(EG)_{25}$ (half charged). Lines connecting data points are a guide to the eye. Error bars are the standard deviation of the reported average, including propagated error.

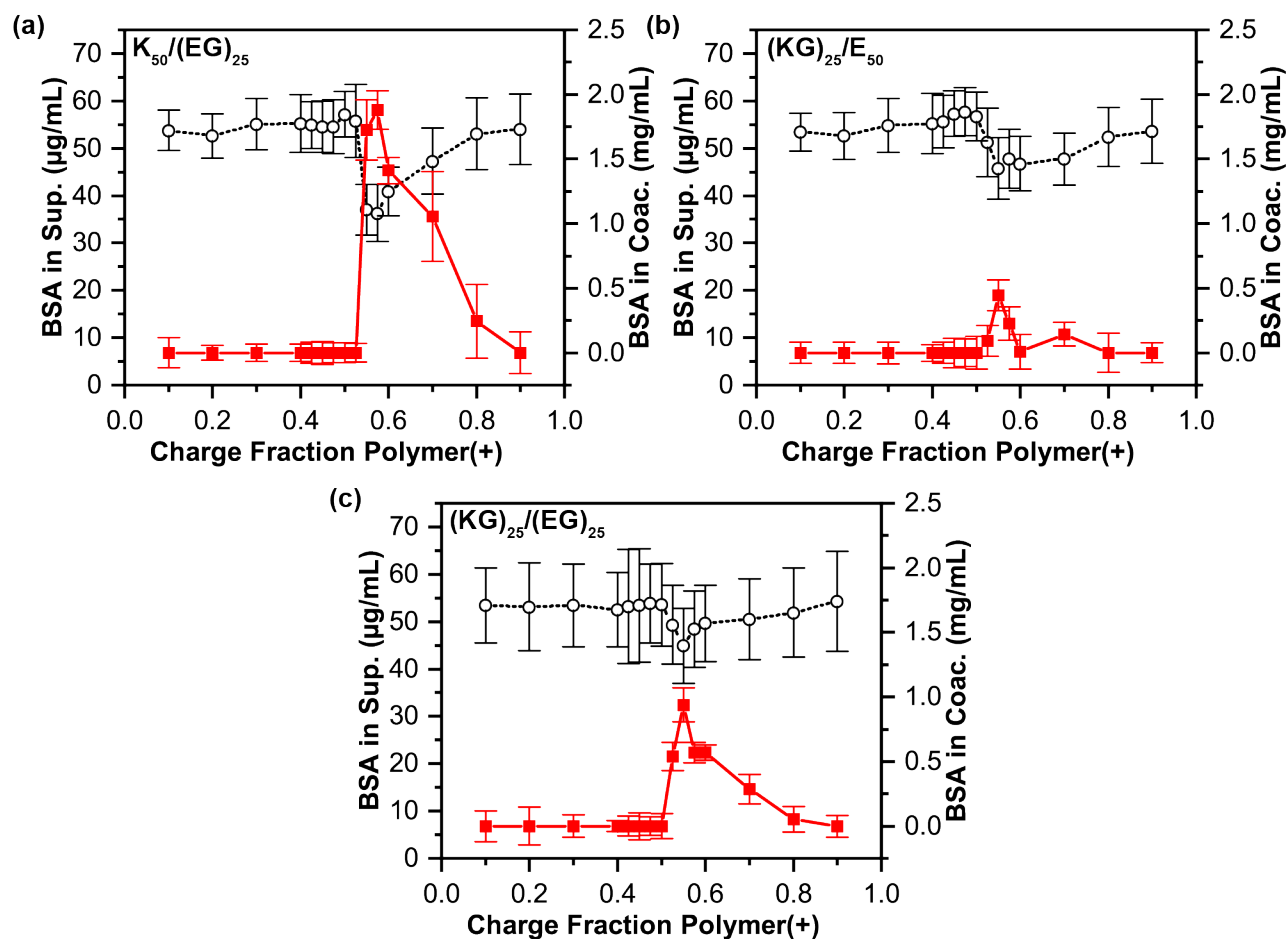


Figure S14. Plots of the concentration of BSA for both the supernatant (open circles) and the coacervate phases (colored squares) as a function of the cationic charge fraction for systems with different amounts of polypeptide charge **(a)** $K_{50}/(EG)_{25}$, **(b)** $(KG)_{25}/E_{50}$, and **(c)** $(KG)_{25}/(EG)_{25}$. We observed the highest levels of protein uptake into the coacervate phase when a fully charged cationic polypeptide was used, and the lowest when a half-charged cationic peptide used, indicating favorable interactions between the anionic BSA and the cationic polypeptide. Lines connecting data points are a guide to the eye. Error bars are the standard deviation of the reported average, including propagated error. Quantification of the maximum partitioning and encapsulation efficiency is shown in Figure 8.

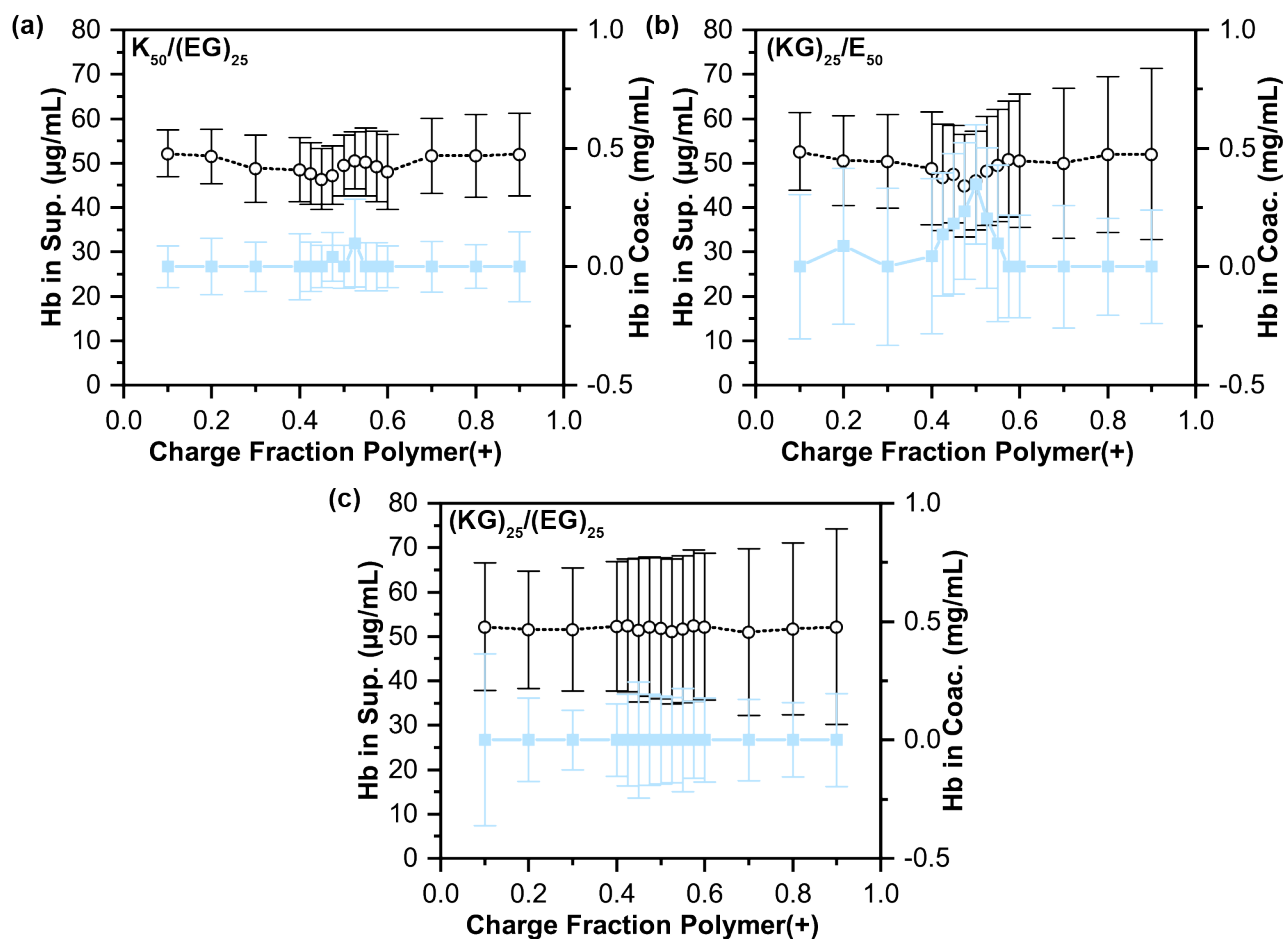


Figure S15. Plots of the concentration of Hb for both the supernatant (open circles) and the coacervate phases (colored squares) as a function of the cationic charge fraction for systems with different amounts of polypeptide charge **(a)** $K_{50}/(EG)_{25}$, **(b)** $(KG)_{25}/E_{50}$, and **(c)** $(KG)_{25}/(EG)_{25}$. We observed the highest levels of protein uptake into the coacervate phase when a fully charged anionic polypeptide was used, and the lowest when a half-charged anionic peptide used, indicating favorable interactions between the cationic Hb and the anionic polypeptide. Lines connecting data points are a guide to the eye. Error bars are the standard deviation of the reported average, including propagated error. Quantification of the maximum partitioning and encapsulation efficiency is shown in Figure 8.

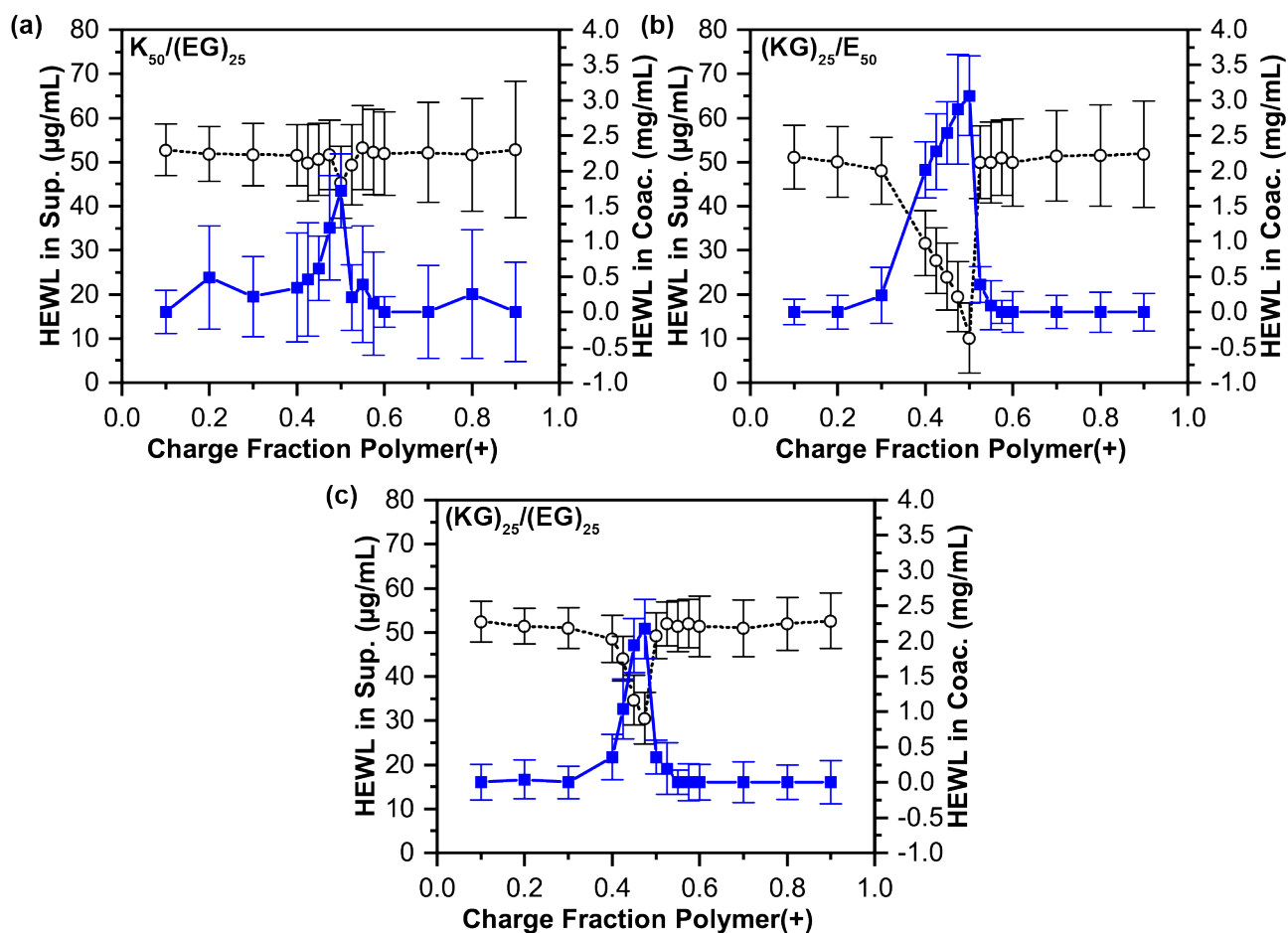


Figure S16. Plots of the concentration of HEWL for both the supernatant (open circles) and the coacervate phases (colored squares) as a function of the cationic charge fraction for systems with different amounts of polypeptide charge **(a)** $K_{50}/(EG)_{25}$, **(b)** $(KG)_{25}/E_{50}$, and **(c)** $(KG)_{25}/(EG)_{25}$. We observed the highest levels of protein uptake into the coacervate phase when a fully charged anionic polypeptide was used, and the lowest when a half-charged anionic peptide used, indicating favorable interactions between the cationic HEWL and the anionic polypeptide. Lines connecting data points are a guide to the eye. Error bars are the standard deviation of the reported average, including propagated error. Quantification of the maximum partitioning and encapsulation efficiency is shown in Figure 8.

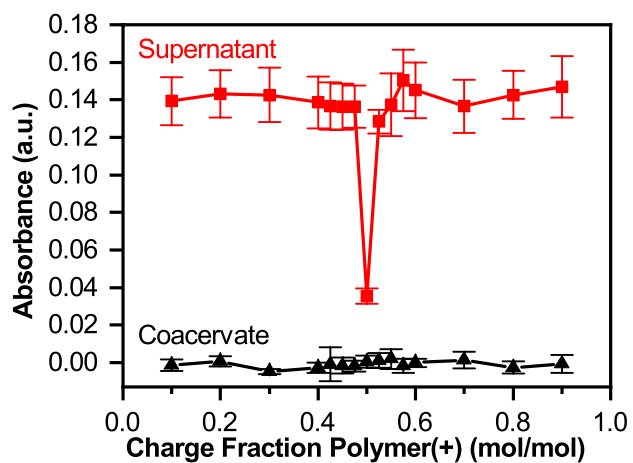


Figure S17. Plots of Coomassie dye absorbance for polymer-only supernatant and coacervate samples as a function of the charge fraction of the polycation for coacervates of K_{50} and E_{50} , in the absence of protein in 10 mM HEPES, pH 7.0. Lines connecting data points are a guide to the eye. Error bars are the standard deviation of the reported average.

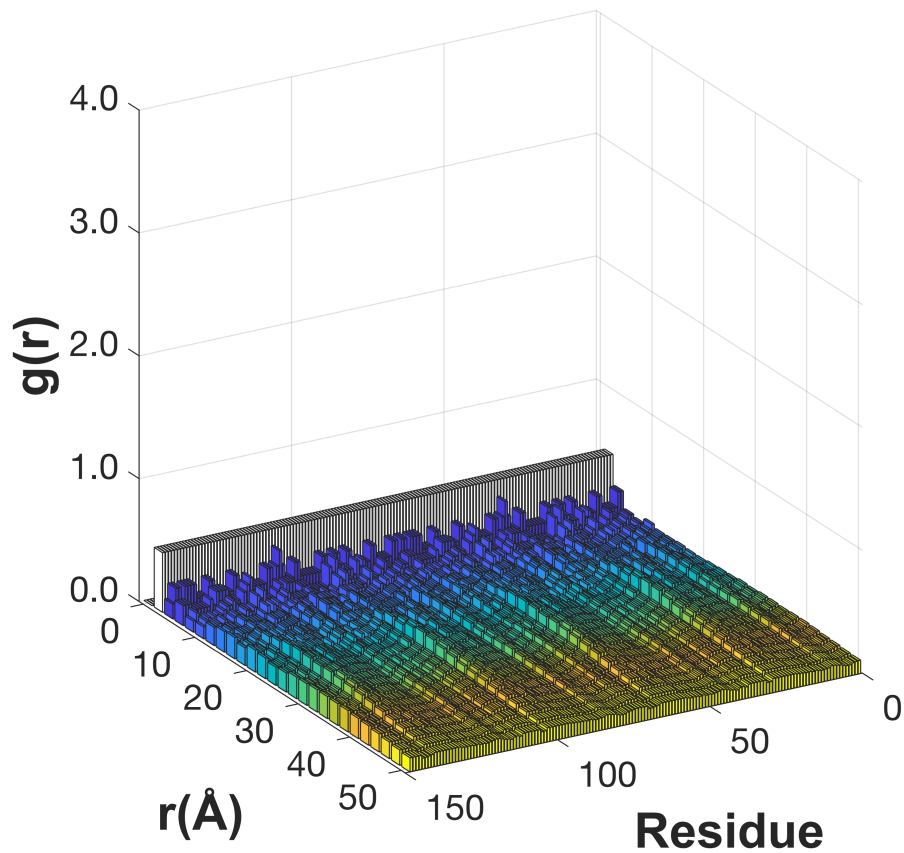


Figure S18. 3D bar plot depiction of the radial distribution function $g(r)$ with respect to the charged amino acids in the full assembly Hb.

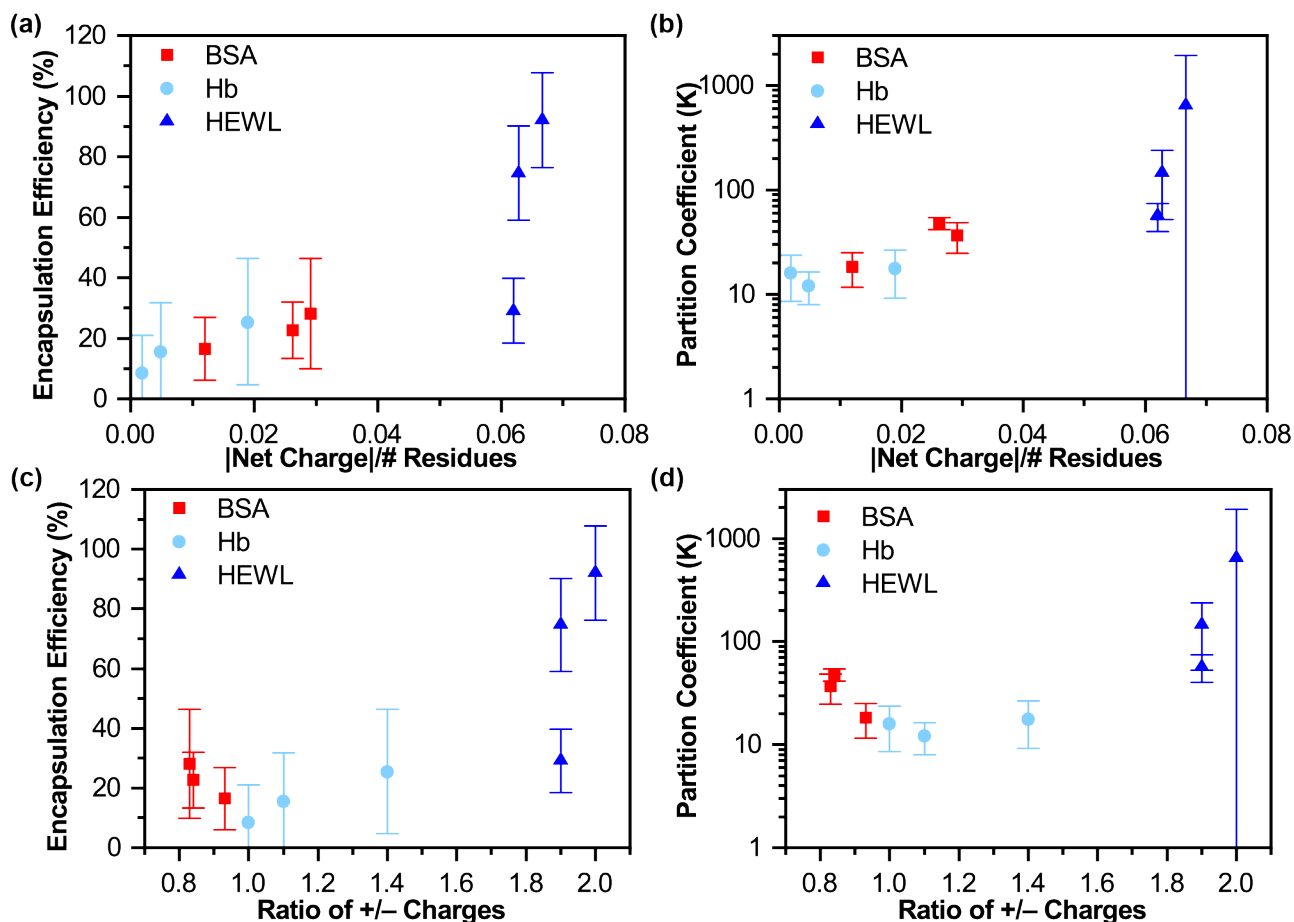


Figure S19. Plots of encapsulation efficiency (a,c) and partition coefficient (b,d) as a function of net charge per number of residues (a,b), and the ratio of positive to negative charges (c,d). Error bars are the standard deviation of the reported average, including propagated error.

References:

- (1) Tanford, C.; Swanson, S. A.; Shore, W. S. Hydrogen Ion Equilibria of Bovine Serum Albumin. *JACS* **1955**, *77* (24), 6414–6421 DOI: 10.1021/ja01629a002.
- (2) van der Veen, M.; Norde, W.; Stuart, M. C. Electrostatic interactions in protein adsorption probed by comparing lysozyme and succinylated lysozyme. *Colloids and Surfaces B: Biointerfaces* **2004**, *35* (1), 33–40 DOI: 10.1016/j.colsurfb.2004.02.005.

MATLAB Code for pI Calculations

```
function AASeq_Charge_Calc

% Amino Acid Sequence Charge Calculator by Whitney Blocher McTigue

% Amino acid information taken from pp 1086-88 of Organic Chemistry by T.W.
% Graham Solomons and Craig B. Fryhle, 10th edition

% Calculations based on Henderson-Hasselbalch equation and Moore, 'Amino
% Acid and Peptide Net Charges: A simple Computational Procedure,'
% Biochemical Education, 13(1), 1985.

clear, clc

seq = input('AA Sequence as a String:\n'); % Sequence input
% This input needs to have single quotes around it.
pH = input('pH Value or Range:\n'); % pH range input

n = length(seq); % Determines the length of the sequence

D = count(seq, 'D'); % Aspartate (Asp)
E = count(seq, 'E'); % Glutamate (Glu)
K = count(seq, 'K'); % Lysine (Lys)
R = count(seq, 'R'); % Arginine (Arg)
H = count(seq, 'H'); % Histidine (His)

Nterm = seq(n); % The N-terminus
Cterm = seq(1); % The C-terminus

% Determines what the N-terminus is and lists the pKa.
if Nterm == 'A'
    NT = 9.7;
elseif Nterm == 'R'
    NT = 9.0;
elseif Nterm == 'N'
    NT = 8.8;
elseif Nterm == 'D'
    NT = 9.8;
elseif Nterm == 'C'
    NT = 10.8;
elseif Nterm == 'Q'
    NT = 9.1;
elseif Nterm == 'E'
    NT = 9.7;
elseif Nterm == 'G'
    NT = 9.6;
elseif Nterm == 'H'
    NT = 9.2;
elseif Nterm == 'I'
    NT = 9.7;
elseif Nterm == 'L'
    NT = 9.6;
elseif Nterm == 'K'
    NT = 9.0;
elseif Nterm == 'M'
    NT = 9.2;
elseif Nterm == 'F'
    NT = 9.1;
elseif Nterm == 'P'
    NT = 10.6;
```

```

elseif Nterm == 'S'
    NT = 9.2;
elseif Nterm == 'T'
    NT = 10.4;
elseif Nterm == 'W'
    NT = 9.4;
elseif Nterm == 'Y'
    NT = 9.1;
elseif Nterm == 'V'
    NT = 9.6;
end

% Determines what the C-terminus is and lists the pKa.
if Cterm == 'A'
    CT = 2.3;
elseif Cterm == 'R'
    CT = 2.2;
elseif Cterm == 'N'
    CT = 2.0;
elseif Cterm == 'D'
    CT = 2.1;
elseif Cterm == 'C'
    CT = 1.7;
elseif Cterm == 'Q'
    CT = 2.2;
elseif Cterm == 'E'
    CT = 2.1;
elseif Cterm == 'G'
    CT = 2.3;
elseif Cterm == 'H'
    CT = 1.8;
elseif Cterm == 'I'
    CT = 2.4;
elseif Cterm == 'L'
    CT = 2.4;
elseif Cterm == 'K'
    CT = 2.2;
elseif Cterm == 'M'
    CT = 2.3;
elseif Cterm == 'F'
    CT = 1.8;
elseif Cterm == 'P'
    CT = 2.0;
elseif Cterm == 'S'
    CT = 2.2;
elseif Cterm == 'T'
    CT = 2.6;
elseif Cterm == 'W'
    CT = 2.4;
elseif Cterm == 'Y'
    CT = 2.2;
elseif Cterm == 'V'
    CT = 2.3;
end

% Calculates the charge at each pH value
Q = 1./(1+10.^(pH-NT)) - 1./(1+10.^(-(pH-CT))) - D./(1+10.^(-(pH-3.9)))...
    - E./(1+10.^(-(pH-4.3))) + K./(1+10.^(pH-10.5))...
    + R./(1+10.^(pH-12.5)) + H./(1+10.^(pH-6.0));

% Calculations for pI value
pH_pI = 0.1:0.001:14; % pH range for the calculation of pI

```

```

Q_pI = 1./(1+10.^(pH_pI-NT)) - 1./(1+10.^(-(pH_pI-CT))) - D./(1+10.^(-(pH_pI-3.9)))...
    - E./(1+10.^(-(pH_pI-4.3))) + K./(1+10.^(pH_pI-10.5))...
    + R./(1+10.^(pH_pI-12.5)) + H./(1+10.^(pH_pI-6.0));
% Charge calculation for each pH point

ub = find(Q_pI<0, 1, 'first'); % When the charge first becomes positive
lb = find(Q_pI>0, 1, 'last'); % When the charge is last to be negative
    % Finds the upper and lower bounds of the sequence where it most closely
    % matches 0.

pI = 0.5.*(0.5.*(2.*pH_pI(lb)-1) + 0.5.*(2.*pH_pI(ub)+1));
    % Uses the upper and lower bounds to then calculate the pI.

pos_Q = K./(1+10.^(pH-10.5)) + R./(1+10.^(pH-12.5)) + H./(1+10.^(pH-6.0));
    % Calculates number of positive charges
neg_Q = D./(1+10.^(-(pH-3.9))) + E./(1+10.^(-(pH-4.3)));
    % Calculates number of negative charges
total_Q = pos_Q./neg_Q;
    figure(2)
    plot(pH_pI, Q_pI)
    xlabel('pH')
    ylabel('Charge')
    title(['Charge vs. pH for ' input('name of sequence\n')])
    hold on
    plot(pI, 0, 'mo')
    hold off

3 = [pH; Q; pos_Q; neg_Q; total_Q];
fprintf('pI = %0.2f\n',pI)
fprintf('pH\t\tQ\t\tPos Q\t\tNeg Q\t\t#Q\n')
fprintf('%0.1f\t\t%0.2f\t\t%0.2f\t\t%0.2f\t\t%0.2f\n',table)

```

MATLAB Script for Pair-Wise Interaction Calculations

The below MATLAB function reads in the information obtained from the corresponding PDB files from an Excel file. The residues are then indexed and, walking along the chain, count the number of charged residues within a specified radius. This number is normalized based on the number of total residues, the volume of the shell that is considered, and the particle density. This calculation only considered aspartate (D), glutamate (E), lysine (K), arginine (R), and histidine (H) as charged residues, and did not take into account possible pH effects or the charge on the N and C termini of the protein.

```
function PDB_Corr

% g(r) Charge Correlation Calculator by Sarah L. Perry

[num,txt,~] = xlsread(input('Type in your filename with extension:\n')); %read in the
modified PDB file

% Separate the information from the PDB file into a numerical array and a
% character array.
myPDBnum(:,1) = num(:,1); % copy the (1) atom index
myPDBnum(:,2:5) = num(:,5:8); % copy the (5) residue index, (6) X, (7) Y, and (8) Z
coordinates
myPDBnum(~any(~isnan(myPDBnum)),:)=[]; % remove any NaNs

myPDBtxt(:,1:2) = txt(:,3:4); % copy the (3) atom name and (4) residue name

arraysize = size(myPDBnum(:,1)); % determine the length of the PDB array

%% Plan of attack
% Scan through the second column of myPDB (atom name), looking for CA.
% For each CA check the identity of the residue.
% If the residue is LYS, HIS, ARG, GLU, ASP then copy to a new matrix.
% Will need to increment a counter for this.

newarrayindex = 1; % initialize a new index for the charged residue matrix

for i = 1:arraysize
    myPDBatom = char(myPDBtxt(i,1)); % create a string array for the atom name
    atomname = 'CA'; % specify search string for atom type

    myPDBres = char(myPDBtxt(i,2)); % create a string array for the residue name
    Lys = 'LYS'; % specify search strings for residue type
    His = 'HIS';
    Arg = 'ARG';
    Glu = 'GLU';
    Asp = 'ASP';

    if strcmp(myPDBatom,atomname) == 1
        if strcmp(myPDBres,Lys) == 1 % check if a Lys

            myChargedtxt(newarrayindex,:) = myPDBtxt(i,:);
            myChargednum(newarrayindex,:) = myPDBnum(i,:);

            newarrayindex = newarrayindex + 1; % increment the counter

        elseif strcmp(myPDBres,His) == 1 % check if a His

            myChargedtxt(newarrayindex,:) = myPDBtxt(i,:);
```



```

        myChargednum(newarrayindex,:) = myPDBnum(i,:);

        newarrayindex = newarrayindex + 1; % increment the counter

elseif strcmp(myPDBres,Arg) == 1 % check if an Arg

        myChargedtxt(newarrayindex,:) = myPDBtxt(i,:);
        myChargednum(newarrayindex,:) = myPDBnum(i,:);

        newarrayindex = newarrayindex + 1; % increment the counter

elseif strcmp(myPDBres,Glu) == 1 % check if a Glu

        myChargedtxt(newarrayindex,:) = myPDBtxt(i,:);
        myChargednum(newarrayindex,:) = myPDBnum(i,:);

        newarrayindex = newarrayindex + 1; % increment the counter

elseif strcmp(myPDBres,Asp) == 1 % check if an Asp

        myChargedtxt(newarrayindex,:) = myPDBtxt(i,:);
        myChargednum(newarrayindex,:) = myPDBnum(i,:);

        newarrayindex = newarrayindex + 1; % increment the counter
else
    'not charged'; % do nothing
end

else
    % do nothing
end

end

end

%% Calculation of Distances
% We now have condensed arrays that only contain information for the
% location of the CA atoms of the charged residues in the protein. Now go
% through this array and calculate the distance between all of the atoms,
% using a loop structure to avoid double counting.

for i = 1:newarrayindex-1

    refx = myChargednum(i,3); % extract the X coordinates for the reference atom
    refy = myChargednum(i,4); % extract the Y coordinates for the reference atom
    refz = myChargednum(i,5); % extract the Z coordinates for the reference atom

    for j = i:newarrayindex-1

        testx = myChargednum(j,3); % extract the X coordinates for the reference atom
        testy = myChargednum(j,4); % extract the Y coordinates for the reference atom
        testz = myChargednum(j,5); % extract the Z coordinates for the reference atom

        distancematrix(i,j) = distance(refx,refy,refz,testx,testy,testz);
        distancematrix(j,i) = distancematrix(i,j); % Diagonal symmetry of the array

    end

end

```

```

end

%% Determination of g(r)
% We will now use the distances determined above to calculate the pair
% correlation function g(r) for a given reference residue

numresidues = newarrayindex-1; % The number of charged residues
maxdistance = max(max(distancematrix)); % Identify the largest residue-residue
separation distance
particledensity = numresidues./(4./3.*pi.*maxdistance.^3); % Calculate the particle
density based
    % on the number of charged residues and the volume of a sphere with radius
equal to max distance
step = 2; % Define the step size for g(r) calculation
rmax = 50; % Define the maximum distance for the g(r) calculation
x = 0:step:rmax; % Create the array of r for graphing

for m = 1:numresidues

    gofrindex = 1; % Initialize an index for saving stuff to g(r)

    for r = 0:step:rmax
        particlecount = 0; % Initialize the particle counter each time through the loop

        for k = 1:numresidues

            if distancematrix(m,k) <= r % if the residue of interest is within the
specified radius
                particlecount = particlecount + 1; % then increment the counter
            else
                % Do nothing
            end

        end

        gofr(m, gofrindex) =
particlecount./numresidues./(4.*pi.*r.^2.*step)./particledensity; % Calculate g(r)

        gofrindex = gofrindex + 1; % Increment the gofrindex

    end

end

end

graph_gofr = gofr;
graph_gofr(:,1) = 0;
graph_gofr(:,2) = 0;
graph_gofr(:,3) = 0;

% Plot the resulting g(r)
figure(2)
bar3(gofr)
xticks([1 6 11 16 21 26])
xticklabels([0 10 20 30 40 50])
xlabel('r(≈)', 'fontsize', 14, 'fontweight', 'bold', 'FontName', 'Arial')
ylabel('Residue', 'fontsize', 14, 'fontweight', 'bold', 'FontName', 'Arial')
zlabel('g(r)', 'fontsize', 14, 'fontweight', 'bold', 'FontName', 'Arial')
ztickformat('%0.1f')
zlim([0 4])

```

```
ghreo = gofr(:,1:2);
hold on
bar3(ghreo,'w')
hold off
figname = input('Input title:\n');
axis square
view(59.8,22)
print(figure(2), '-depsc', figname,'-r600');

%% Subfunctions

function out = distance(refx,refy,refz,testx,testy,testz) % subfunction for
calculating distance

    out = sqrt((refx-testx).^2 + (refy-testy).^2 + (refz-testz).^2);

end
end
```